Пограничный контроллер сессий

ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350, vESBC

Руководство по эксплуатации

Версия ПО 1.9.0

Содержание

1 Введение ... 10
1.1 Аннотация ... 10
1.2 Целевая аудитория ... 10
1.3 Условные обозначения ... 10
1.4 Примечания и предупреждения ... 11
2 Описание изделий ... 12
2.1 Назначение ... 12
2.2 Функции ... 13
2.2.1 Функции интерфейсов ... 13
2.2.2 Функции при работе с MAC-адресами ... 13
2.2.3 Функции второго уровня сетевой модели OSI ... 14
2.2.4 Функции третьего уровня сетевой модели OSI ... 14
2.2.5 Функции туннелирования трафика ... 15
2.2.6 Функции управления и конфигурирования ... 16
2.2.7 Функции сетевой защиты ... 17
2.3 Основные технические характеристики ... 18
2.4 Конструктивное исполнение ... 22
2.4.1 Конструктивное исполнение ESBC-15R ... 22
2.4.2 Конструктивное исполнение ESBC-31 ... 24
2.4.3 Конструктивное исполнение ESBC-3200 ... 27
2.4.4 Конструктивное исполнение ESBC-3250 ... 29
2.4.5 Конструктивное исполнение ESBC-3350 ... 32
2.4.6 Световая индикация ... 34
2.5 Комплект поставки ... 40
2.6 Пусковые токи ESBC-15R, ESBC-31, ESBC-3200 ... 42
3 Установка и подключение ... 43
3.1 Установка устройства в стойку (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250) ... 43
3.2 Установка ESBC-3350 в стойку ... 44
3.3 Установка модулей питания (ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350) ... 46
3.4 Подключение питающей сети ... 47
3.5 Установка и удаление SFP-трансиверов ... 48
3.5.1 Установка трансивера ... 48
3.5.2 Удаление трансивера ... 48
3.6 Подключение к vESBC ... 48
4 Интерфейсы управления ... 49

4.1 Интерфейс командной строки (CLI) ... 49
4.2 Web-интерфейс ... 50
4.3 Типы и порядок именования сетевых интерфейсов пограничного контроллера сессий ... 50
4.4 Типы и порядок именования туннелей пограничного контроллера сессий ... 53
5 Начальная настройка устройства ... 55
5.1 Заводская конфигурация устройства ... 55
5.1.1 ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 ... 55
5.1.2 vESBC ... 55
5.1.3 Описание заводской конфигурации (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350) ... 56
5.2 Подключение и конфигурирование устройства ... 57
5.2.1 Подключение к ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 ... 57
5.2.2 Подключение к vESBC ... 58
5.2.3 Применение изменения конфигурации ... 58
5.2.4 Базовая настройка устройства ... 59
6 Обновление программного обеспечения ... 64
6.1 Создание резервной копии текущей конфигурации ... 65
6.1.1 Подготовка ... 65
6.1.2 Копирование файла резервной копии конфигурации ... 65
6.2 Восстановление конфигурации из резервной копии ... 68
6.2.1 Подготовка ... 68
6.2.2 Копирование файла с резервной копией конфигурации ... 69
6.2.3 Применение и подтверждение загруженной конфигурации ... 71
6.3 Обновление программного обеспечения средствами системы ... 71
6.3.1 Обновление программного обеспечения до версии 1.9.0 при последовательном обновлении с предыдущих версий (только для vESBC) ... 71
6.3.2 Обновление программного обеспечения через CLI ... 72
6.4 Обновление программного обеспечения через web-интерфейс ... 76
6.5 Обновление программного обеспечения с использованием образа ПО .iso (только для vESBC) ... 77
6.6 Обновление программного обеспечения из начального загрузчика (ESBC-15R/31/3200/3250/3350) ... 79
6.7 Обновление вторичного загрузчика (U-Boot) (ESBC-15R/31/3200/3250/3350) ... 80
7 Рекомендации по безопасной настройке ... 82
7.1 Общие рекомендации ... 82
7.2 Настройка системы логирования событий ... 83
7.2.1 Рекомендации ... 83
7.2.2 Предупреждения ... 83

7.2.3 Пример настройки ... 83
7.3 Настройка политики использования паролей ... 84
7.3.1 Рекомендации ... 84
7.3.2 Пример настройки ... 84
7.4 Настройка политики AAA ... 85
7.4.1 Рекомендации ... 85
7.4.2 Предупреждения ... 85
7.4.3 Пример настройки ... 86
7.5 Настройка удалённого управления ... 87
7.5.1 Рекомендации ... 87
7.5.2 Пример настройки ... 87
7.6 Настройка механизмов защиты от сетевых атак ... 88
7.6.1 Рекомендации ... 88
7.6.2 Пример настройки ... 89
8 Примеры подключения ESBC к сети передачи данных ... 90
8.1 Подключение к разным сетям с использованием двух сетевых интерфейсов ... 90
8.2 Подключение к сети с использованием одного сетевого интерфейса ... 91
8.3 Подключение к сети с использованием нескольких сетевых интерфейсов (резервирование линков) ... 91
8.3.1 Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал ... 91
8.3.2 Использование моста (Bridge) для терминации на уровне L3 ... 92
8.4 Подключение к нескольким коммутаторам с использованием нескольких сетевых интерфейсов (резервирование линков и узлов сети) ... 93
8.4.1 Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал ... 93
8.4.2 Использование моста (Bridge) для терминации на уровне L3 ... 93
8.5 Использование кластера ... 93
9 Управление ESBC ... 94
9.1 Общие сведения ... 96
9.2 Настройка абонентских интерфейсов ... 98
9.2.1 Локальная обработка регистрации ... 99
9.2.2 Сброс регистрации абонента на регистраторе ... 126
9.2.3 Инициация перерегистрации абонента через SIP-NOTIFY ... 134
9.3 Настройка SIP-транков ... 143
9.3.1 Динамический режим транка ... 144
9.4 Настройка транковых групп ... 151
9.4.1 Логика работы транковой группы для распределения вызовов на транки, входящие в ее состав ... 154

9.5 Настройка SIP-транспортов ... 157
9.6 Настройка медиаресурсов ... 158
9.7 Использование тегов для маршрутизации и модификации SIP-сообщений ... 160
9.8 Настройка таблиц маршрутизации ... 160
9.8.1 Балансировка правил таблицы маршрутизации ... 162
9.8.2 Код отбоя для правила Reject ... 164
9.8.3 Смена таблиц маршрутизации ... 166
9.8.4 Настройка "закрепления" регистрации ... 172
9.8.5 Использование условий в таблице маршрутизации ... 178
9.9 Настройка модификаторов ... 182
9.9.1 Общие модификаторы ... 183
9.9.2 Модификаторы SIP ... 195
9.10 Настройка SIP-профилей ... 233
9.10.1 Контроль доступности направления ... 234
9.10.2 Список причин отбоя для перехода на следующее направление ... 235
9.10.3 Поведение при перенаправлении ... 237
9.10.4 Игнорирование OPTIONS ... 240
9.10.5 Таймеры SIP-сессий (RFC 4028) ... 241
9.10.6 Транзит сообщений ISUP для работы в режиме SIP-T/SIP-I ... 244
9.10.7 Опционирование обработки заголовка Contact в запросах и ответах SIP ... 244
9.11 Настройка медиапрофилей ... 247
9.11.1 Управление типом медиаданных и кодеками ... 247
9.11.2 Транскодирование ... 254
9.11.3 Таймаут ожидания RTP-пакетов ... 266
9.11.4 Локальная обработка RTCP ... 270
9.11.5 SRTP ... 271
9.11.6 Контроль источника RTP ... 274
9.11.7 Поддержка RFC5168 (PFU) ... 275
9.11.8 Настройка Media Offroad ... 280
9.12 Управление безопасностью системы ... 283
9.12.1 Настройка профилей безопасности ... 283
9.12.2 Общий принцип работы модуля fail2ban ... 283
9.12.3 Фильтрация SIP-флуда ... 284
9.12.4 Блокировка по AOR/User-Agent ... 288
9.12.5 Объединение ошибок по IP-адресу ... 292
9.12.6 Защита от SIP-spoofing атак ... 294
9.12.7 Настройка временных периодов ... 295

9.13 Настройка AAA ... 299
9.13.1 Настройка аутентификации абонентов через RADIUS ... 299
9.13.2 Настройка локальной аутентификации запросов ... 306
9.13.3 Настройка клиентской регистрации транка ... 311
9.14 Настройка NAT ... 318
9.14.1 Настройка NAT comedia-mode ... 318
9.14.2 Настройка Public IP ... 322
9.14.3 Настройка STUN ... 325
9.15 Настройка QoS ... 331
9.16 Контроль трафика ... 332
9.16.1 Контроль входящего трафика ... 332
9.16.2 Контроль исходящего трафика ... 341
9.17 Мониторинг ... 346
9.17.1 Активные вызовы ... 346
9.17.2 Расширенный мониторинг вызовов (call-flow) ... 346
9.17.3 Черный список ... 350
9.17.4 Белый список ... 351
9.17.5 Состояние транков ... 351
9.17.6 Список зарегистрированных абонентов ... 352
9.17.7 Статистика SIP ... 352
9.17.8 Мониторинг параметров трафика (KPI) ... 355
9.17.9 Утилита Sngrep ... 361
9.18 Аварии ... 363
9.18.1 Отправка аварийных SNMP-трапов ... 365
9.19 Настройка CDR ... 368
9.20 Работа с логами ... 370
9.21 Изменение количества модулей ... 374
9.22 Настройка VPN (PPTP и L2TP over IPSec) ... 376
9.22.1 Пример настройки PPTP-сервера для подключения SIP-транков ... 376
9.22.2 Пример настройки PPTP-клиента для подключения SIP-транков ... 381
9.23 Примеры настройки ESBC ... 384
9.23.1 Настройка для SIP-абонентов ... 384
9.23.2 Настройка для SIP-транков ... 387
9.23.3 Настройка для SIP-абонентов, использующих WebRTC ... 390
9.23.4 Пример настройки балансировки сессий для резервирования направлений ... 393
10 Управление интерфейсами ... 402
11 Управление туннелированием ... 402

12 Управление функциями второго уровня (L2) ... 402
13 Управление QoS ... 402
14 Управление маршрутизацией ... 402
15 Управление технологией MPLS ... 402
16 Управление безопасностью ... 403
16.1 Настройка AAA ... 404
16.1.1 Алгоритм настройки локальной аутентификации ... 404
16.1.2 Алгоритм настройки AAA по протоколу RADIUS ... 409
16.1.3 Алгоритм настройки AAA по протоколу TACACS ... 413
16.1.4 Алгоритм настройки AAA по протоколу LDAP ... 418
16.1.5 Пример настройки аутентификации по Telnet через RADIUS-сервер ... 422
16.2 Настройка привилегий команд ... 423
16.2.1 Алгоритм настройки ... 423
16.2.2 Пример настройки привилегий команд ... 424
16.3 Настройка логирования и защиты от ARP-spoofing атак ... 424
16.3.1 Алгоритм настройки ... 424
16.3.2 Пример настройки защиты от ARP-spoofing атак ... 424
16.4 Настройка логирования и защиты от сетевых атак ... 425
16.4.1 Алгоритм настройки ... 425
16.4.2 Описание механизмов защиты от атак ... 428
16.4.3 Пример настройки логирования и защиты от сетевых атак ... 431
16.5 Конфигурирование Firewall ... 433
16.5.1 Алгоритм настройки ... 433
16.5.2 Пример настройки Firewall ... 444
16.5.3 Пример настройки Firewall по доменным именам ... 447
16.5.4 Пример настройки фильтрации приложений (DPI) ... 449
16.5.5 Пример настройки Firewall по расписанию ... 451
16.6 Настройка списков доступа (IP ACL) ... 453
16.6.1 Алгоритм настройки ... 453
16.6.2 Пример настройки списка доступа ... 456
16.7 Настройка списков доступа (MAC ACL) ... 456
16.7.1 Алгоритм настройки ... 457
16.7.2 Пример настройки списка доступа ... 458
16.8 Проксирование HTTP/HTTPS-трафика ... 459
16.8.1 Алгоритм настройки группы вышестоящих серверов ... 460
16.8.2 Алгоритм настройки таблицы модификации ... 462
16.8.3 Алгоритм настройки HTTP/HTTPS-прокси сервера ... 463

16.8.4 Пример настройки HTTP-прокси ... 466
16.9 Настройка IPS/IDS ... 469
16.9.1 Алгоритм базовой настройки ... 469
16.9.2 Алгоритм настройки автообновления правил IPS/IDS из внешних источников ... 472
16.9.3 Рекомендуемые открытые источники обновления правил ... 473
16.9.4 Пример настройки IPS/IDS с автообновлением правил ... 475
16.9.5 Алгоритм настройки базовых пользовательских правил ... 476
16.9.6 Пример настройки базовых пользовательских правил ... 486
16.9.7 Алгоритм настройки расширенных пользовательских правил ... 488
16.9.8 Пример настройки расширенных пользовательских правил ... 489
17 Управление сертификатами и ключами ... 490
17.1 Настройка криптопрофилей ... 490
17.2 Автоматическое распространение ключей и сертификатов X.509 ... 495
17.2.1 Общее описание инфраструктуры открытых ключей ... 495
17.2.2 Планирование инфраструктуры открытых ключей ... 496
17.2.3 Настройка PKI-сервера в роли корневого удостоверяющего центра ... 497
17.2.4 Настройка PKI-клиента ... 504
17.2.5 Процесс автоматического перевыпуска сертификата PKI-клиента ... 513
17.2.6 Процесс автоматического перевыпуска сертификата PKI-сервера ... 515
17.3 Ручная генерация и распространение ключей и сертификатов X.509 ... 519
17.3.1 Алгоритм генерации ключей и запросов на сертификацию ... 519
17.3.2 Пример ручного выпуска сертификата через внешний удостоверяющий центр ... 523
18 Управление резервированием ... 528
19 Управление кластеризацией ... 529
19.1 Настройка кластера на ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 ... 530
19.1.1 Первичная настройка кластера ... 530
19.1.2 Настройка внешних сетевых интерфейсов ... 532
19.1.3 Настройка кластерного интерфейса ... 533
19.1.4 Настройка кластера ... 530
19.1.5 Настройка синхронизации сертификатов и ключей ... 536
19.2 Настройка кластера на vESBC ... 539
19.2.1 Пример настройки кластера vESBC в гипервизоре VirtualBox ... 539
19.2.2 Пример настройки кластера vESBC в гипервизоре QEMU/KVM ... 541
19.2.3 Пример настройки кластера vESBC в гипервизоре XEN ... 543
19.2.4 Пример настройки кластера vESBC в гипервизоре XCP-ng ... 545
19.2.5 Особенности настройки гипервизора ESXi для организации кластера vESBC ... 546
19.2.6 Пример настройки кластера vESBC в гипервизоре zVirt ... 547

19.2.7 Подключение второго юнита в кластере vESBC с использованием ZTP 551
20 Управление удаленным доступом 553
21 Управление сервисами 553
22 Мониторинг 553
23 Управление лицензированием 553
23.1 Виды лицензий ESBC 554
23.1.1 vESBC 554
23.1.2 ESBC-15R/31/3200/3250/3350 554
23.2 Способы получения лицензии 555
23.3 Статусы лицензий 555
23.4 Лицензирование через ELM 555
23.4.1 Периодичность обращений к ELM 555
23.4.2 Условия преждевременного сброса лицензии 556
23.4.3 Алгоритм настройки 556
23.4.4 Пример настройки 558
23.5 Загрузка и активация файловой лицензии (ESBC-15R/31/3200/3250/3350) 560
23.6 Лицензирование в кластере 561
23.6.1 Синхронизация файловых лицензий 561
23.6.2 Установка файловых лицензий 561
24 Часто задаваемые вопросы 562
25 Приложение A. Packet Flow 564
25.1 Порядок обработки входящего/исходящего трафика сетевыми службами пограничного контроллера сессий ESBC 564
25.2 Порядок обработки транзитного трафика сетевыми службами пограничного контроллера сессий ESBC 566

1 Введение

- Аннотация
- Целевая аудитория
- Условные обозначения
- Примечания и предупреждения

1.1 Аннотация

Производительность, надёжность и безопасность — ключевые приоритеты при организации VoIP-телефонии в корпоративной сети. Необходимо обеспечить не только совместимость оборудования на всех уровнях и его отлаженную работу, но и защиту от различных атак. Игнорирование последнего приводит к взлому VoIP-сети злоумышленниками.

Пограничный контроллер сессий (ESBC) поможет избежать этих проблем. Он используется для сокрытия топологии VoIP-сети, защиты от несанкционированного доступа, а также управления трафиком.

В данном руководстве по эксплуатации изложены назначение, технические характеристики, конструктивное исполнение, порядок установки, рекомендации по начальной настройке и обновлению программного обеспечения пограничного контроллера сессий ESBC (далее ESBC или устройство).

1.2 Целевая аудитория

Данное руководство пользователя предназначено для технического персонала, выполняющего установку, настройку и мониторинг устройств посредством интерфейса командной строки (CLI), а также процедуры по обслуживанию системы и обновлению ПО. Квалификация технического персонала предполагает знание основ работы стеков протоколов TCP/IP, принципов построения Ethernet-сетей.

1.3 Условные обозначения

Обозначение	Описание
[ ]	В квадратных скобках в командной строке указываются необязательные параметры, но их ввод предоставляет определенные дополнительные опции.
{ }	В фигурных скобках в командной строке указываются возможные обязательные параметры. Необходимо выбрать один из параметров.
«,» **«-»**	Данные знаки в описании команды используются для указания диапазонов.
«|»	Данный знак в описании команды обозначает «или».
<Полужирный курсив>	Полужирным курсивом в угловых скобках указываются названия клавиш на клавиатуре.

Обозначение	Описание
`Текст в рамке`	В рамках с текстом указаны примеры и результаты выполнения команд. В примерах могут встречаться названия vesbc, esbc, esr. Команды, приведенные в таких примерах, применимы для ESBC-15R/31/3200/3250/3350 и vESBC.

1.4 Примечания и предупреждения

Примечания содержат важную информацию, советы или рекомендации по использованию и настройке устройства.

Предупреждения информируют пользователя о ситуациях, которые могут нанести вред устройству или человеку, привести к некорректной работе устройства или потере данных.

Информация содержит справочные данные об использовании устройства.

2 Описание изделий

- Назначение
- Функции
  - Функции интерфейсов
  - Функции при работе с MAC-адресами
  - Функции второго уровня сетевой модели OSI
  - Функции третьего уровня сетевой модели OSI
  - Функции туннелирования трафика
  - Функции управления и конфигурирования
  - Функции сетевой защиты
- Основные технические характеристики
- Конструктивное исполнение
  - Конструктивное исполнение ESBC-15R
  - Конструктивное исполнение ESBC-31
  - Конструктивное исполнение ESBC-3200
  - Конструктивное исполнение ESBC-3250
  - Конструктивное исполнение ESBC-3350
  - Световая индикация
- Комплект поставки
- Пусковые токи ESBC-15R, ESBC-31, ESBC-3200

2.1 Назначение

Пограничный контроллер сессий ESBC предназначен для решения задач сопряжения разнородных VoIP-сетей, обеспечивая совместную работу терминалов с различными протоколами сигнализации и наборами используемых кодеков. Кроме того, за счет функциональности Firewall, NAT и проксирования сигнального и медиатрафика он защищает корпоративную сеть от атак и скрывает ее внутреннюю структуру. ESBC всегда устанавливается на границе корпоративной или операторской VoIP-сети и выполняет те функции, которые нецелесообразно возлагать на устройства оператора (например, гибкий коммутатор Softswitch).

Устройства серии ESBC являются высокопроизводительными многоцелевыми сетевыми устройствами, которые объединяют в себе традиционные сетевые функции и комплексный многоуровневый подход к безопасности маршрутизации, что позволяет обеспечить надежную защиту для корпоративной среды.

Устройства поддерживают функции межсетевого экрана для защиты сети организации и своей сетевой инфраструктуры, а также сочетают в себе новейшие средства обеспечения безопасности данных, шифрования, аутентификации и защиты от вторжений.

Устройства содержат в себе средства для программной и аппаратной обработки данных. За счет оптимального распределения функций обработки данных между частями устройства достигается максимальная производительность.

2.2 Функции

2.2.1 Функции интерфейсов

В таблице 1 приведен список функций интерфейсов устройства.

Таблица 1 – Функции интерфейсов устройства

Определение полярности подключения кабеля (Auto MDI/MDIX)	Автоматическое определение типа кабеля – перекрестный кабель или кабель прямого подключения. • MDI (Medium Dependent Interface – прямой) – стандарт кабелей для подключения оконечных устройств; • MDIX (Medium Dependent Interface with Crossover – перекрестный) – стандарт кабелей для подключения концентраторов и коммутаторов.
Поддержка обратного давления (Back pressure)	Метод обратного давления используется на полудуплексных соединениях для регулирования потока данных от встречного устройства путем создания коллизий. Метод позволяет избежать переполнения буферной памяти устройства и потери данных.
Агрегирование каналов (LAG, Link aggregation)	Агрегирование (объединение) каналов позволяет увеличить пропускную способность канала связи и повысить его надежность. Пограничный контроллер сессий поддерживает статическое и динамическое агрегирование каналов. При динамическом агрегировании используется протокол LACP для управления группой каналов.

2.2.2 Функции при работе с MAC-адресами

В таблице 2 приведены функции устройства при работе с MAC-адресами.

Таблица 2 – Функции работы с MAC-адресами

Таблица MAC-адресов	Таблица MAC-адресов устанавливает соответствие между MAC-адресами и интерфейсами устройства и используется для маршрутизации пакетов данных. Пограничные контроллеры сессий имеют таблицу емкостью до 128k MAC-адресов и резервируют определенные MAC-адреса для использования системой.
Режим обучения	MAC-таблица может содержать либо статические адреса, либо адреса, изученные при прохождении пакетов данных через устройство. Изучение происходит за счет регистрации MAC-адресов отправителей пакетов с привязкой их к портам и VLAN. Впоследствии эти данные используются для маршрутизации встречных пакетов. Время хранения зарегистрированных MAC-адресов ограничено, его продолжительность может настраиваться администратором. Если MAC-адрес получателя, указанный в принятом устройством пакете, отсутствует в таблице, то такой пакет отправляется далее как широковещательный в пределах L2-сегмента сети.

2.2.3 Функции второго уровня сетевой модели OSI

В таблице 3 приведены функции и особенности второго уровня (уровень 2 OSI).

Таблица 3 – Описание функций второго уровня (уровень 2 OSI)

Поддержка VLAN	VLAN (Virtual Local Area Network) – это средство разделения сети на изолированные сегменты на уровне L2. Использование VLAN позволяет повысить устойчивость работы крупных сетей за счет деления их на более мелкие сети, изолировать разнородный трафик данных между собой и решить многие другие задачи. Пограничные контроллеры сессий поддерживают различные способы организации VLAN: • VLAN на базе меток пакетов данных, в соответствии с IEEE 802.1Q; • VLAN на базе портов устройства (port-based); • VLAN на базе использования правил классификации данных (policy-based).
Протокол связующего дерева (Spanning Tree Protocol)	Задачей протокола Spanning Tree является исключение избыточных сетевых соединений и приведение топологии сети к древовидной. Основные применения протокола связаны с предотвращением зацикливания сетевого трафика и с организацией резервных каналов связи.

2.2.4 Функции третьего уровня сетевой модели OSI

В таблице 4 приведены функции третьего уровня (уровень 3 OSI).

Таблица 4 – Описание функций третьего уровня (Layer 3)

Статические IP-маршруты	Администратор пограничного контроллера сессий имеет возможность добавлять и удалять статические записи в таблицу маршрутизации.
Динамическая маршрутизация	Протоколы динамической маршрутизации позволяют устройству обмениваться маршрутной информацией с соседними пограничными контроллерами сессий и автоматически составлять таблицу маршрутов. Пограничный контроллер сессий поддерживает следующие протоколы: RIPv2, RIPng, OSPFv2, OSPFv3, IS-IS, BGP.
Таблица ARP	ARP (Address Resolution Protocol) – протокол для выяснения соответствия адресов сетевого и канального уровней. Таблица ARP содержит информацию об изученном соответствии. Соответствие устанавливается на основе анализа ответов от сетевых устройств, адреса устройств запрашиваются с помощью широковещательных пакетов.
Клиент DHCP	Протокол DHCP (Dynamic Host Configuration Protocol) даёт возможность автоматизировать управление сетевыми устройствами. Клиент DHCP позволяет пограничному контроллеру сессий получать сетевой адрес и дополнительные параметры от внешнего DHCP-сервера. Как правило, этот способ используется для получения сетевых настроек оператора публичной сети (WAN).

Сервер DHCP	Сервер DHCP предназначен для автоматизации и централизации конфигурирования сетевых устройств. Размещение DHCP-сервера на пограничном контроллере сессий позволяет получить законченное решение для поддержки локальной сети. DHCP-сервер, входящий в состав пограничного контроллера сессий, позволяет назначать IP-адреса сетевым устройствам и передавать дополнительные сетевые параметры – адреса серверов, адреса шлюзов сети и другие необходимые параметры.
DHCP Relay	Функция DHCP Relay предназначена для перенаправления широковещательных DHCP Discover-пакетов из одного широковещательного домена в одноадресные (unicast) DHCP Discover-пакеты в другом широковещательном домене.
Трансляция сетевых адресов (NAT, Network Address Translation)	Трансляция сетевых адресов – это механизм, который позволяет преобразовывать IP-адреса и номера портов транзитных пакетов. Функция NAT позволяет использовать меньшее количество IP-адресов, транслируя несколько IP-адресов внутренней сети в один внешний публичный IP-адрес. Использование NAT позволяет увеличить защищённость локальной сети за счёт скрытия её внутренней структуры. Пограничные контроллеры сессий поддерживают следующие варианты NAT: • Source NAT (SNAT) – выполняется замена адреса, а также номера порта источника при прохождении пакета в одну сторону и обратной замене адреса назначения в ответном пакете; • Destination NAT (DNAT) – когда обращения извне транслируются пограничным контроллером сессий на компьютер пользователя в локальной сети, имеющий внутренний адрес и потому недоступный извне сети непосредственно (без NAT).

2.2.5 Функции туннелирования трафика

Таблица 5 – Функции туннелирования трафика

Протоколы туннелирования	Туннелирование – это способ преобразования пакетов данных при передаче их по сети, при котором происходит замена, модификация или добавление нового сетевого заголовка пакета. Такой способ может быть использован для согласования транспортных протоколов при прохождении данных через транзитную сеть, для создания защищенных соединений, при которых туннелированные данные подвергаются шифрованию. Пограничные контроллеры сессий поддерживают следующие виды туннелей: • GRE – инкапсуляция IP-пакета/Ethernet-кадра в другой IP-пакет с добавлением GRE (General Routing Encapsulation)-заголовка; • IPv4-IPv4 – туннель, использующий инкапсуляцию исходных IP-пакетов в IP-пакеты с другими сетевыми параметрами; • L2TPv3 – туннель для передачи L2-трафика с помощью IP-пакетов; • IPsec – туннель с шифрованием передаваемых данных; • L2TP, PPTP, PPPoE, OpenVPN, WireGuard – туннели, использующиеся для организации удаленного доступа клиент-сервер.

2.2.6 Функции управления и конфигурирования

Таблица 6 – Основные функции управления и конфигурирования

Загрузка и выгрузка файла настройки	Параметры устройства сохраняются в файле настройки, который содержит данные конфигурации как всей системы в целом, так и определенного порта устройства. Для передачи файлов могут использоваться протоколы TFTP, FTP, SCP.
Интерфейс командной строки (CLI)	Управление посредством CLI осуществляется локально через последовательный порт RS-232 либо удаленно через Telnet, SSH. Интерфейс командной строки консоли (CLI) является промышленным стандартом. Интерпретатор CLI предоставляет список команд и ключевых слов для помощи пользователю и сокращению объема вводимых данных.
Syslog	Протокол Syslog обеспечивает передачу информационных сообщений о происходящих в системе событиях и ведение журнала событий.
Сетевые утилиты ping, traceroute	Утилиты ping и traceroute – предназначены для проверки доступности сетевых устройств и для определения маршрутов передачи данных в IP-сетях.
Управление контролируемым доступом – уровни привилегий	Пограничные контроллеры сессий поддерживают управление уровнем доступа пользователей к системе. Уровни доступа позволяют управлять зонами ответственности администраторов устройств. Уровни доступа нумеруются от 1 до 15, уровень 15 соответствует полному доступу к управлению устройством.
Аутентификация	Аутентификация – это процедура проверки подлинности пользователя. Пограничные контроллеры сессий поддерживают следующие методы аутентификации: • локальная – для аутентификации используется локальная база данных пользователей, хранящаяся на самом устройстве; • групповая – база данных пользователей хранится на сервере аутентификации. Для взаимодействия с сервером используются протоколы RADIUS и TACACS.
Сервер SSH/ сервер Telnet	Функции сервера SSH и Telnet позволяют установить соединение с устройством для управления им.
Автоматическое восстановление конфигурации	Устройство поддерживает автоматическую систему восстановления конфигурации, которая предотвращает ситуации потери удаленного доступа к устройству после смены конфигурации. Если в течение заданного времени после изменения конфигурации не было введено подтверждение – произойдет автоматический откат конфигурации до предыдущего использовавшегося состояния.

2.2.7 Функции сетевой защиты

В таблице 7 приведены функции сетевой защиты, выполняемые устройством.

Таблица 7 − Функции сетевой защиты

Зоны безопасности	Все интерфейсы пограничного контроллера сессий распределяются по зонам безопасности. Для каждой пары зон настраиваются правила, определяющие возможность или невозможность прохождения данных между зонами, правила фильтрации трафика данных.
Фильтрация данных	Для каждой пары зон безопасности составляется набор правил, которые позволяют управлять фильтрацией данных, проходящих через пограничные контроллеры сессий. Командный интерфейс устройства предоставляет средства для детальной настройки правил классификации трафика и для назначения результирующего решения о пропуске трафика.

2.3 Основные технические характеристики

Основные технические параметры аппаратных пограничных контроллеров сессий ESBC приведены в таблице 8.

Таблица 8 – Основные технические характеристики

Общие параметры		
Интерфейсы	ESBC-15R	4 × Ethernet 10/100/1000BASE-T (RJ-45) 2 × 1000BASE-X (SFP) 1 × Консольный порт RS-232 (RJ-45) 2 × USB 2.0
	ESBC-31	8 × Ethernet 10/100/1000BASE-T (RJ-45) 6 × Ethernet 1000BASE-X (SFP) 2 × 10GBASE-R (SFP+)/1000BASE-X (SFP) 3 × Последовательный порт RS-232 1 × Консольный порт RS-232 (RJ-45) 1 × USB 3.0 1 × USB 2.0 1 × Слот для microSD-карты
	ESBC-3200	12 × 1000BASE-X/10GBASE-R/25GBASE-R 1 × Консольный порт RS-232 (RJ-45) 1 × Порт OOB 1 × USB 2.0 1 × Слот для microSD-карты
	ESBC-3250	8 × Combo Ethernet 10/100/1000BASE-T/1000BASE-X (SFP) 4 × 1000BASE-X (SFP)/10GBASE-R (SFP+)/25GBASE-R (SFP28) 1 × Консольный порт RS-232 (RJ-45) 2 × USB 3.0 1 × Слот для microSD-карты
	ESBC-3350	8 × Combo Ethernet 10/100/1000BASE-T/1000BASE-X (SFP) 4 × 1000BASE-X (SFP)/10GBASE-R (SFP+)/25GBASE-R (SFP28) 1 × Консольный порт RS-232 (RJ-45) 2 × USB 3.0 1 × Слот для microSD-карты

Типы оптических трансиверов	ESBC-15R	1000BASE-X SFP
	ESBC-31	1000BASE-X SFP 10GBASE-R SFP+
	ESBC-3200 ESBC-3250 ESBC-3350	25GBASE-R SFP28 10GBASE-R SFP+ 1000BASE-X SFP
Дуплексный и полудуплексный режимы интерфейсов		• дуплексный и полудуплексный режимы для электрических портов • дуплексный режим для оптических портов
Скорость передачи данных	ESBC-15R	• электрические интерфейсы 10/100/1000 Мбит/с • оптические интерфейсы 1 Гбит/с
	ESBC-31	• электрические интерфейсы 10/100/1000 Мбит/с • оптические интерфейсы 1/10 Гбит/с
	ESBC-3200	• оптические интерфейсы 1/10/25 Гбит/с
	ESBC-3250 ESBC-3350	• электрические интерфейсы 10/100/1000 Мбит/с • оптические интерфейсы 1/10/25 Гбит/с
Количество VPN-туннелей	ESBC-15R	10
	ESBC-31	250
	ESBC-3200 ESBC-3250 ESBC-3350	500
Количество статических маршрутов	ESBC-15R	1k
	ESBC-31 ESBC-3200 ESBC-3250 ESBC-3350	11k
Максимальное количество конкурентных сессий	ESBC-15R	300k
	ESBC-31	3,26M
	ESBC-3200 ESBC-3250 ESBC-3350	8,5M

Таблица VLAN		4094
Количество маршрутов BGPv4/BGPv6	ESBC-15R	1M
	ESBC-31	2,5M
	ESBC-3200 ESBC-3250 ESBC-3350	5M
Количество маршрутов OSPFv2/OSPFv3/IS-IS	ESBC-15R	30k
	ESBC-31	300k
	ESBC-3200 ESBC-3250 ESBC-3350	500k
Количество маршрутов RIP/RIPng	ESBC-15R	1k
	ESBC-31 ESBC-3200 ESBC-3250 ESBC-3350	10k
Размер базы FIB	ESBC-15R	1M
	ESBC-31	1,4M
	ESBC-3350 ESBC-3250 ESBC-3200	1,7M
VRF		32
Количество L3-интерфейсов	ESBC-15R	200
	ESBC-31 ESBC-3200 ESBC-3250 ESBC-3350	4000

Соответствие стандартам		IEEE 802.3 10BASE-T Ethernet IEEE 802.3u 100BASE-T Fast Ethernet IEEE 802.3ab 1000BASE-T Gigabit Ethernet IEEE 802.3z Fiber Gigabit Ethernet IEEE 802.3cc 25GBASE-LR Ethernet IEEE 802.3by 25GBASE-SR Ethernet ANSI/IEEE 802.3 автоопределение скорости IEEE 802.3x контроль потоков данных IEEE 802.3ad объединение каналов LACP IEEE 802.1Q виртуальные локальные сети VLAN IEEE 802.1v, IEEE 802.3ac, IEEE 802.3ae, IEEE 802.1D, IEEE 802.1w, IEEE 802.1s
Управление		
Локальное управление		CLI
Удаленное управление		Telnet, SSH
Физические характеристики и условия окружающей среды		
Источники питания	ESBC-15R	Сеть переменного тока: 100–264 В, 50–60 Гц
	ESBC-31 ESBC-3200 ESBC-3250	Сеть переменного тока: 100–240 В, 50–60 Гц Сеть постоянного тока: 36–72 В Варианты питания: • один источник питания постоянного или переменного тока; • два источника питания постоянного или переменного тока с возможностью горячей замены.
	ESBC-3350	Сеть переменного тока: 200–240 В, 50–60 Гц Сеть постоянного тока: 36–72 В Варианты питания: • один источник питания постоянного или переменного тока; • два источника питания постоянного или переменного тока с возможностью горячей замены.
Максимальная потребляемая мощность	ESBC-15R	18 Вт
	ESBC-31	44 Вт
	ESBC-3200	117 Вт
	ESBC-3250	74 Вт

	ESBC-3350	110 Вт
Масса	ESBC-15R	1,405 кг
	ESBC-31	3,425 кг
	ESBC-3200	5 кг
	ESBC-3250	4,232 кг
	ESBC-3350	5,638 кг
Габаритные размеры (Ш × В × Г)	ESBC-15R	267 × 44 × 160 мм
	ESBC-31	430 × 44 × 275 мм
	ESBC-3200	430 × 44 × 330 мм
	ESBC-3250	430 × 43,6 × 329 мм
	ESBC-3350	430 × 43,6 × 424 мм
Интервал рабочих температур	ESBC-15R	от -0 до +40 °C
	ESBC-31 ESBC-3200 ESBC-3250 ESBC-3350	от -10 до +45 °C
Интервал температуры хранения		от -40 до +70 °C
Относительная влажность при эксплуатации (без образования конденсата)		не более 80 %
Относительная влажность при хранении (без образования конденсата)		от 10 до 95 %
Срок службы		не менее 15 лет

2.4 Конструктивное исполнение

В данном разделе описано конструктивное исполнение устройства. Представлены изображения передней, задней и боковых панелей устройства. Описаны разъемы, светодиодные индикаторы и органы управления.

2.4.1 Конструктивное исполнение ESBC-15R

Устройство выполнено в металлическом корпусе с возможностью установки в 19” конструктив, высота корпуса 1U.

Передняя панель устройства ESBC-15R

Внешний вид передней панели ESBC-15R показан на рисунке 1.

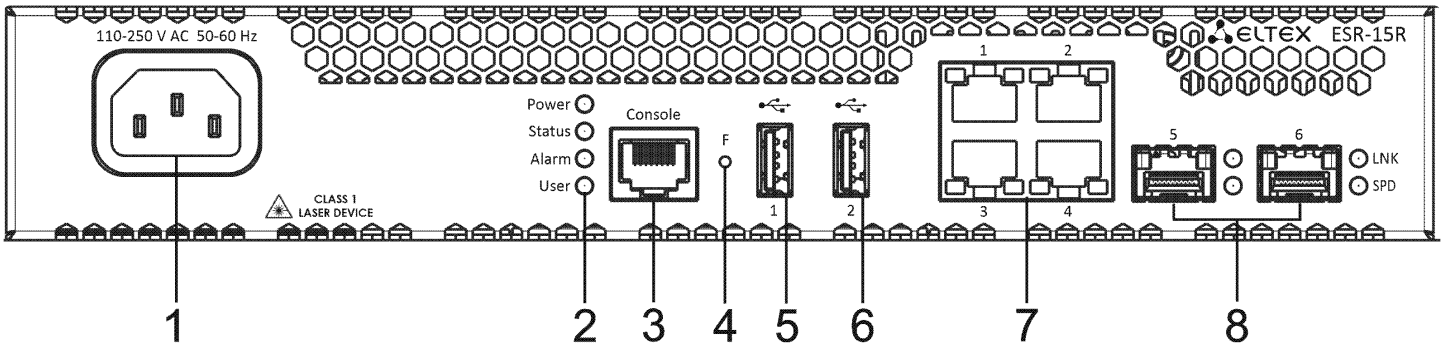


Рисунок 1 – Передняя панель ESBC-15R

В таблице 9 приведен перечень разъемов, светодиодных индикаторов и органов управления, расположенных на передней панели ESBC-15R.

Таблица 9 – Описание разъемов, индикаторов и органов управления передней панели ESBC-15R

№	Элемент передней панели	Описание
1	110-250V AC	Источник питания.
2	Power	Индикатор питания устройства.
	Status	Индикатор текущего состояния устройства.
	Alarm	Индикатор наличия и уровня аварии устройства.
	User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.
3	Console	Консольный порт RS-232 (RJ-45) для локального управления устройством.
4	F	Функциональная кнопка для перезагрузки устройства и сброса к заводским настройкам: • при длительности нажатия на кнопку менее 10 секунд происходит перезагрузка устройства; • при длительности нажатия на кнопку более 10 секунд происходит сброс устройства к заводской конфигурации.
5	USB1	Разъем для подключения внешних USB-устройств.
6	USB2	Разъем для подключения внешних USB-устройств.
7	[1 .. 4]	4 порта Gigabit Ethernet 10/100/1000BASE-T (RJ-45).
8	5, 6	2 порта Gigabit Ethernet 1000BASE-X SFP.

Задняя панель устройств ESBC-15R

Внешний вид задней панели ESBC-15R показан на рисунке 2.

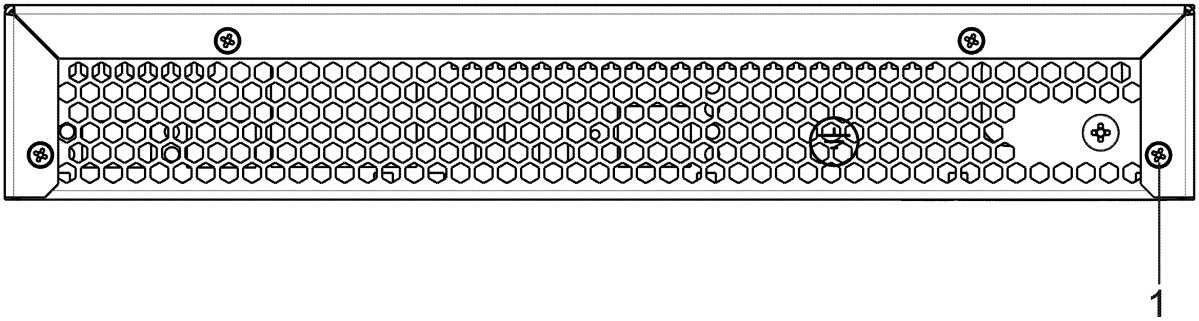


Рисунок 2 – Задняя панель ESBC-15R

В таблице 10 приведен перечень разъемов, расположенных на задней панели ESBC-15R.

Таблица 10 – Описание разъемов задней панели ESBC-15R

№	Описание
1	Клемма для заземления устройства.

Боковые панели устройства ESBC-15R

Внешний вид боковых панелей ESBC-15R приведен на рисунках ниже.

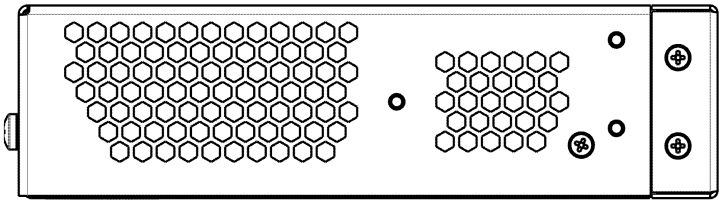


Рисунок 3 – Левая боковая панель ESBC-15R

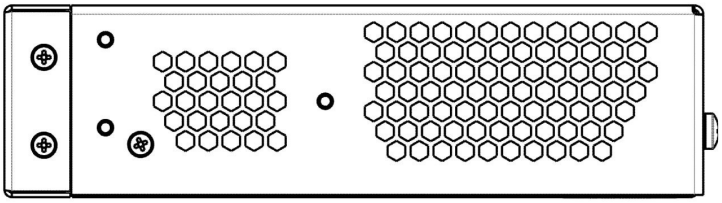


Рисунок 4 – Правая боковая панель ESBC-15R

На боковых панелях устройства расположены вентиляционные решетки, которые служат для отвода тепла. Не закрывайте вентиляционные отверстия посторонними предметами. Это может привести к перегреву компонентов устройства и вызвать нарушения в его работе. Рекомендации по установке устройства расположены в разделе Установка и подключение.

2.4.2 Конструктивное исполнение ESBC-31

Устройство выполнено в металлическом корпусе с возможностью установки в 19” конструктив, высота корпуса 1U.

Передняя панель устройства ESBC-31

Внешний вид передней панели ESBC-31 показан на рисунке 5.

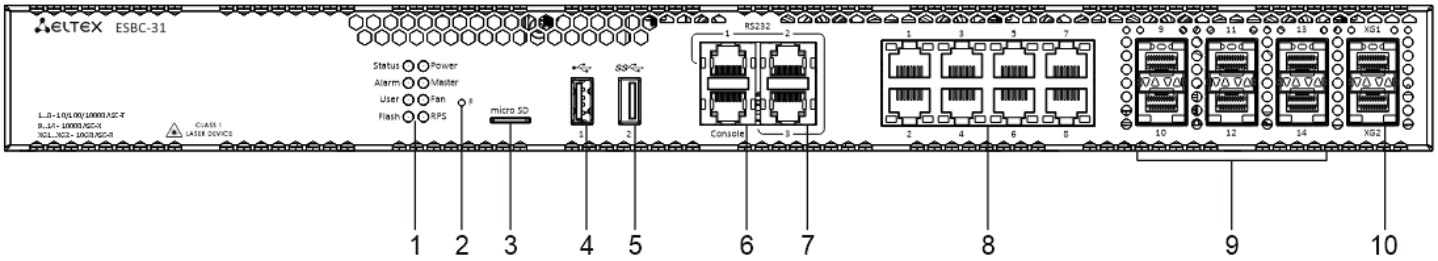


Рисунок 5 – Передняя панель ESBC-31

В таблице 11 приведен перечень разъемов, светодиодных индикаторов и органов управления, расположенных на передней панели ESBC-31.

Таблица 11 – Описание разъемов, индикаторов и органов управления передней панели ESBC-31

№	Элемент передней панели	Описание
1	Status	Индикатор текущего состояния устройства.
	Alarm	Индикатор наличия и уровня аварии устройства.
	User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.
	Flash	Индикатор активности обмена с накопителем данных – microSD-картой или USB Flash.
	Power	Индикатор питания устройства.
	Master	Индикатор работы устройства в failover-режимах (не используется в текущей версии).
	Fan	Индикатор аварии вентиляторов.
	RPS	Индикатор резервного источника электропитания.
2	F	Функциональная кнопка для перезагрузки устройства и сброса к заводским настройкам: • при удержании кнопки менее 10 секунд происходит перезагрузка устройства; • при удержании кнопки более 10 секунд происходит перезагрузка устройства и сброс к заводским настройкам.
3	microSD	Разъем для установки microSD-карты памяти.
4	USB1	Порт USB 2.0 для подключения USB-устройств.
5	USB2	Порт USB 3.0 для подключения USB-устройств.

№	Элемент передней панели	Описание
6	Console	Консольный порт RS-232 (RJ-45) для локального управления устройством.
7	RS-232	3 последовательных порта.
8	[1 .. 8]	8 портов Gigabit Ethernet 10/100/1000BASE-T (RJ-45).
9	Optical Port	6 портов Gigabit Ethernet 10/100/1000BASE-X (SFP).
10	XG1 – XG2	Слоты для установки трансиверов 10G SFP+/1G SFP.

Задняя панель устройств ESBC-31

Внешний вид задней панели ESBC-31 показан на рисунке 6.

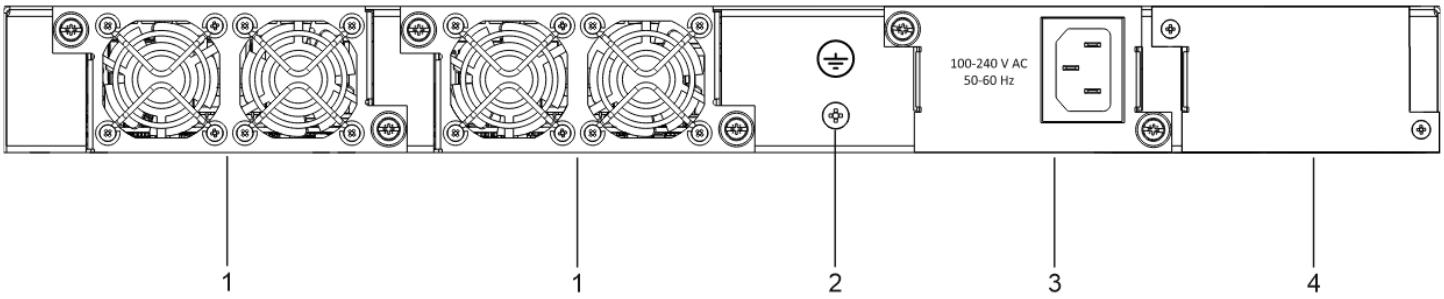


Рисунок 6 – Задняя панель ESBC-31

В таблице 12 приведен перечень разъемов, расположенных на задней панели ESBC-31.

Таблица 12 – Описание разъемов задней панели ESBC-31

№	Описание
1	Съемные вентиляционные модули с возможностью горячей замены.
2	Клемма для заземления устройства.
3	Основной источник питания.
4	Место для установки резервного источника питания.

Боковые панели устройства ESBC-31

Внешний вид боковых панелей ESBC-31 приведен на рисунках ниже.

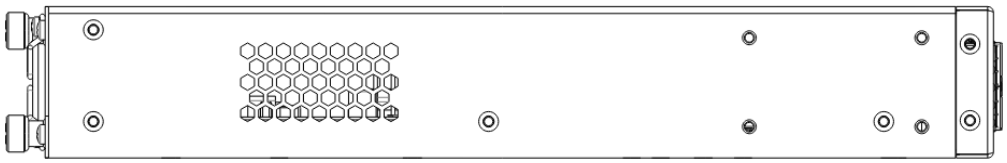


Рисунок 7 – Левая панель ESBC-31

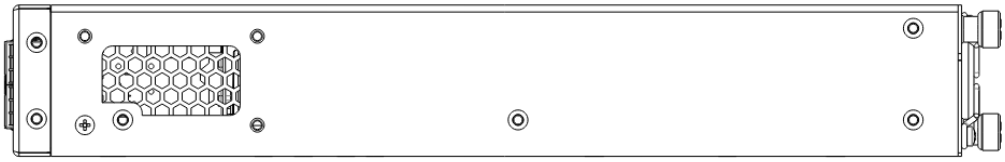


Рисунок 8 – Правая панель ESBC-31

На боковых панелях устройства расположены вентиляционные решетки, которые служат для отвода тепла. Не закрывайте вентиляционные отверстия посторонними предметами. Это может привести к перегреву компонентов устройства и вызвать нарушения в его работе. Рекомендации по установке устройства расположены в разделе Установка и подключение.

2.4.3 Конструктивное исполнение ESBC-3200

Устройство выполнено в металлическом корпусе с возможностью установки в 19” конструктив, высота корпуса 1U.

Передняя панель устройства ESBC-3200

Внешний вид передней панели ESBC-3200 показан на рисунке 9.

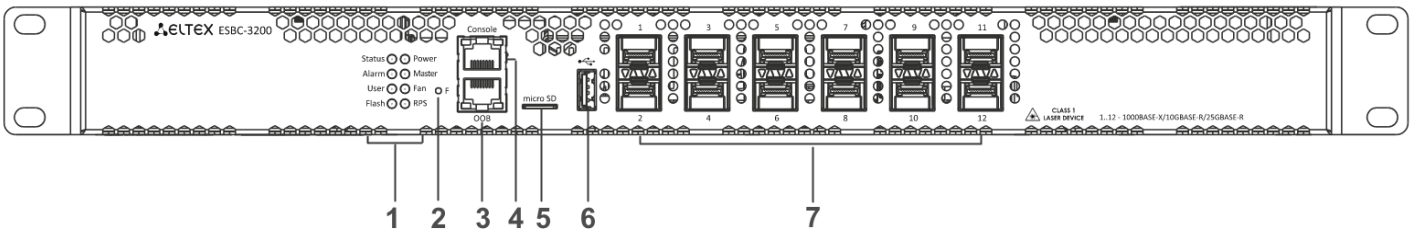


Рисунок 9 – Передняя панель ESBC-3200

Таблица 13 – Описание разъемов, индикаторов и органов управления передней панели ESBC-3200

№	Элемент передней панели	Описание
1	Status	Индикатор текущего состояния устройства.
	Alarm	Индикатор наличия и уровня аварии устройства.
	User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.
	Flash	Индикатор активности обмена с накопителем данных – microSD-картой или USB Flash.
	Power	Индикатор питания устройства.
	Master	Индикатор работы устройства в failover-режимах (не используется в текущей версии).
	Fan	Индикатор аварии вентиляторов.
	RPS	Индикатор резервного источника электропитания.

№	Элемент передней панели	Описание
2	F	Функциональная кнопка для перезагрузки устройства и сброса к заводским настройкам: • при удержании кнопки менее 10 секунд происходит перезагрузка устройства; • при удержании кнопки более 10 секунд происходит перезагрузка устройства и сброс к заводским настройкам.
3	OOB	Ethernet-порт используется только для обновления программного обеспечения через вторичный загрузчик U-Boot. Данный порт не используется для удаленного доступа и управления. Данный интерфейс не может участвовать в маршрутизации транзитного трафика.
4	Console	Консольный порт RS-232 (RJ-45) для локального управления устройством.
5	microSD	Разъем для установки microSD-карт памяти.
6	USB1	Порт USB 2.0 для подключения USB-устройств.
7	[1 .. 12]	Слоты для установки трансиверов 25G SFP28/10G SFP+/1G SFP.

Задняя панель устройства ESBC-3200

Внешний вид задней панели ESBC-3200 приведен на рисунке 10.

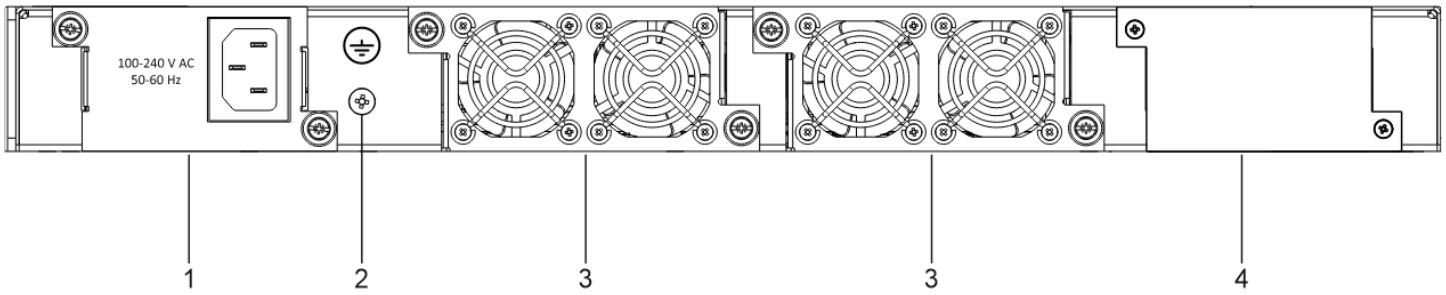


Рисунок 10 – Задняя панель ESBC-3200

Таблица 14 – Описание разъемов задней панели ESBC-3200

№	Описание
1	Основной источник питания.
2	Клемма для заземления устройства.
3	Съемные вентиляционные модули с возможностью горячей замены.
4	Место для установки резервного источника питания.

Боковые панели устройства ESBC-3200

Внешний вид боковых панелей приведен на рисунках ниже.

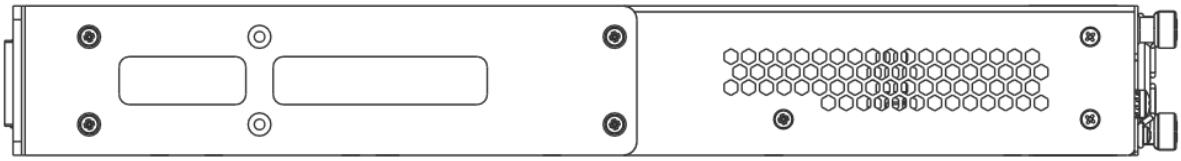


Рисунок 11 – Правая боковая панель ESBC-3200

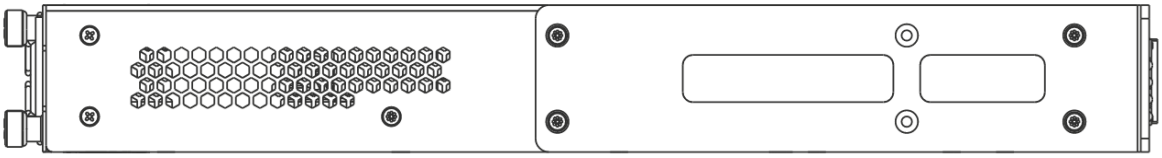


Рисунок 12 – Левая боковая панель ESBC-3200

На боковых панелях устройства расположены вентиляционные решетки, которые служат для отвода тепла. Не закрывайте вентиляционные отверстия посторонними предметами. Это может привести к перегреву компонентов устройства и вызвать нарушения в его работе. Рекомендации по установке устройства расположены в разделе Установка и подключение.

2.4.4 Конструктивное исполнение ESBC-3250

Устройство выполнено в металлическом корпусе с возможностью установки в 19” конструктив, высота корпуса 1U.

Передняя панель устройства ESBC-3250

Внешний вид передней панели ESBC-3250 показан на рисунке 13.

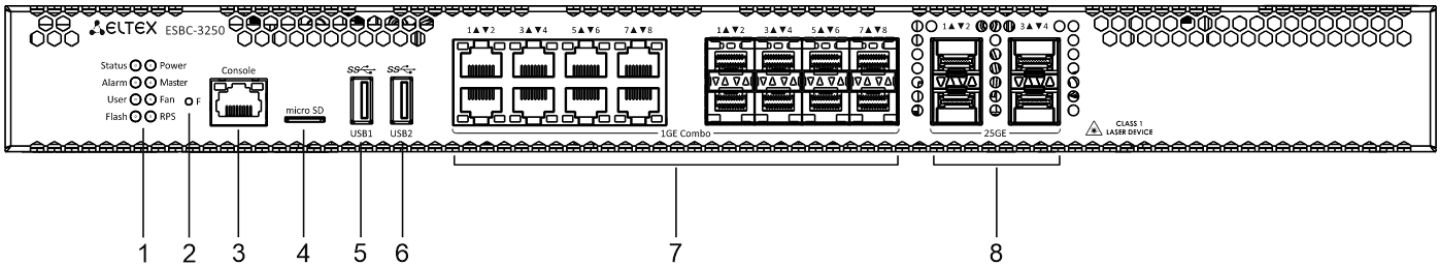


Рисунок 13 – Передняя панель ESBC-3250

В таблице 15 приведен перечень разъемов, светодиодных индикаторов и органов управления, расположенных на передней панели ESBC-3250.

Таблица 15 – Описание разъемов, индикаторов и органов управления передней панели ESBC-3250

№	Элемент передней панели	Описание
1	Status	Индикатор текущего состояния устройства.
	Alarm	Индикатор наличия и уровня аварии устройства.
	User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.
	Flash	Индикатор активности обмена с накопителем данных – microSD-картой или USB Flash.
	Power	Индикатор питания устройства.
	Master	Индикатор работы устройства в failover-режимах (не используется в текущей версии).
	Fan	Индикатор аварии вентиляторов.
	RPS	Индикатор резервного источника электропитания.
2	F	Функциональная кнопка для перезагрузки устройства и сброса к заводским настройкам: • при удержании кнопки менее 10 секунд происходит перезагрузка устройства; • при удержании кнопки более 10 секунд происходит перезагрузка устройства и сброс к заводским настройкам.
3	Console	Консольный порт RS-232 (RJ-45) для локального управления устройством.
4	microSD	Разъем для установки microSD-карт памяти.
5	USB 1	Порт USB 3.0 для подключения USB-устройств.
6	USB 2	Порт USB 3.0 для подключения USB-устройств.
7	1GE Combo [1 .. 8]	8 портов Gigabit Ethernet 10/100/1000BASE-X (SFP).
8	25 GE [1 .. 4]	4 порта 1000BASE-X/10GBASE-R/25GBASE-R.

Задняя панель устройства ESBC-3250

Внешний вид задней панели ESBC-3250 приведен на рисунке ниже.

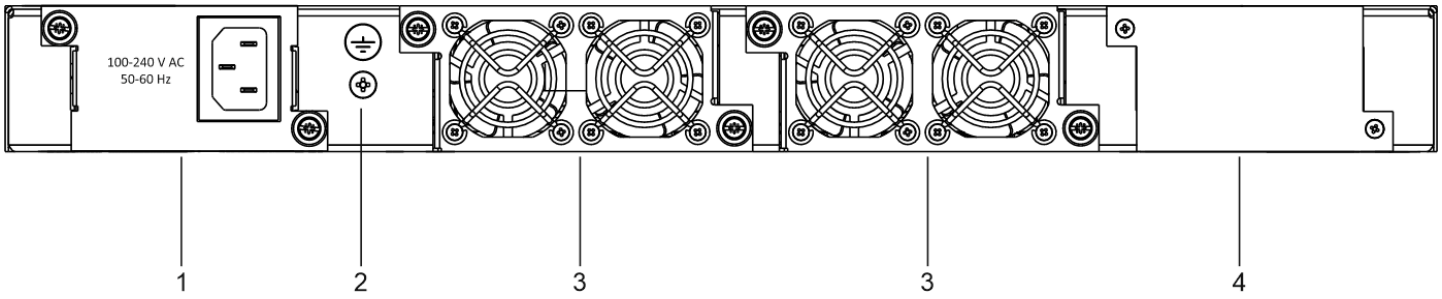


Рисунок 14 – Задняя панель ESBC-3250

В таблице 16 приведен перечень разъемов, расположенных на задней панели ESBC-3250.

Таблица 16 – Описание разъемов задней панели ESBC-3250

№	Описание
1	Основной источник питания.
2	Клемма для заземления устройства.
3	Съемные вентиляционные модули с возможностью горячей замены.
4	Место для установки резервного источника питания.

Боковые панели устройства ESBC-3250

Внешний вид боковых панелей ESBC-3250 приведен на рисунках ниже.

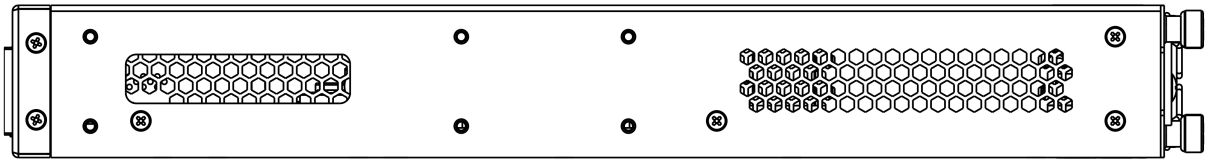


Рисунок 15 – Правая боковая панель ESBC-3250

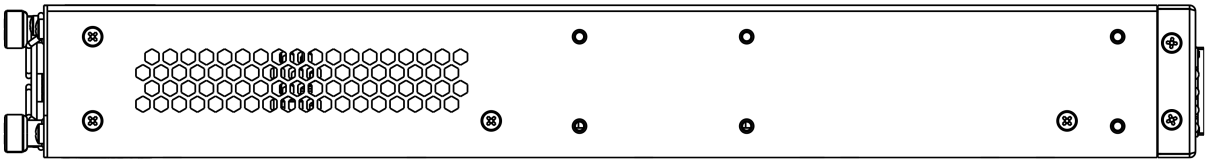


Рисунок 16 – Левая боковая панель ESBC-3250

На боковых панелях устройства расположены вентиляционные решетки, которые служат для отвода тепла. Не закрывайте вентиляционные отверстия посторонними предметами. Это может привести к перегреву компонентов устройства и вызвать нарушения в его работе. Рекомендации по установке устройства расположены в разделе Установка и подключение.

2.4.5 Конструктивное исполнение ESBC-3350

Устройство выполнено в металлическом корпусе с возможностью установки в 19” конструктив, высота корпуса 1U.

Передняя панель устройства ESBC-3350

Внешний вид передней панели ESBC-3350 показан на рисунке 17.

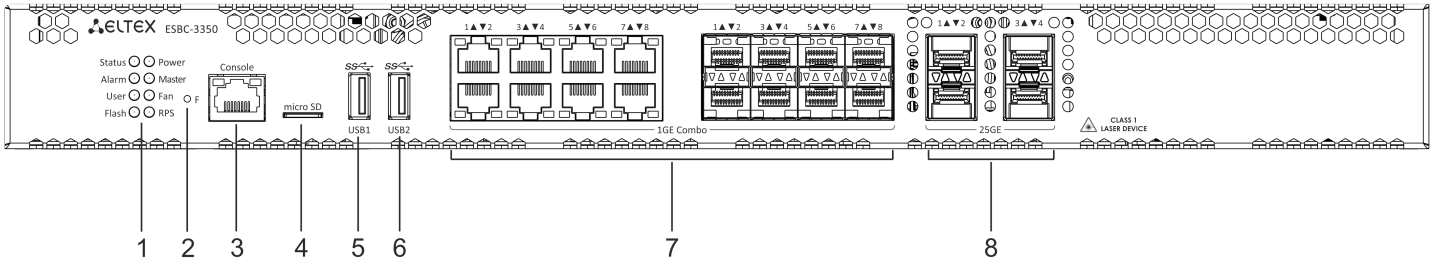


Рисунок 17 – Передняя панель ESBC-3350

В таблице 17 приведен перечень разъемов, светодиодных индикаторов и органов управления, расположенных на передней панели ESBC-3350.

Таблица 17 – Описание разъемов, индикаторов и органов управления передней панели ESBC-3350

№	Элемент передней панели	Описание
1	Status	Индикатор текущего состояния устройства.
	Alarm	Индикатор наличия и уровня аварии устройства.
	User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.
	Flash	Индикатор активности обмена с накопителем данных – microSD-картой или USB Flash.
	Power	Индикатор питания устройства.
	Master	Индикатор работы устройства в failover-режимах (не используется в текущей версии).
	Fan	Индикатор аварии вентиляторов.
	RPS	Индикатор резервного источника электропитания.
2	F	Функциональная кнопка для перезагрузки устройства и сброса к заводским настройкам: • при удержании кнопки менее 10 секунд происходит перезагрузка устройства; • при удержании кнопки более 10 секунд происходит перезагрузка устройства и сброс к заводским настройкам.

№	Элемент передней панели	Описание
3	Console	Консольный порт RS-232 (RJ-45) для локального управления устройством.
4	microSD	Разъем для установки microSD-карт памяти.
5	USB 1	Порт USB 3.0 для подключения USB-устройств.
6	USB 2	Порт USB 3.0 для подключения USB-устройств.
7	1GE Combo [1 .. 8]	8 портов Gigabit Ethernet 10/100/1000BASE-X (SFP).
8	25GE [1 .. 4]	4 порта 1000BASE-X/10GBASE-R/25GBASE-R.

Задняя панель устройства ESBC-3350

Внешний вид задней панели ESBC-3350 приведен на рисунке ниже.

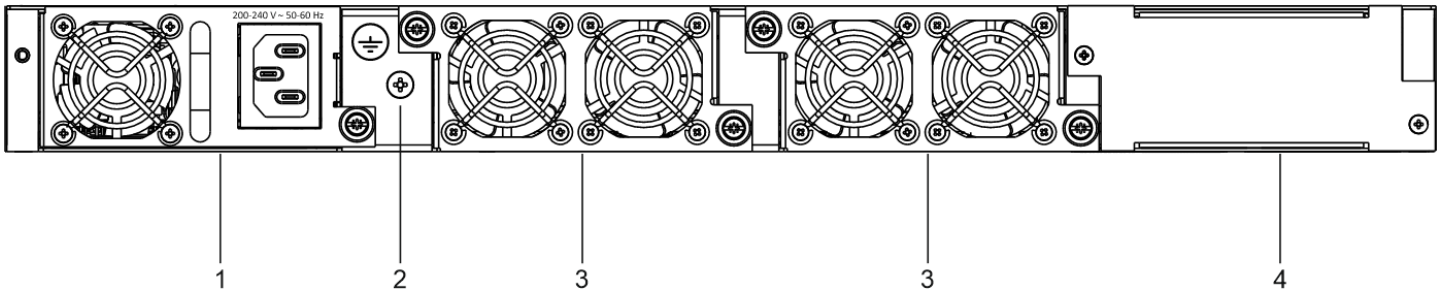


Рисунок 18 – Задняя панель ESBC-3350

В таблице 18 приведен перечень разъемов, расположенных на задней панели ESBC-3350.

Таблица 18 – Описание разъемов задней панели ESBC-3350

№	Описание
1	Основной источник питания.
2	Клемма для заземления устройства.
3	Съемные вентиляционные модули с возможностью горячей замены.
4	Место для установки резервного источника питания.

Боковые панели устройства ESBC-3350

Внешний вид боковых панелей ESBC-3350 приведен на рисунках ниже.

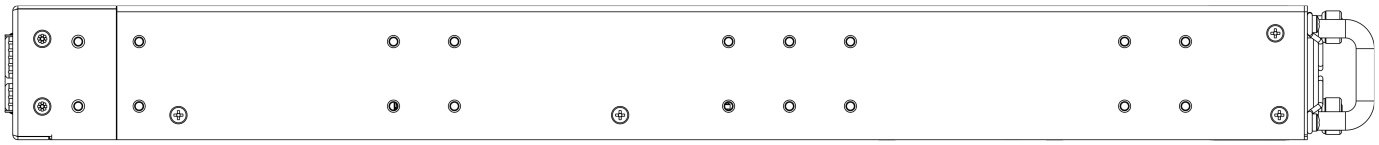


Рисунок 19 – Правая боковая панель ESBC-3350

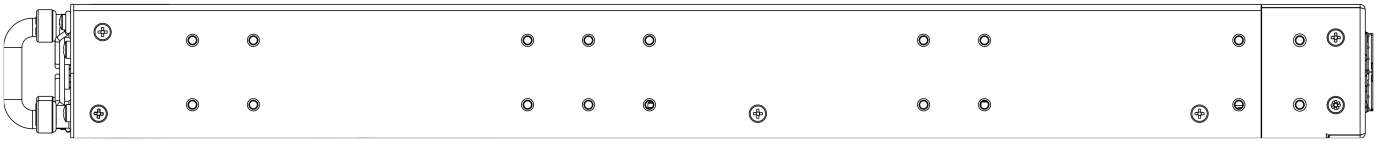


Рисунок 20 – Левая боковая панель ESBC-3350

На боковых панелях устройства расположены вентиляционные решетки, которые служат для отвода тепла. Не закрывайте вентиляционные отверстия посторонними предметами. Это может привести к перегреву компонентов устройства и вызвать нарушения в его работе. Рекомендации по установке устройства расположены в разделе Установка и подключение.

2.4.6 Световая индикация

Световая индикация ESBC-15R

Состояние интерфейсов Ethernet отображается двумя светодиодными индикаторами – *LINK/ACT* зеленого цвета и *SPEED* янтарного цвета.

Таблица 19 – Световая индикация состояния интерфейсов Ethernet

Свечение индикатора SPEED	Свечение индикатора LINK/ACT	Состояние интерфейса Ethernet
Выключен	Выключен	Порт выключен или соединение не установлено.
Выключен	Горит постоянно	Установлено соединение на скорости 10 или 100 Мбит/с.
Горит постоянно	Горит постоянно	Установлено соединение на скорости 1000 Мбит/с.
X	Мигает	Идет передача данных.

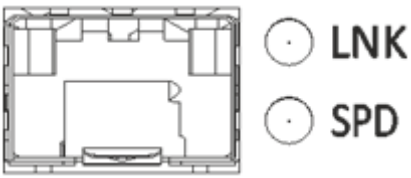


Рисунок 21 – Расположение индикаторов разъема SFP

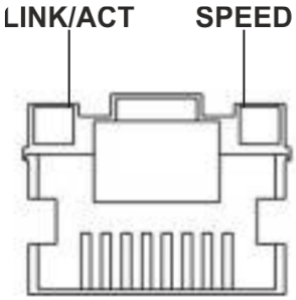


Рисунок 22 − Расположение индикаторов разъема RJ-45

В следующей таблице приведено описание состояний системных индикаторов устройства и их значений.

Таблица 20 − Состояния системных индикаторов

Название индикатора	Функция индикатора	Состояние индикатора	Состояние устройства
Power	Индикатор питания устройства.	Зеленый	Питание устройства в норме. Источник питания работает нормально, основное ПО прогружено.
		Красный	Не прогружено основное ПО.
		Выключен	Отказ внутренних источников питания устройства.
Status	Индикатор текущего состояния устройства.	Зеленый	Устройство работает нормально.
		Красный	Устройство находится в состоянии загрузки ПО.
Alarm	Индикатор наличия и уровня аварии устройства.	-	-
User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.	-	-

Световая индикация ESBC-31

Состояние интерфейсов Ethernet отображается двумя светодиодными индикаторами – *LINK/ACT* зеленого цвета и *SPEED* янтарного цвета.

Таблица 21 – Световая индикация состояния интерфейсов Ethernet

Свечение индикатора SPEED	Свечение индикатора LINK/ACT	Состояние интерфейса Ethernet
Выключен	Выключен	Порт выключен или соединение не установлено.
Выключен	Горит постоянно зелёным	Установлено соединение на скорости 10 или 100 Мбит/с.
Горит постоянно янтарным	Горит постоянно зелёным	Установлено соединение на скорости 1000 Мбит/с.
X	Мигает	Идет передача данных.

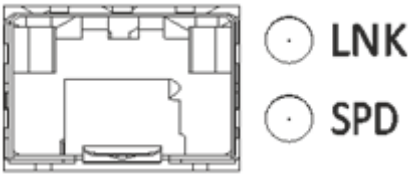


Рисунок 23 – Расположение индикаторов состояния SFP/SFP+-интерфейсов

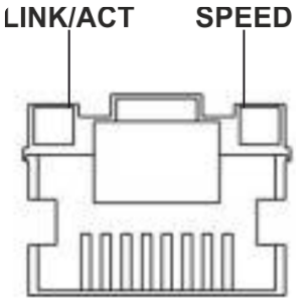


Рисунок 24 – Расположение индикаторов разъема RJ-45

В следующей таблице приведено описание состояний системных индикаторов устройства и их значений.

Таблица 22 – Состояния системных индикаторов

Название индикатора	Функция индикатора	Состояние индикатора	Состояние устройства
Power	Индикатор питания устройства.	Зеленый	Питание устройства в норме. Источник питания работает нормально, основное ПО прогружено.
		Красный	Не прогружено основное ПО.

Название индикатора	Функция индикатора	Состояние индикатора	Состояние устройства
		Выключен	Отказ внутренних источников питания устройства.
Status	Индикатор текущего состояния устройства.	Зеленый	Устройство работает нормально.
		Мигает зеленым	Устройство находится в состоянии загрузки ПО.
Alarm	Индикатор наличия и уровня аварии устройства.	-	-
User	Индикатор, использующийся для пользовательских сценариев, которые можно назначить в режиме конфигурирования устройства.	-	-
Flash	Индикатор активности обмена с накопителем данных: SD-картой или USB Flash.	Зеленый	Выполнение операций чтения/записи по команде «copy».
Master	Индикатор работы устройства в failover-режимах.	-	-
Fan	Состояние вентилятора охлаждения).	Зеленый	Все вентиляторы исправны.
		Красный	Отказ одного или более вентиляторов. Причиной возникновения аварии может быть неработоспособность хотя бы одного из вентиляторов − остановка или пониженная частота оборотов.
RPS	Режим работы резервного источника питания.	Зеленый	Резервный источник установлен и исправен.
		Выключен	Резервный источник не установлен.
		Красный	Отсутствие первичного питания резервного источника или его неисправность.

Световая индикация ESBC-3350, ESBC-3250, ESBC-3200

Состояние интерфейсов Ethernet отображается двумя светодиодными индикаторами – *LINK/ACT* зеленого цвета и *SPEED* янтарного цвета. Расположение светодиодных индикаторов интерфейсов показано ниже на рисунках 25 и 26. Значения световой индикации описаны в таблицах 23 и 24 соответственно.

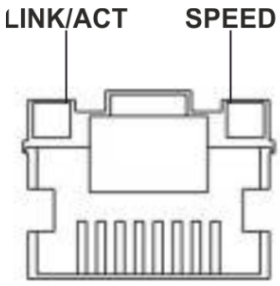


Рисунок 25 – Расположение индикаторов разъема RJ-45

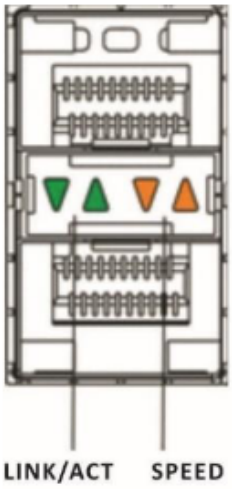


Рисунок 26 – Расположение индикаторов состояния SFP/SFP+/SFP28-интерфейсов

Таблица 23 – Световая индикация состояния RJ-45 интерфейсов

Свечение индикатора SPEED	Свечение индикатора LINK/ACT	Состояние интерфейса Ethernet
Выключен	Выключен	Порт выключен или соединение не установлено.
Выключен	Горит постоянно зелёным	Установлено соединение на скорости 10 или 100 Мбит/с.
Горит постоянно янтарным	Горит постоянно зелёным	Установлено соединение на скорости 1000 Мбит/с.
X	Мигает	Идет передача данных.

Таблица 24 – Световая индикация состояния состояния SFP/SFP+/SFP28-интерфейсов

Свечение индикатора SPEED	Свечение индикатора LINK/ACT	Состояние интерфейса Ethernet
Выключен	Выключен	Порт выключен или соединение не установлено.
Выключен	Горит постоянно зелёным	Установлено соединение на скорости 1 Гбит/с.
Горит постоянно янтарным	Горит постоянно зелёным	Установлено соединение на скорости 10 Гбит/с.
X	Мигает	Идет передача данных.

В следующей таблице приведено описание состояний системных индикаторов устройства и их значений.

Таблица 25 – Состояния системных индикаторов

Название индикатора	Функция индикатора	Состояние индикатора	Состояние устройства
Status	Индикатор текущего состояния устройства.	Зеленый	Устройство работает нормально.
		Мигает зеленым	Устройство находится в состоянии загрузки ПО.
Alarm	Индикатор наличия и уровня аварии устройства.	-	-
User	Индикатор пользовательских сценариев.	-	-
Flash	Индикатор активности обмена с накопителем данных: SD-картой или USB Flash.	Зеленый	Выполнение операций чтения/записи по команде «copy».
Power	Индикатор питания устройства.	Зеленый	Питание устройства в норме. Основной источник питания, если он установлен, работает нормально.
		Красный	Неработоспособность основного источника питания, авария или отсутствие первичной сети.
		Выключен	Отказ внутренних источников питания устройства.
Master	Индикатор работы устройства в failover-режимах.	-	-
Fan	Состояние вентилятора охлаждения.	Выключен	Все вентиляторы исправны.

Название индикатора	Функция индикатора	Состояние индикатора	Состояние устройства
		Красный	Отказ одного или более вентиляторов. Причиной возникновения аварии может быть неработоспособность хотя бы одного из вентиляторов – остановка или пониженная частота оборотов.
RPS	Режим работы резервного источника питания.	Зеленый	Резервный источник установлен и исправен.
		Выключен	Резервный источник не установлен.
		Красный	Отсутствие первичного питания резервного источника или его неисправность.

2.5 Комплект поставки

В базовый комплект поставки ESBC-15R входят:

- пограничный контроллер сессий ESBC-15R;
- кабель питания;
- консольный кабель;
- комплект для крепления устройства в стойку 19”;
- сертификат соответствия;
- паспорт;
- документация (опционально);
- памятка о документации.

В базовый комплект поставки ESBC-31 входят:

- пограничный контроллер сессий ESBC-31;
- консольный кабель;
- комплект для крепления устройства в стойку 19”;
- сертификат соответствия;
- паспорт;
- документация (опционально);
- памятка о документации.

В базовый комплект поставки ESBC-3200 входят:

- пограничный контроллер сессий ESBC-3200;
- консольный кабель;
- комплект для крепления устройства в стойку 19”;
- сертификат соответствия;
- паспорт;
- документация (опционально);
- памятка о документации.

В базовый комплект поставки ESBC-3250 входят:

- пограничный контроллер сессий ESBC-3250;
- консольный кабель;
- комплект для крепления устройства в стойку 19”;
- сертификат соответствия;
- паспорт;
- документация (опционально);
- памятка о документации.

В базовый комплект поставки ESBC-3350 входят:

- пограничный контроллер сессий ESBC-3350;
- консольный кабель;
- комплект для крепления устройства в стойку 19”;
- сертификат соответствия;
- паспорт;
- документация (опционально);
- памятка о документации.

По заказу покупателя в комплект поставки могут быть включены SFP/SFP+-трансиверы.

2.6 Пусковые токи ESBC-15R, ESBC-31, ESBC-3200

Устройство	Наименование БП	Характеристика пускового тока в рамках лаб. тестов	
		Пиковое значение, А	Длительность, мс
ESBC-15R	SPM35-220/12	34	6
	SPM35-220/12 rev.B	16.5	8
ESBC-31	PM160-220/12	40	4
	PM160-220/12 rev.B	50	8
	PM160-220/12 rev.C	50	8
	PM160-220/12 rev.D (опытная)	27	5
	PM100-48/12	200 при Vin = 76 В, 140 при Vin = 48 В	0.325 при Vin = 76 В, 0.125 при Vin = 48 В
	PM100-48/12 rev.B	160 при Vin = 72 В, 100 при Vin = 48 В	0.150 при Vin = 72 В, 0.150 при Vin = 48 В
ESBC-3200	PM160-220/12	40	4
	PM160-220/12 rev.B	50	8
	PM160-220/12 rev.C	50	8
	PM160-220/12 rev.D (опытная)	27	5
	PM160-48/12 DC	250 при Vin = 72 В, 180 при Vin = 48 В	0.2 при Vin = 72 В, 0.22 при Vin = 48 В

3 Установка и подключение

- Установка устройства в стойку (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250)
- Установка ESBC-3350 в стойку
- Установка модулей питания (ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350)
- Подключение питающей сети
- Установка и удаление SFP-трансиверов
  - Установка трансивера
  - Удаление трансивера
- Подключение к vESBC

В данном разделе описаны процедуры установки устройства в стойку и подключения к питающей сети.

3.1 Установка устройства в стойку (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250)

Для установки устройства в стойку:

1. Выберите необходимое положение кронштейна (рисунок 27). Совместите четыре отверстия кронштейна с четырьмя отверстиями на боковой панели устройства. С помощью отвертки прикрепите кронштейн винтами к корпусу.
2. Повторите шаг 1 для другой боковой панели устройства.
3. Совместите отверстия кронштейнов с отверстиями на передних вертикальных направляющих стойки. Используйте отверстия в направляющих на одном уровне с обеих сторон стойки, для того чтобы устройство располагалось горизонтально.
4. С помощью отвертки прикрепите устройство к стойке винтами.

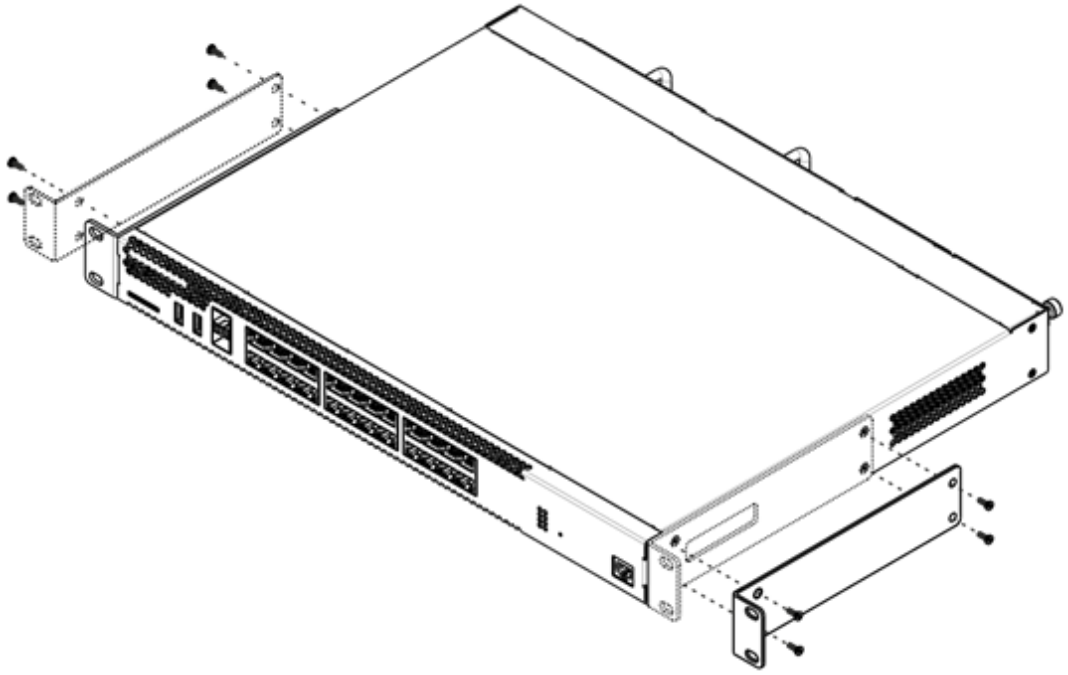


Рисунок 27 – Крепление кронштейнов к устройству

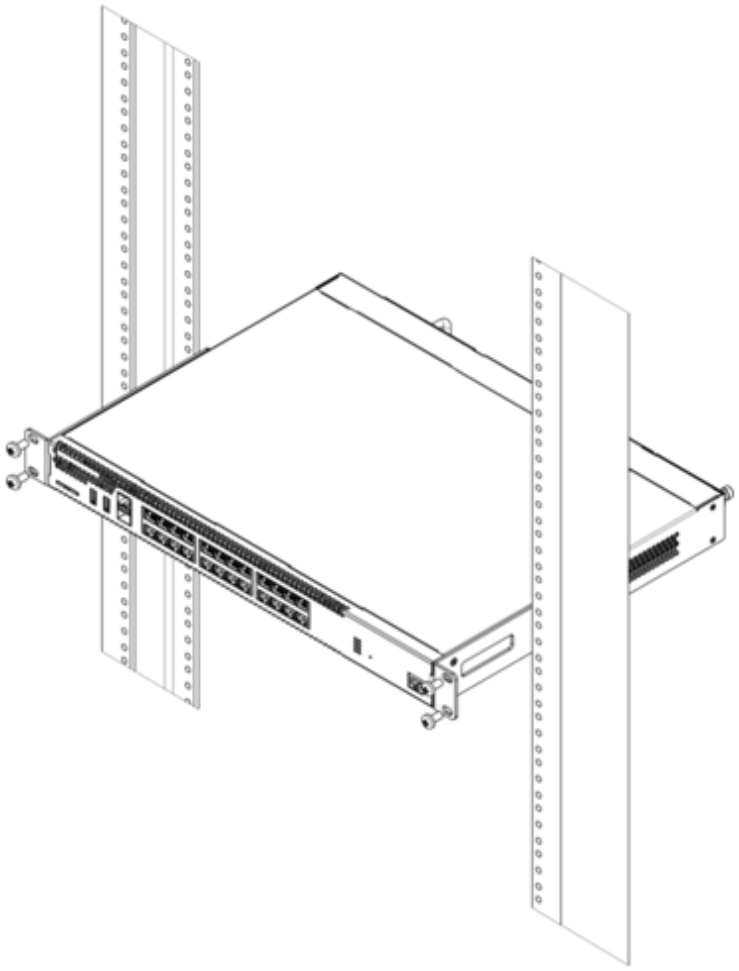


Рисунок 28 – Установка устройства в стойку

Вентиляция устройства организована по схеме фронт-тыл. На передней и боковых панелях устройства расположены вентиляционные отверстия, с задней стороны устройства расположены вентиляционные модули. Не закрывайте входные и выходные вентиляционные отверстия посторонними предметами во избежание перегрева компонентов устройства и нарушения его работы.

3.2 Установка ESBC-3350 в стойку

Для деталей длинного кронштейна предусмотрено несколько положений: в каждом случае необходимое положение определяется глубиной используемой стойки. Минимальная глубина, на которую рассчитан кронштейн — 537.5 мм, максимальная — 787.5 мм.

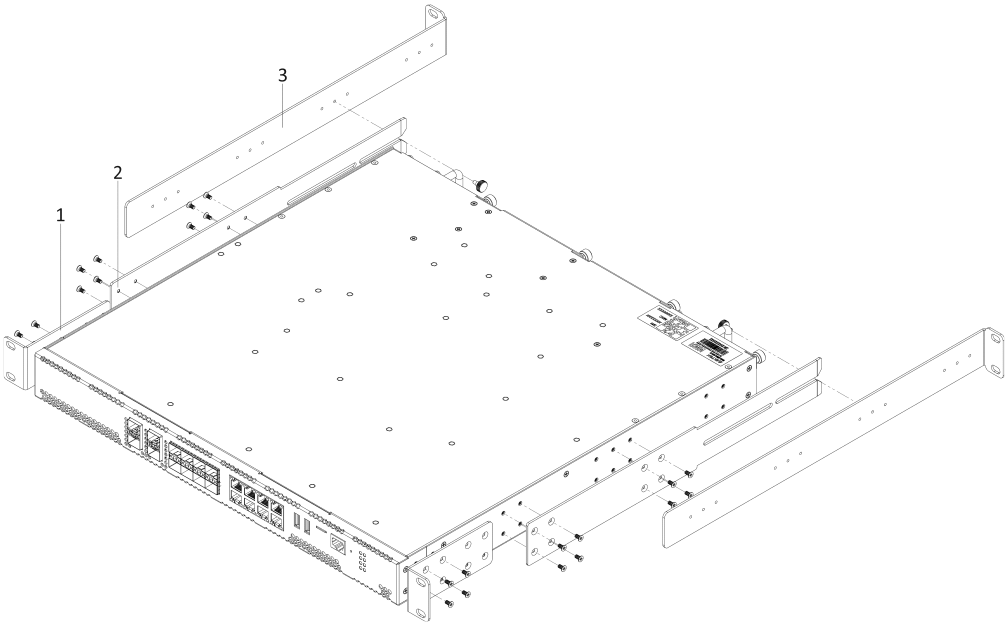


Рисунок 29 − Крепление кронштейнов к ESBC-3350

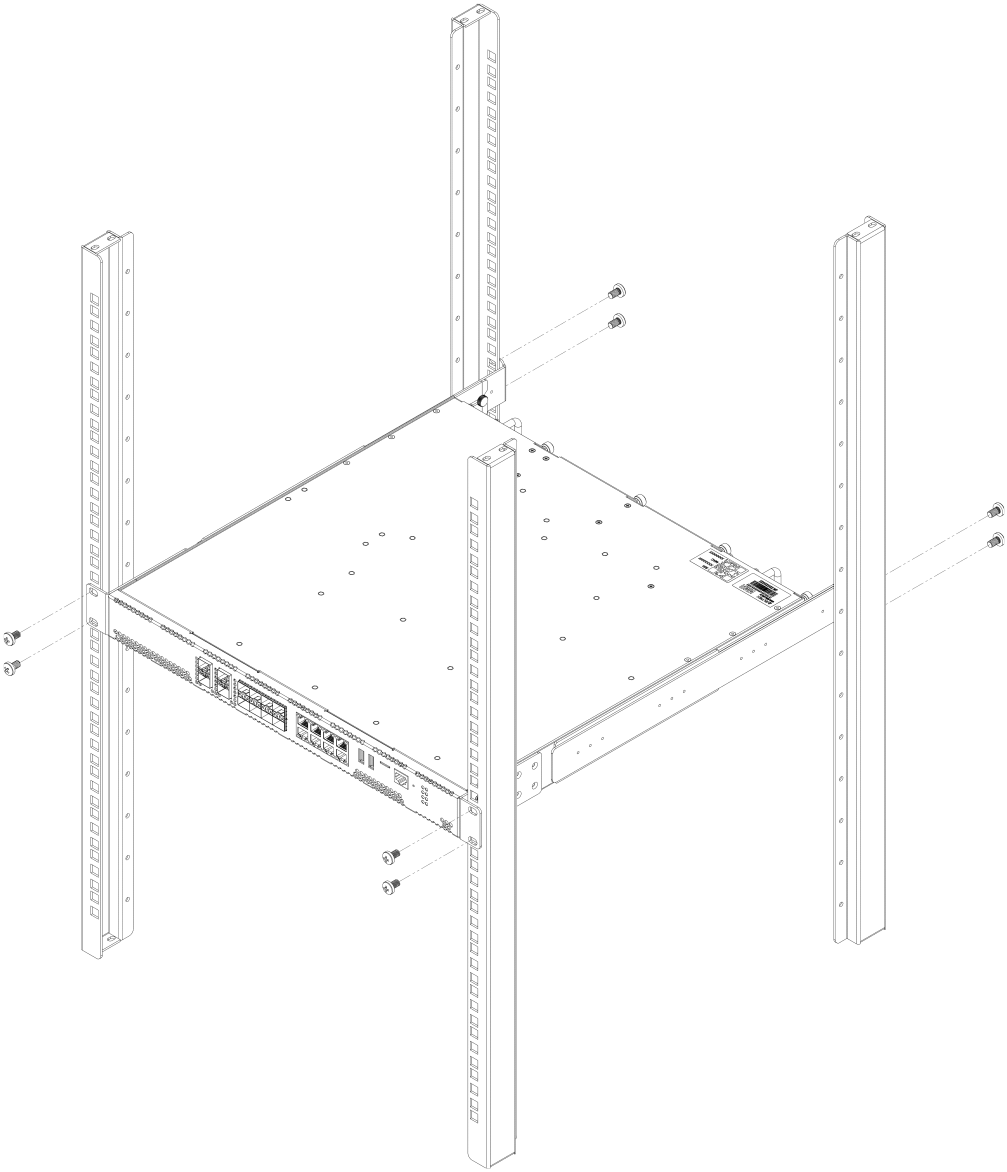


Рисунок 30 − Установка ESBC-3350 в стойку

Для установки устройства в стойку:

1. Выберите необходимое положение детали 1. Совместите четыре отверстия на детали 1 с четырьмя отверстиями на боковой панели устройства. С помощью отвертки прикрепите деталь 1 винтами к корпусу.
2. (Если необходимо) С помощью отвертки извлеките направляющий винт, соединяющий детали 2 и 3, и разъедините их.
3. Выберите необходимое положение детали 2. Совместите восемь отверстий на детали 2 с восемью отверстиями на боковой панели устройства. С помощью отвертки прикрепите деталь 2 винтами к корпусу.
4. Повторите шаги 1–3 для другой боковой панели устройства.
5. Совместите отверстия кронштейна на деталях 3 с отверстиями на задних вертикальных направляющих стойки и прикрепите детали к стойке винтами (рисунок 30). В зависимости от глубины стойки вкрутите направляющие винты в детали 3.
6. Совместите отверстия на деталях 1 с отверстиями на передних вертикальных направляющих стойки (рисунок 30). Используйте отверстия в направляющих на одном уровне с обеих сторон стойки, для того чтобы устройство располагалось горизонтально.
7. Установите устройство в стойку, совместив прорезь детали 2 с направляющим винтом в детали 3.
8. С помощью отвертки прикрепите устройство к стойке винтами.

3.3 Установка модулей питания (ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350)

ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 могут работать с одним или двумя модулями питания. Установка второго модуля питания необходима в случае использования устройства в условиях, требующих повышенной надежности.

Места для установки модулей питания с электрической точки зрения равноценны. С точки зрения использования устройства, модуль питания может находиться в основном и резервном разъеме, информация о приоритетности находится в таблице "Описание разъемов задней панели пограничного контроллера сессий". Модули питания могут устанавливаться и извлекаться без выключения устройства. При установке или извлечении дополнительного модуля питания пограничный контроллер сессий продолжает работу без перезапуска.

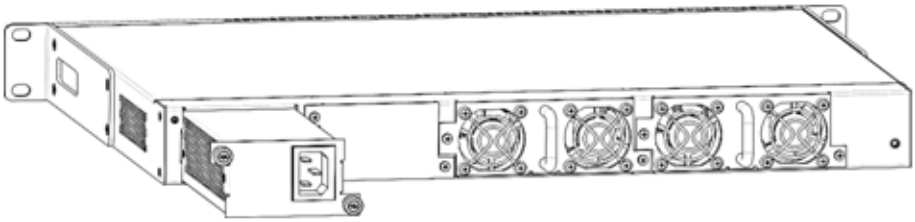


Рисунок 31– Установка модулей питания

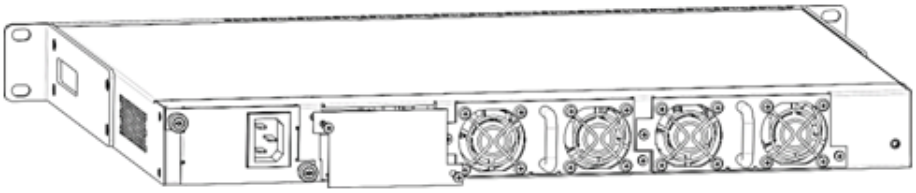


Рисунок 32 – Установка заглушки

Индикация аварии модуля питания может быть вызвана не только отказом модуля, но и отсутствием первичного питания.

Состояние модулей питания может быть проверено по индикации на передней панели пограничного контроллера сессий (см. раздел Световая индикация) или по диагностике, доступной через интерфейсы управления пограничным контроллером сессий.

3.4 Подключение питающей сети

1. Прежде чем к устройству будет подключена питающая сеть, необходимо заземлить корпус устройства через заземляющий винт М4. Заземление необходимо выполнять изолированным многожильным проводом. Устройство заземления и сечение заземляющего провода должны соответствовать требованиями Правил устройства электроустановок (ПУЭ).
2. Если предполагается подключение компьютера или иного оборудования к консольному порту пограничного контроллера сессий, это оборудование также должно быть надежно заземлено.
3. Подключите к устройству кабель питания. В зависимости от комплектации устройства, питание может осуществляться от сети переменного тока либо от сети постоянного тока. При подключении сети переменного тока следует использовать кабель, входящий в комплект устройства. Для подключения к сети постоянного тока используйте провод сечением не менее 1 мм^2.
4. Включите питание устройства и убедитесь в отсутствии аварий по состоянию индикаторов на передней панели.

3.5 Установка и удаление SFP-трансиверов

⚠ Установка оптических модулей может производиться как при выключенном, так и при включенном устройстве.

3.5.1 Установка трансивера

1. Вставьте верхний SFP-модуль в слот открытой частью разъема вниз, а нижний SFP-модуль – открытой частью разъема вверх.

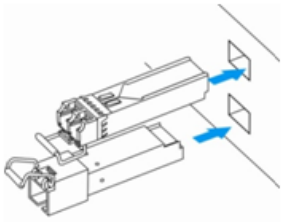


Рисунок 33 – Установка SFP-трансиверов

2. Надавите на модуль по направлению внутрь корпуса устройства до появления характерного щелчка фиксации модуля.

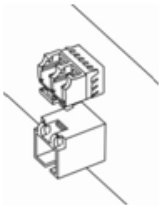


Рисунок 34 – Установленные SFP-трансиверы

3.5.2 Удаление трансивера

1. Откиньте рукоятку модуля, это приведет к разблокированию удерживающей защелки.

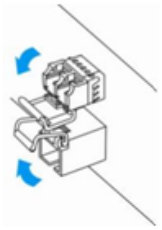


Рисунок 35 – Открытие защелки SFP-трансиверов

2. Извлеките модуль из слота.

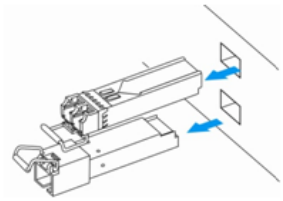


Рисунок 36 – Извлечение SFP-трансиверов

3.6 Подключение к vESBC

Для получения информации об установке и подключении к vESBC перейдите в раздел документации vESBC. Руководство по установке и настройке. Версия 1.9.0.

4 Интерфейсы управления

- Интерфейс командной строки (CLI)
- Web-интерфейс
- Типы и порядок именования сетевых интерфейсов пограничного контроллера сессий
- Типы и порядок именования туннелей пограничного контроллера сессий

Настройка и мониторинг устройства могут осуществляться через различные интерфейсы управления.

Для доступа к устройству может использоваться сетевое подключение по протоколам Telnet и SSH или прямое подключение через консольный порт, соответствующий спецификации RS-232. При доступе по протоколам Telnet, SSH и при подключении через консольный порт для управления устройством используется интерфейс командной строки.

Только для ESBC-15R/31/3200/3250/3350
Заводская конфигурация содержит описание доверенной зоны trusted и IP-адрес для доступа к управлению устройством – 192.168.1.1/24.
В доверенную зону входят интерфейсы:
- для ESBC-15R: GigabitEthernet 1/0/2-5;
- для ESBC-31: GigabitEthernet 1/0/2-14;
- для ESBC-3200: Twentyfivegigabitethernet 1/0/3-12;
- для ESBC-3250, ESBC-3350: GigabitEthernet 1/0/2-8, Twentyfivegigabitethernet 1/0/3-4.

Только для vESBC:
В заводской конфигурации отсутствует конфигурация сетевых интерфейсов и зон безопасности. После выполнения инсталляции vESBC требуется выполнить эти настройки через интерфейс гипервизора или через последовательный порт.
Для ESBC-15R/31/3200/3250/3350 и vESBC:
В заводской конфигурации по умолчанию создан пользователь «admin» с паролем «password».
Протоколы семейства STP (STP, RSTP, VSTP) отключены.
Заводскую конфигурацию можно сбросить командой *copy system:default-config system:candidate-config*. После сброса необходимо настроить ESBC с помощью консольного порта.

При использовании любого из перечисленных интерфейсов управления действуют единые принципы работы с конфигурацией. Должна соблюдаться определенная, описанная здесь, последовательность изменения и применения конфигурации, позволяющая защитить устройство от некорректного конфигурирования.

4.1 Интерфейс командной строки (CLI)

Интерфейс командной строки (Command Line Interface, CLI) – интерфейс, предназначенный для управления, просмотра состояния и мониторинга устройства. Для работы потребуется любая установленная на ПК программа, поддерживающая работу по протоколам Telnet, SSH или прямое подключение через консольный порт (например, HyperTerminal).

Интерфейс командной строки обеспечивает авторизацию пользователей и ограничивает их доступ к командам на основании уровня доступа, заданного администратором.

В системе может быть создано необходимое количество пользователей, права доступа задаются индивидуально для каждого из них.

Для обеспечения безопасности командного интерфейса все команды разделены на две категории – привилегированные и непривилегированные. К привилегированным в основном относятся команды конфигурирования. К непривилегированным – команды мониторинга.

Система позволяет нескольким пользователям одновременно подключаться к устройству.

4.2 Web-интерфейс

Web-интерфейс имеет ограниченный набор параметров конфигурирования устройства, а также содержит инструменты мониторинга.

Описание web-интерфейса приведено в разделе Управление через web-интерфейс.

4.3 Типы и порядок именования сетевых интерфейсов пограничного контроллера сессий

При работе пограничного контроллера сессий используются сетевые интерфейсы различного типа и назначения. Система именования позволяет однозначно адресовать интерфейсы по их функциональному назначению и местоположению в системе. Далее в таблице приведен перечень типов интерфейсов.

Таблица 26 – Типы и порядок именования интерфейсов пограничного контроллера сессий

Физические интерфейсы	Обозначение физического интерфейса включает в себя его тип и идентификатор. Идентификатор физических интерфейсов имеет вид **<UNIT>/<SLOT>/<PORT>**, где: • **<UNIT>** – номер устройства в группе устройств [1..4], • **<SLOT>** – номер модуля в составе устройства или 0 при отсутствии деления устройства на модули, • **<PORT>** – порядковый номер порта.
Порты 1 Гбит/с	**gigabitethernet <UNIT>/<SLOT>/<PORT>** Пример обозначения: **gigabitethernet 1/0/12** ⚠ Допускается использовать сокращенное наименование, например gi1/0/12.
Порты 10 Гбит/с	**tengigabitethernet <UNIT>/<SLOT>/<PORT>** Пример обозначения: **tengigabitethernet 1/0/2** ⚠ Допускается использовать сокращенное наименование, например te1/0/2.
Порты 25 Гбит/с	**twentyfivegigabitethernet <UNIT>/<SLOT>/<PORT>** Пример обозначения: **twentyfivegigabitethernet 1/0/2** ⚠ Допускается использовать сокращенное наименование, например twe1/0/2.
Группы агрегации каналов	Обозначение группы агрегации каналов включает в себя его тип и идентификатор. Идентификатор port-channel-интерфейсов может иметь вид **{ <CHANNEL_ID> | <UNIT>/<CHANNEL_ID> }**, где • **<UNIT>** – номер устройства в группе устройств [1..4], • **<CHANNEL_ID>** – порядковый номер группы агрегации каналов [1..12] Примеры обозначений: **port-channel 6** **port-channel 1/6** ⚠ Допускается использовать сокращенное наименование, например, po1.

Саб-интерфейсы	Обозначение саб-интерфейса образуется из обозначения базового интерфейса и идентификатора (VLAN) саб-интерфейса, разделенных точкой. Примеры обозначений: • **gigabitethernet 1/0/12.100** • **tengigabitethernet 1/0/2.123** • **twentyfivegigabitethernet 1/0/2.200** • **fortygigabitethernet 1/0/2.1024** • **port-channel 1.6** • **port-channel 1/6.6** Идентификатор саб-интерфейса может принимать значения [2..4094].
Q-in-Q интерфейсы	Обозначение Q-in-Q интерфейса образуется из обозначения базового интерфейса, идентификатора сервисного VLAN и идентификатора пользовательского VLAN, разделенных точкой. Примеры обозначений: • **gigabitethernet 1/0/12.100.10** • **tengigabitethernet 1/0/2.45.12** • **twentyfivegigabitethernet 1/0/2.100.200** • **fortygigabitethernet 1/0/2.408.507** • **port-channel 1.6.34** • **port-channel 1/6.6.34** Идентификатор сервисного и пользовательского VLAN может принимать значения [1..4094].
Логические интерфейсы	Обозначение логического интерфейса является порядковым номером интерфейса: Примеры обозначений: • **loopback 4** • **bridge 60** • **service-port 1**

1. Количество интерфейсов каждого типа зависит от модели пограничного контроллера.
2. Текущая версия ПО поддерживает кластеризацию устройств единой модели. Номер unit в группе устройств может принимать значение в диапазоне от 1 до 4. В текущей версии ПО поддерживается кластеризация двух устройств, поэтому следует использовать номера unit 1 и 2.
3. Некоторые команды поддерживают одновременную работу с группой интерфейсов. Для указания группы интерфейсов может быть использовано перечисление через запятую или указание диапазона идентификаторов через дефис «-».
Примеры указания групп интерфейсов:

```
interface gigabitethernet 1/0/1, gigabitethernet 1/0/5
interface tengigabitethernet 1/0/1-2
interface twentyfivegigabitethernet 1/0/3-4
interface fortygigabitethernet 1/0/1-2
interface gi1/0/1-3,gi1/0/7,te1/0/1,fo1/0/1
```

4.4 Типы и порядок именования туннелей пограничного контроллера сессий

При работе пограничного контроллера сессий используются сетевые туннели различного типа и назначения. Система именования позволяет однозначно адресовать туннели по их функциональному назначению. Далее в таблице приведен перечень типов туннелей.

Таблица 27 – Типы и порядок именования туннелей пограничного контроллера сессий

Тип туннеля	Обозначение
L2TP-туннель	Обозначение L2TP-туннеля состоит из обозначения типа и порядкового номера туннеля: **l2tp <L2TP_ID>** Пример обозначения: **l2tp 1**
L2TPv3-туннель	Обозначение L2TPv3-туннеля состоит из обозначения типа и порядкового номера туннеля: **l2tpv3 <L2TPV3_ID>** Пример обозначения: **l2tpv3 1**
GRE-туннель	Обозначение GRE-туннеля состоит из обозначения типа и порядкового номера туннеля: **gre <GRE_ID>** Пример обозначения: **gre 1**
IPv4-over-IPv4-туннель	Обозначение IPv4-over-IPv4-туннеля состоит из обозначения типа и порядкового номера туннеля: **ip4ip4 <IPIP_ID>** Пример обозначения: **ip4ip4 1**
IPsec-туннель	Обозначение виртуального IPsec-туннеля состоит из обозначения типа и порядкового номера туннеля: **vti <VTI_ID>** Пример обозначения: **vti 1**
Логический туннель (туннель между VRF)	Обозначение логического туннеля состоит из обозначения типа и порядкового номера туннеля: **lt <LT_ID>** Пример обозначения: **lt 1**

Тип туннеля	Обозначение
PPPoE-туннель	Обозначение PPPoE-туннеля состоит из обозначения типа и порядкового номера туннеля: **pppoe <PPPOE_ID>** Пример обозначения: **pppoe 1**
OpenVPN-туннель	Обозначение OpenVPN-туннеля состоит из обозначения типа и порядкового номера туннеля: **openvpn <OPENVPN_ID>** Пример обозначения: **openvpn 1**
PPTP-туннель	Обозначение PPTP-туннеля состоит из обозначения типа и порядкового номера туннеля: **pptp <PPTP_ID>** Пример обозначения: **pptp 1**
Wireguard-туннель	Обозначение Wireguard-туннеля состоит из обозначения типа и порядкового номера туннеля: **wireguard <WG_ID>** Пример обозначения: **wireguard 1**

⚠ Количество туннелей каждого типа зависит от модели и ПО пограничного контроллера сессий.

5 Начальная настройка устройства

- Заводская конфигурация устройства
  - ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350
  - vESBC
  - Описание заводской конфигурации (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350)
- Подключение и конфигурирование устройства
  - Подключение к ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350
    - Подключение по локальной сети Ethernet
    - Подключение через консольный порт RS-232
  - Подключение к vESBC
  - Применение изменения конфигурации
  - Базовая настройка устройства
    - Изменение пароля пользователя «admin» при первой авторизации
    - Создание новых пользователей
    - Назначение имени устройства
    - Настройка параметров публичной сети
    - Настройка удаленного доступа к устройству

5.1 Заводская конфигурация устройства

5.1.1 ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350

При отгрузке ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 клиенту, на пограничном контроллере сессий будет загружена заводская конфигурация, которая включает минимально необходимые базовые настройки. Заводская конфигурация позволяет использовать пограничный контроллер сессий в качестве шлюза с функцией SNAT без необходимости применять дополнительные настройки. Кроме того, заводская конфигурация содержит настройки, позволяющие получить сетевой доступ к устройству для выполнения расширенного конфигурирования.

Заводскую конфигурацию можно сбросить командой *copy system:default-config system:candidate-config*. После сброса необходимо настроить аппаратный пограничный контроллер сессий с помощью консольного порта.

5.1.2 vESBC

В заводской конфигурации vESBC отсутствует конфигурация сетевых интерфейсов и зон безопасности. После выполнения инсталляции vESBC требуется выполнить эти настройки через графический интерфейс гипервизора или через последовательный порт.

> Процесс установки и первоначальной настройки vESBC описан в разделе vESBC. Руководство по установке и настройке.

Заводскую конфигурацию можно сбросить командой *copy system:default-config system:candidate-config*

> Для обеспечения возможности конфигурирования устройства при первом включении в конфигурации устройства создана учётная запись администратора "admin" с паролем "password".
> Пользователю будет предложено изменить пароль администратора при начальном конфигурировании устройства.

5.1.3 Описание заводской конфигурации (ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350)

Для подключения к сетям в конфигурации описаны 2 зоны безопасности с наименованиями «Trusted» для локальной сети и «Untrusted» для публичной сети. Все интерфейсы разделены между двух зон безопасности:

1. **Зона «Untrusted»** предназначена для подключения к публичной сети (WAN). В этой зоне открыты порты DHCP-протокола для получения динамического IP-адреса от провайдера. Все входящие соединения из данной зоны на пограничный контроллер сессий запрещены.
   В данную зону безопасности входят интерфейсы:
   - для ESBC-15R: GigabitEthernet1/0/1; GigabitEthernet1/0/6;
   - для ESBC-31: GigabitEthernet 1/0/1; TengigabitEthernet 1/0/1-2;
   - для ESBC-3200: Twentyfivegigabitethernet 1/0/1-2;
   - для ESBC-3250/3350: GigabitEthernet 1/0/1, TwentyfivegigabitEthernet 1/0/1-2.

Интерфейсы зоны объединены в один L2-сегмент через сетевой мост *Bridge 2*.

1. **Зона «Trusted»** предназначена для подключения к локальной сети (LAN). В этой зоне открыты порты протоколов Telnet и SSH для удаленного доступа, ICMP-протокола для проверки доступности пограничного контроллера сессий, DHCP-протокола для получения клиентами IP-адресов от устройства. Исходящие соединения из данной зоны в зону «Untrusted» разрешены.
   В данную зону безопасности входят интерфейсы:
   - для ESBC-15R: GigabitEthernet 1/0/2-5;
   - для ESBC-31: GigabitEthernet 1/0/2-14;
   - для ESBC-3200: Twentyfivegigabitethernet 1/0/3-12;
   - для ESBC-3250/3350: GigabitEthernet 1/0/2-8, Twentyfivegigabitethernet 1/0/3-4.

Интерфейсы зоны объединены в один L2-сегмент через сетевой мост *Bridge 1*.

На интерфейсе *Bridge 2* включен DHCP-клиент для получения динамического IP-адреса от провайдера. На интерфейсе *Bridge 1* сконфигурирован статический IP-адрес 192.168.1.1/24. Созданный IP-интерфейс выступает в качестве шлюза для клиентов локальной сети. Для клиентов локальной сети настроен DHCP-пул адресов 192.168.1.2-192.168.1.254 с маской 255.255.255.0. Для получения клиентами локальной сети доступа к Internet на пограничном контроллере сессий включен сервис Source NAT.

Политики зон безопасности настроены следующим образом (см. таблицу 28).

Таблица 28 – Описание политик зон безопасности

Зона, из которой идет трафик	Зона, в которую идет трафик	Тип трафика	Действие
Trusted	Untrusted	TCP, UDP, ICMP	разрешен
Trusted	Trusted	TCP, UDP, ICMP	разрешен
Trusted	self	TCP/22 (SSH), ICMP, UDP/67 (DHCP Server), UDP/123 (NTP)	разрешен
Untrusted	self	UDP/68 (DHCP Client)	разрешен

Для обеспечения возможности конфигурирования устройства при первом включении в конфигурации устройства создана учётная запись администратора "admin" с паролем "password".
Пользователю будет предложено изменить пароль администратора при начальном конфигурировании устройства.

Для сетевого доступа к управлению пограничным контроллером сессий при первом включении в конфигурации задан статический IP-адрес на интерфейсе Bridge 1 – 192.168.1.1/24.

5.2 Подключение и конфигурирование устройства

Пограничные контроллеры сессий ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350/vESBC предназначены для выполнения функций пограничного шлюза VoIP и обеспечения безопасности сети пользователя при подключении ее к публичным сетям передачи данных.

Базовая настройка устройства должна включать:

- назначение IP-адресов (статических или динамических) интерфейсам, участвующим в обработке сетевого и VoIP-трафика;
- создание зон безопасности и распределение интерфейсов по зонам;
- создание политик, регулирующих прохождение данных между зонами;
- настройка сервисов, сопутствующих обеспечению маршрутизации VoIP-трафика (DNS, Firewall и прочие);
- настройка функционала пограничного шлюза VoIP.

Расширенные настройки зависят от требований конкретной схемы применения устройства и легко могут быть добавлены или изменены с помощью имеющихся интерфейсов управления.

5.2.1 Подключение к ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350

Ниже описаны предусмотренные способы подключения к устройству ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350.

Для подключения к vESBC используется графический интерфейс гипервизора или последовательный порт.

Подключение по локальной сети Ethernet

При первоначальном старте устройство загружается с заводской конфигурацией. Описание заводской конфигурации приведено в разделе Заводская конфигурация устройства данного руководства.

Подключите сетевой кабель передачи данных (патч-корд) к любому порту, входящему в зону «***Trusted***», и к компьютеру, предназначенному для управления.

В заводской конфигурации пограничного контроллера сессий активирован DHCP-сервер с пулом IP-адресов в подсети **192.168.1.0/24**.

При подключении сетевого интерфейса управляющего компьютера он должен получить сетевой адрес от сервера.

Если IP-адрес не получен по какой-либо причине, то следует назначить адрес интерфейса вручную, используя любой адрес, кроме 192.168.1.1, в подсети 192.168.1.0/24.

Подключение через консольный порт RS-232

При помощи кабеля RJ-45/DBF9, который входит в комплект поставки устройства, соедините порт **«Console»** пограничного контроллера сессий с портом RS-232 компьютера.

Запустите терминальную программу (например, HyperTerminal или Minicom) и создайте новое подключение. Должен быть использован режим эмуляции терминала VT100.

Выполните следующие настройки интерфейса RS-232:

Скорость: 115200 бит/с
Биты данных: 8 бит
Четность: нет
Стоповые биты: 1
Управление потоком: нет

5.2.2 Подключение к vESBC

Подключение к vESBC после инсталляции ПО осуществляется через графический интерфейс гипервизора или через последовательный порт.

⚠ Процесс установки и первоначальной настройки vESBC описан в разделе vESBC. Руководство по установке и настройке.

5.2.3 Применение изменения конфигурации

Любые изменения, внесенные в конфигурацию, вступят в действие только после применения команды:

```
esbc# commit
Configuration has been successfully committed
```

После применения данной команды запускается таймер "отката" конфигурации. Для остановки таймера и механизма "отката" используется команда:

```
esbc# confirm
Configuration has been successfully confirmed
```

Значение таймера "отката" по умолчанию – 600 секунд. Для изменения данного таймера используется команда:

```
esbc(config)# system config-confirm timeout <TIME>
```

- <TIME> – интервал времени ожидания подтверждения конфигурации, принимает значение в секундах [120..86400].

5.2.4 Базовая настройка устройства

Процедура настройки устройств при первом включении состоит из следующих этапов:

- Изменение пароля пользователя «admin».
- Создание новых пользователей.
- Назначение имени устройства (Hostname).
- Установка параметров подключения к публичной сети в соответствии с требованиями провайдера.
- Настройка удаленного доступа к пограничному контроллеру сессий.
- Применение базовых настроек.

Изменение пароля пользователя «admin» при первой авторизации

При первом входе в систему необходимо сменить пароль по умолчанию привилегированного пользователя «admin». До смены пароля пользовательская настройка устройства недоступна.

После указания нового пароля необходимо применить изменения в конфигурации командой **commit** и подтвердить изменения командой **confirm**:

```
esbc(change-expired-password)# password <new password>
esbc(change-expired-password)# commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
esbc(change-expired-password)# confirm
Configuration has been confirmed. Commit timer canceled.
esbc#
```

Создание новых пользователей

Для управления устройством на ESBC существует возможность создавать пользовательские учетные записи, у которых администратор может индивидуально задать:

- пароль;
- уровень привилегий;
- режим работы учетной записи.

Для создания нового пользователя системы или настройки любого из параметров: имени пользователя, пароля, уровня привилегий и режима работы – используются команды:

```
esbc(config)# username <name>
esbc(config-user)# password <password>
esbc(config-user)# privilege <privilege>
esbc(config-user)# mode <mode>
esbc(config-user)# exit
```

Уровни привилегий 1–9 разрешают доступ к устройству и просмотр его оперативного состояния, но запрещают настройку. Уровни привилегий 10–14 разрешают как доступ, так и настройку большей части функций устройства. Уровень привилегий 15 разрешает как доступ, так и настройку всех функций устройства.

У учетных записей есть несколько режимов работы:

- cli – режим работы по умолчанию, пользователь получает доступ к интерфейсу командной строки, предназначенному для управления, просмотра состояния и мониторинга устройства;
- techsupport – пользователь получает доступ к командной оболочке, в которой выполняется процедура отладки устройства совместно с специалистами технической поддержки;
- sftp – пользователь используется для организации доступа к встроенному SFTP-серверу, возможность работы в какой-либо командой оболочке при этом у пользователя отсутствует.

Пользователь «admin» является единственным предустановленным пользователем в конфигурации устройства. Это приводит к определенным особенностям работы с ним:
1) Применение команды **no username admin** не удаляет пользователя «admin» из конфигурации, а приводит его к настройкам по умолчанию – паролю «password» и 15 уровню привилегий.
2) Отключить возможность авторизации пользователя «admin» можно командой **no admin login enable**.
3) Пользователь «admin» с настройками по умолчанию (пароль «password», уровень привилегий 15) не отображается в выводах команд **show running-config** и **show candidate-config** без модификатора «full».

Пример команд для создания нескольких учетных записей – пользователя «**netmaster**» с уровнем привилегий **15** для управления оборудованием, пользователя «**watcher**» с уровнем привилегий **1** для ограниченного просмотра оперативной информации, а также пользователя «**techsup**» для отладки устройства совместно с сотрудниками технической поддержки:

```
esbc# configure
esbc(config)# username netmaster
esbc(config-user)# password P@ssW0rd
esbc(config-user)# privilege 15
esbc(config-user)# exit
esbc(config)# username watcher
esbc(config-user)# password password
esbc(config-user)# privilege 1
esbc(config-user)# exit
esbc(config)# username techsup
esbc(config-user)# password PsWdTs
esbc(config-user)# mode techsupport
esbc(config-user)# exit
esbc(config)#
```

Назначение имени устройства

Для назначения имени устройства используются следующие команды:

```
esbc# configure
esbc(config)# hostname <new-name>
```

После применения конфигурации приглашение командной строки изменится на значение, заданное параметром **<new-name>**.

Настройка параметров публичной сети

Для настройки сетевого интерфейса пограничного контроллера сессий в публичной сети необходимо назначить устройству параметры, определённые провайдером сети – IP-адрес, маска подсети и адрес шлюза по умолчанию.

Пример команд настройки статического IP-адреса для саб-интерфейса **Gigabit Ethernet 1/0/2.150** для доступа к устройству через **VLAN 150**.

Параметры интерфейса:

- IP-адрес – 192.168.16.144;
- Маска подсети – 255.255.255.0;
- IP-адрес шлюза по умолчанию – 192.168.16.1.

```
esbc# configure
esbc(config)# interface gigabitethernet 1/0/2.150
esbc(config-if-sub)# ip address 192.168.16.144/24
esbc(config-if-sub)# exit
esbc(config)# ip route 0.0.0.0/0 192.168.16.1
```

Для того чтобы убедиться, что адрес был назначен интерфейсу, после применения конфигурации введите следующую команду:

```
esbc# show ip interfaces
IP address                    Interface              Admin   Link    Type    Precedence
---------------------------   --------------------   -----   -----   -----   ---------
192.168.16.144/24             gi1/0/2.150            Up      Up      static  primary
```

Провайдер может использовать динамически назначаемые адреса в своей сети. Для получения IP-адреса может использоваться протокол DHCP, если в сети присутствует сервер DHCP.

Пример настройки, предназначенной для получения динамического IP-адреса от DHCP-сервера на интерфейсе **Gigabit Ethernet 1/0/10**:

```
esbc# configure
esbc(config)# interface gigabitethernet 1/0/10
esbc(config-if)# ip address dhcp
esbc(config-if)# exit
```

Для того чтобы убедиться, что адрес был назначен интерфейсу, введите следующую команду после применения конфигурации:

```
esbc# show ip interfaces
IP address                  Interface              Admin   Link    Type   Precedence
------------------------    --------------------   -----   -----   ----   ----------
192.168.11.5/25             gi1/0/10               Up      Up      DHCP   --
```

Настройка удаленного доступа к устройству

В заводской конфигурации ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350 разрешен удаленный доступ к устройству по протоколу SSH из зоны **«trusted»**. Для того чтобы разрешить удаленный доступ к устройству из других зон, например, из публичной сети, необходимо создать соответствующие правила в firewall.

При конфигурировании доступа к устройству правила создаются для пары зон:

- **source-zone** – зона, из которой будет осуществляться удаленный доступ;
- **self** – зона, в которой находится интерфейс управления устройством.

Для создания разрешающего правила используются следующие команды:

```
esbc# configure
esbc(config)# security zone-pair <source-zone> self
esbc(config-zone-pair)# rule <number>
esbc(config-zone-rule)# action permit
esbc(config-zone-rule)# match protocol tcp
esbc(config-zone-rule)# match source-address object-group network <network object-group>
esbc(config-zone-rule)# match destination-address object-group network <network object-group>
esbc(config-zone-rule)# match destination-port object-group <service object-group>
esbc(config-zone-rule)# enable
esbc(config-zone-rule)# exit
esbc(config-zone-pair)# exit
```

Пример команд для разрешения пользователям из зоны «**untrusted**» с IP-адресами **132.16.0.5-132.16.0.10** подключаться к устройству с IP-адресом **40.13.1.22** по протоколу SSH:

```
esbc# configure
esbc(config)# object-group network clients
esbc(config-addr-set)# ip address-range 132.16.0.5-132.16.0.10
esbc(config-addr-set)# exit
esbc(config)# object-group network gateway
esbc(config-addr-set)# ip address-range 40.13.1.22
esbc(config-addr-set)# exit
esbc(config)# object-group service ssh
esbc(config-port-set)# port-range 22
esbc(config-port-set)# exit
esbc(config)# security zone-pair untrusted self
esbc(config-zone-pair)# rule 10
esbc(config-zone-rule)# action permit
esbc(config-zone-rule)# match protocol tcp
esbc(config-zone-rule)# match source-address object-group network clients
esbc(config-zone-rule)# match destination-address object-group network gateway
esbc(config-zone-rule)# match destination-port object-group ssh
esbc(config-zone-rule)# enable
esbc(config-zone-rule)# exit
esbc(config-zone-pair)# exit
```

6 Обновление программного обеспечения

- Создание резервной копии текущей конфигурации
  - Подготовка
  - Копирование файла резервной копии конфигурации
    - С использованием протоколов удаленного копирования файлов
    - На локально подключенный USB/MMC-носитель
    - С использованием web-интерфейса
- Восстановление конфигурации из резервной копии
  - Подготовка
  - Копирование файла с резервной копией конфигурации
    - С использованием протоколов удаленного копирования файлов
    - С локально подключенного USB/MMC-носителя
    - С использованием web-интерфейса
  - Применение и подтверждение загруженной конфигурации
- Обновление программного обеспечения средствами системы
  - Обновление программного обеспечения до версии 1.9.0 при последовательном обновлении с предыдущих версий (только для vESBC)
  - Обновление программного обеспечения через CLI
    - Подготовка к загрузке ПО
    - Загрузка ПО
    - Выбор образа ПО обновленной версии для следующей загрузки
    - Перезагрузка ESBC
- Обновление программного обеспечения через web-интерфейс
- Обновление программного обеспечения с использованием образа ПО .iso (только для vESBC)
- Обновление программного обеспечения из начального загрузчика (ESBC-15R/31/3200/3250/3350)
- Обновление вторичного загрузчика (U-Boot) (ESBC-15R/31/3200/3250/3350)

6.1 Создание резервной копии текущей конфигурации

Перед началом работ по обновлению ПО на пограничном контроллере сессий ESBC необходимо сделать резервную копию текущей конфигурации.

Копирование текущей конфигурации с ESBC осуществляется:

1. на удаленный сервер, с использованием протоколов удаленного копирования файлов (scp, tftp, ftp, sftp);
2. на USB/MMC-носители, подключенные локально;
3. с использованием web-интерфейса (см. раздел Меню «Работа с файлами конфигурации» справочника ESBC-Series. Управление через web-интерфейс)

Ниже представлено описание процесса создания резервной копии текущей конфигурации способами 1 и 2.

6.1.1 Подготовка

Для создания резервной копии текущей конфигурации ESBC с использованием серверов удаленного копирования файлов необходимо:

1. Запустить соответствующий сервер на ПК/сервере в сети.
2. Обеспечить возможность сохранения файлов в рабочем разделе сервера.
3. Обеспечить IP-связность между обновляемым ESBC и сервером удаленного копирования файлов (маршрутизация).
4. Обеспечить работу протокола удаленного копирования между ESBC и сервером удаленного копирования файлов (промежуточные firewall).
5. При необходимости (для протоколов ftp, sftp, scp) узнать имя пользователя и пароль для записи необходимого файла.

Для создания резервной копии текущей конфигурации ESBC на локально подключенный USB/MMC-носитель необходимо выполнить следующие условия:

1. Раздел USB/MMC-носителя должен быть отформатирован в формате FAT32.
2. Подключить USB/MMC-носитель в соответствующий слот ESBC.

6.1.2 Копирование файла резервной копии конфигурации

С использованием протоколов удаленного копирования файлов

В зависимости от протокола удаленного копирования файлов, в CLI ESBC необходимо выполнить одну из следующих команд:

Резервное копирование конфигурации по протоколу tftp

```
esbc# copy system:running-config tftp://<tftp-server-ip>:/<config-file-name>
```

Резервное копирование конфигурации по протоколу ftp

```
esbc# copy system:running-config ftp://<ftp-username>:<ftp-userpassword>@<ftp-server-
ip>:/<config-file-name>
```

Резервное копирование конфигурации по протоколу sftp

```
esbc# copy system:running-config sftp://<sftp-username>:<sftp-userpassword>@<sftp-server-ip>:/<config-file-name>
```

Резервное копирование конфигурации по протоколу scp

```
esbc# copy system:running-config scp://<scp-username>:<scp-userpassword>@<scp-server-ip>:/<config-file-name>
```

- <config-file-name> — имя файла, с которым будет сохранена текущая конфигурация сервисного маршрутизатора;
- <tftp-server-ip> — IP-адрес используемого TFTP-сервера;
- <ftp-username> — имя пользователя на FTP-сервере;
- <ftp-userpassword> — пароль пользователя на FTP-сервере;
- <ftp-server-ip> — IP-адрес используемого FTP-сервера;
- <sftp-username> — имя пользователя на SFTP-сервере;
- <sftp-userpassword> — пароль пользователя на SFTP-сервере;
- <sftp-server-ip> — IP-адрес используемого SFTP-сервера;
- <scp-username> — имя пользователя на SCP-сервере;
- <ftp-userpassword> — пароль пользователя на FTP-сервере;
- <scp-server-ip> — IP-адрес используемого SCP-сервера.

На локально подключенный USB/MMC-носитель

1. Определить метку тома подключенного USB/MMC-накопителя:

Определение имени метки тома на USB-накопителе

```
esbc# show storage-devices usb

Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<USB_DISK>                      vfat        7664.01     6391.69     1272.32
```

Определение имени метки тома на MMC-накопителе

```
esbc# show storage-devices mmc
Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<MMC_DISK>                      vfat        7664.01     6391.69     1272.32
```

2. Скопировать файл на используемый USB/MMC-накопитель:

> При выполнении команд копирования на USB/MMC-носители необходимо вместо полей <USB_DISK> или <MMC_DISK> использовать настоящие метки тома, определенные при выполнении пункта 1.

Резервное копирование конфигурации на USB-носитель

```
esbc# copy system:running-config usb://<USB_DISK>:/<config-file-name>

|****************************************| 100% (576B) Success!
```

Резервное копирование конфигурации на MMC-носитель

```
esbc# copy system:running-config mmc://<MMC_DISK>:/<config-file-name>

|****************************************| 100% (576B) Success!
```

- <config-file-name> — имя файла, с которым будет сохранена текущая конфигурация сервисного маршрутизатора;
- <USB_DISK> — имя раздела на USB-носителе;
- <MMC_DISK> — имя раздела на MMC-носителе.

С использованием web-интерфейса

Процесс получения файла конфигурации (running-config) через web-интерфейс описан в разделе Меню «Работа с файлами конфигурации» справочника ESBC-Series. Управление через web-интерфейс.

6.2 Восстановление конфигурации из резервной копии

В случае потери конфигурации на ESBC в процессе эксплуатации, обновления или "отката" на более старую версию ПО, конфигурацию ESBC можно восстановить, используя созданную ранее резервную копию.

Копирование резервной копии конфигурации на ESBC возможно как с использованием протоколов удаленного копирования файлов, так и с помощью локально подключенных USB/MMC-носителей.

При переходе с более новой версии ПО на более старую (downgrade) вероятна ситуация, когда более старая версия ПО не сможет применить конфигурацию, сохраненную в более новой версии. В результате конфигурация будет утеряна и ESBC загрузится с пустой конфигурацией. При пустой конфигурации к ESBC можно подключиться только используя консольное подключение и логин/пароль по умолчанию (**admin**/**password**).

Если в конфигурации ESBC используются медиапрофили, в которых были удалены автоматически заполненные шаблоны вида codec allow <pattern>, то при восстановлении конфигурации из файла, эти шаблоны будут заполнены заново. После восстановления конфигурации требуется повторное удаление таких шаблонов.

6.2.1 Подготовка

Для восстановления конфигурации ESBC из резервной копии с использованием серверов удаленного копирования файлов необходимо:

1. Запустить соответствующий сервер на ПК/сервере в сети.
2. Разместить в рабочем разделе сервера файл с созданной ранее резервной копией ESBC.
3. Настроить ESBC для появления IP-связности с сервером удаленного копирования файлов.
4. Обеспечить IP-связность между обновляемым ESBC и сервером удаленного копирования файлов (маршрутизация).
5. Обеспечить работу протокола удаленного копирования между ESBC и сервером удаленного копирования файлов (промежуточные firewall).
6. При необходимости (для протоколов ftp, sftp, scp) узнать имя пользователя и пароль для скачивания необходимого файла.

Для восстановления конфигурации ESBC из резервной копии с локально подключенного USB/MMC-носителя необходимо:

1. Раздел USB/MMC-носителя должен быть отформатирован в формате FAT32.
2. На USB/MMC-носителе должен быть помещен файл с ранее созданной резервной копией конфигурации ESBC.
3. Подключить USB/MMC-носитель в соответствующий слот ESBC.

6.2.2 Копирование файла с резервной копией конфигурации

С использованием протоколов удаленного копирования файлов

В зависимости от протокола удаленного копирования файлов в CLI ESBC необходимо выполнить одну из следующих команд:

Резервное копирование конфигурации по протоколу tftp

```
esbc# copy tftp://<tftp-server-ip>:/<config-file-name> system:candidate-config
```

Резервное копирование конфигурации по протоколу ftp

```
esbc# copy ftp://<ftp-username>:<ftp-userpassword>@<ftp-server-ip>:/<config-file-name>
system:candidate-config
```

Резервное копирование конфигурации по протоколу sftp

```
esbc# copy sftp://<sftp-username>:<sftp-userpassword>@<sftp-server-ip>:/<config-file-
name> system:candidate-config
```

Резервное копирование конфигурации по протоколу scp

```
esbc# copy scp://<scp-username>:<scp-userpassword>@<scp-server-ip>:/<config-file-name>
system:candidate-config
```

Резервное копирование конфигурации по протоколу http

```
esbc# copy http://<http-username>:<http-userpassword>@<http-server-ip>:/<config-file-
name> system:candidate-config
```

- <config-file-name> — имя файла резервной копии конфигурации сервисного маршрутизатора.
- <tftp-server-ip> — IP-адрес используемого TFTP-сервера.
- <ftp-username> — имя пользователя на FTP-сервере.
- <ftp-userpassword> — пароль пользователя на FTP-сервере.
- <ftp-server-ip> — IP-адрес используемого FTP-сервера.
- <sftp-username> — имя пользователя на SFTP-сервере.
- <sftp-userpassword> — пароль пользователя на SFTP-сервере.
- <sftp-server-ip> — IP-адрес используемого SFTP-сервера.
- <scp-username> — имя пользователя на SCP-сервере.
- <ftp-userpassword> — пароль пользователя на FTP-сервере.
- <scp-server-ip> — IP-адрес используемого SCP-сервера.
- <http-username> — имя пользователя на HTTP-сервере.
- <http-userpassword> — пароль пользователя на HTTP-сервере.
- <http-server-ip> — IP-адрес используемого HTTP-сервера.

С локально подключенного USB/MMC-носителя

1. Определить метку тома подключенного USB/MMC-накопителя:

Определение имени метки тома на USB-накопителе

```
esbc# show storage-devices usb
Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<USB_DISK>                      vfat        7664.01     6391.69     1272.32
```

Определение имени метки тома на MMC-накопителе

```
esbc# show storage-devices mmc
Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<MMC_DISK>                      vfat        7664.01     6391.69     1272.32
```

2. Скопировать файл на используемый USB/MMC-накопитель:

> При выполнении команд копирования на USB/MMC-носители необходимо вместо полей <USB_DISK> или <MMC_DISK> использовать настоящие метки тома, определенные при выполнении пункта 1.

Резервное копирование конфигурации на USB-носитель

```
esbc# copy usb://<USB_DISK>:/<config-file-name> system:candidate-config

|****************************************| 100% (576B) Success!
```

Резервное копирование конфигурации на MMC-носитель

```
esbc# copy mmc://<MMC_DISK>:/<config-file-name> system:candidate-config

|****************************************| 100% (576B) Success!
```

- <config-file-name> — имя файла резервной копии конфигурации ESBC;
- <USB_DISK> — имя раздела на USB-носителе;
- <MMC_DISK> — имя раздела на MMC-носителе.

С использованием web-интерфейса

Процесс загрузки файла резервной копии конфигурации через web-интерфейс описан в разделе Меню «Работа с файлами конфигурации» справочника ESBC-Series. Управление через web-интерфейс.

6.2.3 Применение и подтверждение загруженной конфигурации

Для применения и подтверждения работы конфигурации, загруженной ранее в раздел "system:candidate-config", необходимо выполнить команды:

Резервное копирование конфигурации на MMC-носитель

```
esbc# commit
Configuration has been successfully applied and saved to flash. Commit timer started,
changes will be.

esbc# confirm
Configuration has been confirmed. Commit timer canceled.
```

6.3 Обновление программного обеспечения средствами системы

ПО текущей версии является кумулятивным (содержит обновленные версии первичного и вторичного загрузчиков (только для ESBC-15R/31/3200/3250/3350)), поэтому будет достаточно:

- Загрузить ПО (firmware-файл) на ESBC.
- Выбрать образ ПО обновленной версии для следующей загрузки.

Отключение питания до окончания выполнения команды *boot system {image-1|image-2}* может привести к неисправности ESBC.

- Перезагрузить ESBC.

В рамках кумулятивного обновления ESBC-15R/31/3200/3250/3350 загрузчики могут не обновляться, если между установленным и устанавливаемыми загрузчиками нет различий. В этом случае будет получено следующее сообщение:

```
Boot image set successfully.
Skip due to same versions: xload, uboot
```

Сравнение выполняется устройством самостоятельно при выполнении кумулятивного обновления.

При обновлении программного обеспечения конфигурация пограничного контроллера сессий конвертируется в соответствии с новой версией. При загрузке пограничного контроллера сессий с более старой версией программного обеспечения, чем загруженная ранее, конфигурация не конвертируется и впоследствии удаляется. Перед обновлением ПО необходимо сделать резервную копию текущей конфигурации.

6.3.1 Обновление программного обеспечения до версии 1.9.0 при последовательном обновлении с предыдущих версий (только для vESBC)

Перечисленные ниже способы обновления ПО для vESBC **применимы только при обновлении с версии 1.6.0 и более поздних, при условии что версия 1.6.0 была установлена изначально**.

Если изначально была установлена более ранняя версия, то обновление до версии 1.9.0 следует производить путем полной переустановки ПО, описанным в разделе Процесс установки vESBC.

При переустановке ПО все пользовательские данные, включая конфигурацию будут потеряны. Необходимо сделать резервную копию текущей конфигурации.

После установки ПО 1.9.0 следует выполнить минимальные настройки сети для загрузки файла резервной копии на vESBC.

Процессы создания резервной копии конфигурации и ее восстановления описаны выше.

6.3.2 Обновление программного обеспечения через CLI

Подготовка к загрузке ПО

При загрузке ПО с использованием серверов удаленного копирования файлов необходимо:

1. Запустить соответствующий сервер в сети (tftp/ftp/sftp/http/https/scp).
2. Скопировать файл ПО (<firmware-file>) в рабочий раздел сервера удаленной загрузки файлов.
3. Обеспечить IP-связность между обновляемым ESBC и сервером удаленного копирования файлов (маршрутизация).
4. Обеспечить работу протокола удаленного копирования между ESBC и сервером удаленного копирования файлов (промежуточные firewall).
5. При необходимости (для протоколов ftp, sftp, scp, http, https) узнать имя пользователя и пароль для скачивания необходимого файла.

При загрузке ПО с использованием USB/MMC-носителя необходимо:

1. Раздел USB/MMC-носителя должен быть отформатирован в формате FAT32 или exFAT.
2. Скопировать файл ПО (<firmware-file>) в корневой раздел USB/MMC-носителя.
3. Подключить USB/MMC-носитель в соответствующий слот ESBC.
4. Определить метку тома подключенного USB/MMC-накопителя.

Загрузка ПО

С использованием одного из протоколов удаленной загрузки файлов

Загрузка ПО по протоколу tftp

```
esbc# copy tftp://<tftp-server-ip>:/<firmware-file> system:firmware
|****************************************| 100% (0B) Firmware updated successfully.
```

Загрузка ПО по протоколу ftp

```
esbc# copy ftp://<ftp-username>:<ftp-userpassword>@<ftp-server-ip>:/<firmware-file>
system:firmware

|****************************************| 100% (0B) Firmware updated successfully.
```

Загрузка ПО по протоколу sftp

```
esbc# copy sftp://<sftp-username>:<sftp-userpassword>@<sftp-server-ip>:/<firmware-
file> system:firmware

|****************************************| 100% (0B) Firmware updated successfully.
```

Загрузка ПО по протоколу scp

```
esbc# copy scp://<scp-username>:<scp-userpassword>@<scp-server-ip>:/<firmware-file>
system:firmware

|****************************************| 100% (0B) Firmware updated successfully.
```

Загрузка ПО по протоколу http

```
esbc# copy http://<http-username>:<http-userpassword>@<http-server-ip>:/<firmware-
file> system:firmware

|****************************************| 100% (0B) Firmware updated successfully.
```

Загрузка ПО по протоколу https

```
esbc# copy https://<https-username>:<https-userpassword>@<http-server-ip>:/<firmware-
file> system:firmware

|****************************************| 100% (0B) Firmware updated successfully.
```

- <tftp-server-ip> — IP-адрес используемого TFTP-сервера;
- <ftp-username> — имя пользователя на FTP-сервере;
- <ftp-userpassword> — пароль пользователя на FTP-сервере;
- <ftp-server-ip> — IP-адрес используемого FTP-сервера;
- <sftp-username> — имя пользователя на SFTP-сервере;
- <sftp-userpassword> — пароль пользователя на SFTP-сервере;
- <sftp-server-ip> — IP-адрес используемого SFTP-сервера;
- <scp-username> — имя пользователя на SCP-сервере;
- <ftp-userpassword> — пароль пользователя на FTP-сервере;
- <scp-server-ip> — IP-адрес используемого SCP-сервера;
- <http-username> — имя пользователя на HTTP-сервере;
- <http-userpassword> — пароль пользователя на HTTP-сервере;
- <http-server-ip> — IP-адрес используемого HTTP-сервера.

С использованием USB/MMC-накопителя

1. Определение имени метки тома подключенного USB/MMC-накопителя:

Определение имени метки тома на USB-накопителе

```
esbc# show storage-devices usb
Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<USB_DISK>                      vfat        7664.01     6391.69     1272.32
```

Определение имени метки тома на MMC-накопителе

```
esbc# show storage-devices mmc
Name                            Filesystem  Total, MB   Used, MB    Free, MB
------------------------------  ----------  ----------  ----------  ----------
<MMC_DISK>                      vfat        7664.01     6391.69     1272.32
```

2. Копирование файла с используемого USB/MMC-накопителя:

> При выполнении команд копирования с USB/MMC-носителей необходимо вместо полей <USB_DISK> или <MMC_DISK> использовать настоящие метки тома, определенные выше.

Загрузка ПО с USB-носителя

```
esbc# copy usb://<USB_DISK>:/<firmware-file> system:firmware

|****************************************| 100% (73786kB) Firmware updated
successfully
```

Загрузка ПО с MMC-носителя

```
esbc# copy mmc://<MMC_DISK>:/<firmware-file> system:firmware

|****************************************| 100% (73786kB) Firmware updated
successfully.
```

- <USB_DISK> – имя раздела на USB-носителе;
- <MMC_DISK> – имя раздела на MMC-носителе.

Выбор образа ПО обновленной версии для следующей загрузки

На ESBC одновременно хранятся два образа ПО (image-1 и image-2).

1. Проверить содержимое образов ПО, загруженных на ESBC:

```
esbc# show bootvar
Image   Version                     Date                   Status         After reboot

-----   -------------------------   --------------------   ------------   ------------

1       1.44.0 build                2026-08-31 14:43:29    Not Active

        26[99302781f1]


2       1.38.1 build                2026-01-27 10:46:08    Active         *

        2[e229e4b49a]
```

При загрузке файла ПО в раздел system:firmware загрузка осуществляется всегда в неактивный (`Not Active`) в данный момент раздел.

2. Выбрать раздел, содержащий ПО обновленной версии, в качестве загрузочного:

Выбор раздела ПО для загрузки

```
esbc# boot system image-1
This command cannot be interrupted, do not turn off device during process.
Continue? (y/N): y
2000-01-07T18:51:19+00:00 %FILE_MGR-I-INFO: operation started: 'boot system
image-1' (index: 4, origin: CLI)
2000-01-07T18:51:22+00:00 %FIRMWARE-I-INFO: Writing data...
2000-01-07T18:51:31+00:00 %FIRMWARE-I-INFO: Writing data...
2000-01-07T18:51:37+00:00 %FILE_MGR-I-INFO: operation is finished: 'boot system
image-1' (index: 4, origin: CLI)
Boot image set successfully.
```

или

Выбор неактивного ПО для загрузки

```
esbc# boot system inactive
This command cannot be interrupted, do not turn off device during process.
Continue? (y/N): y
1970-02-04T20:52:43+00:00 %FILE_MGR-I-INFO: operation started: 'boot system
image-1' (index: 3, origi)
1970-02-04T20:52:45+00:00 %FIRMWARE-I-INFO: Writing data...
1970-02-04T20:52:55+00:00 %FIRMWARE-I-INFO: Writing data...
1970-02-04T20:53:09+00:00 %FILE_MGR-I-INFO: operation is finished: 'boot system
image-1' (index: 3, o)
Boot image set successfully.
Successfully updated (bootloader's directory is dirty): bl1, uboot
```

Запрещается отключение питания ESBC в момент выполнения команд *boot system {image-1|image-2}* или *boot system inactive*.
Отключение питания до окончания выполнения команд *boot system {image-1|image-2}* или *boot system inactive* может привести к неисправности ESBC.

3. Проверить, что образ, содержащий ПО обновленной версии, выбран для загрузки:

```
esbc# show bootvar
Image   Version                      Date                   Status         After reboot

-----   -------------------------    --------------------   ------------   ------------

1       1.44.0 build                 2026-08-31 14:43:29    Not Active     *

        26[99302781f1]


2       1.38.1 build                 2026-01-27 10:46:08    Active

        2[e229e4b49a]
```

Перезагрузка ESBC

Перезагрузить ESBC при помощи команды:

Перезагрузка маршрутизатора в CLI основного ПО

```
esbc# reload system

Do you really want to reload system ? (y/N): y
```

6.4 Обновление программного обеспечения через web-интерфейс

Описание процесса обновления программного обеспечения приведено в разделе Меню «ПО устройства» справочника ESBC-Series. Управление через web-интерфейс.

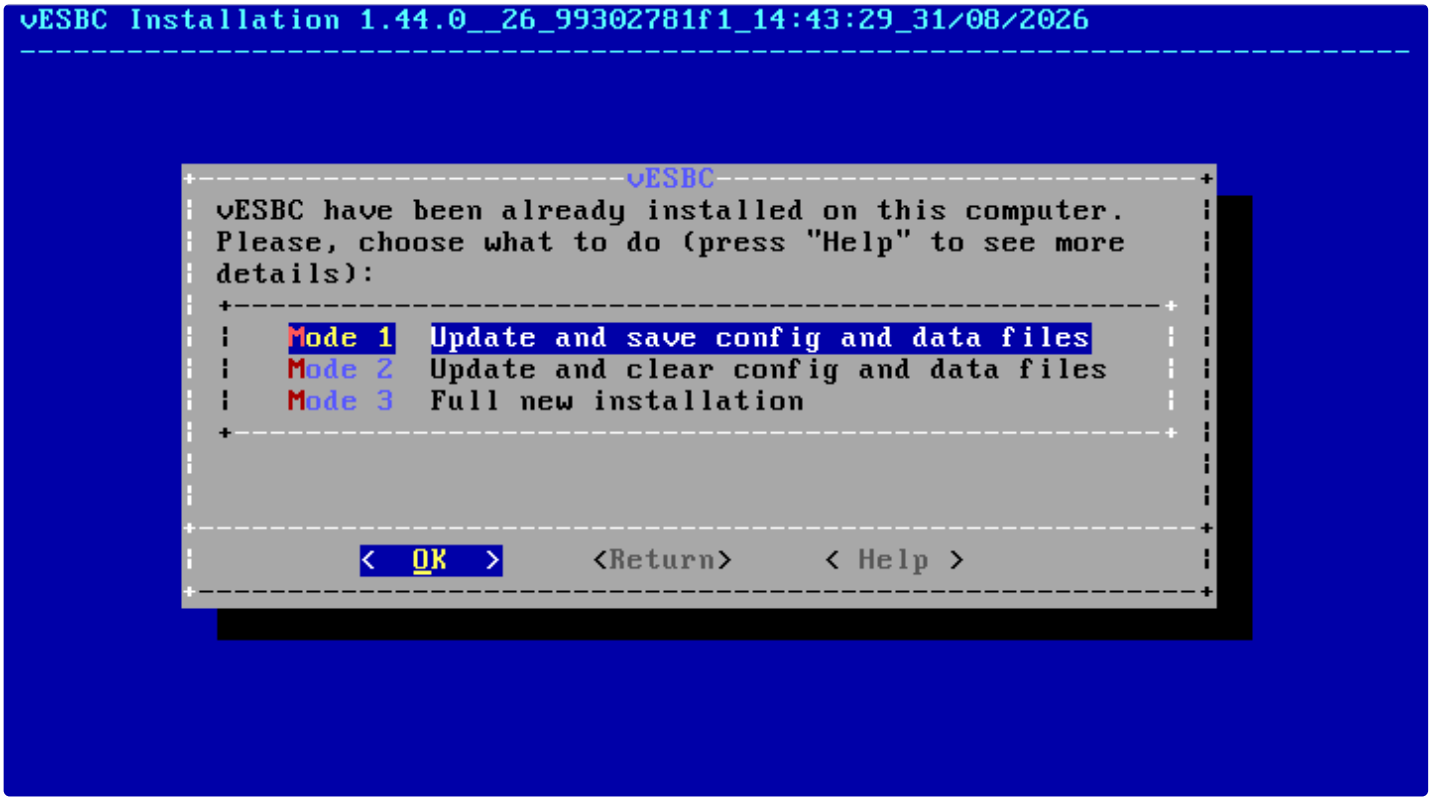
6.5 Обновление программного обеспечения с использованием образа ПО .iso (только для vESBC)

Процесс обновления программного обеспечения с помощью образа ПО .iso выполняется аналогично первоначальной инсталляции vESBC, описанной в разделе Процесс установки vESBC.

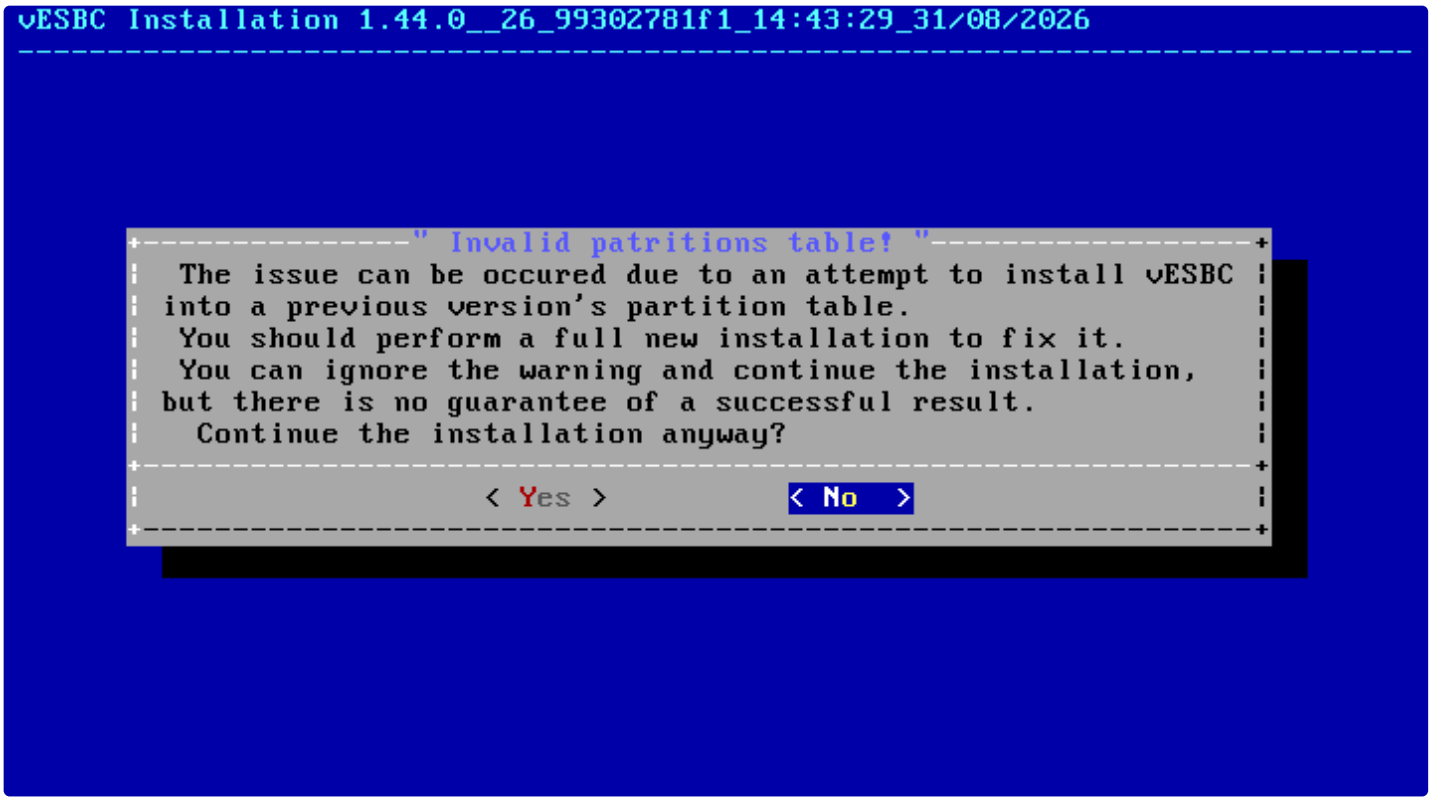
Для обновления необходимо выполнить **шаги 1-3** согласно процедуре первоначальной инсталляции vESBC и перейти на шаги описанные ниже.

Шаг 4. В появившемся окне необходимо выбрать:

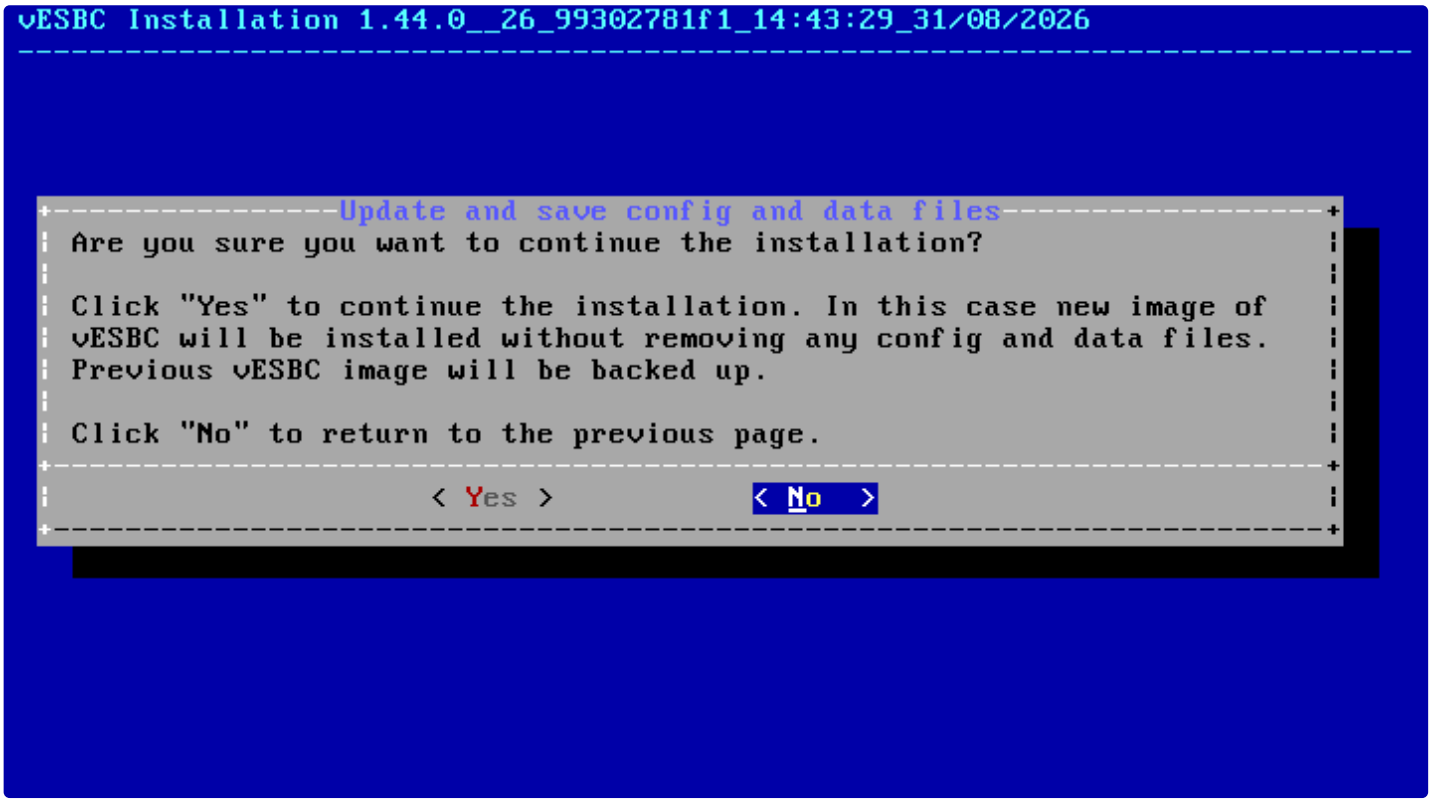
- или пункт "***Mode 1***" для обновления программного обеспечения и **сохранения** конфигурации и пользовательских файлов.
- или пункт "***Mode 2***" для обновления программного обеспечения и **удаления** конфигурации и пользовательских файлов.



Шаг 5. Подтвердить установку в раздел с разметкой предыдуших версий ПО, выбрав ***"Yes"***:



Шаг 6. В случае если на каком-либо этапе была допущена ошибка, возможно вернуться на предыдущие страницы, выбрав ***"No"***. Если ошибок допущено не было, можно продолжить обновление vESBC, выбрав ***"Yes"***:



Также следует пропустить **шаги с 5 по 8** из раздела Процесс установки vESBC, если при обновлении не требуется изменять серийный номер vESBC.

Порядок обновления:

1. Смонтировать файл .iso в CD-привод гипервизора и установить загрузку с образа.
2. Перезагрузить виртуальную машину ESBC.
3. Выполнить обновление в соответствии с описанием выше.
4. После установки убрать образ из CD-привода гипервизора.
5. Перезагрузить виртуальную машину ESBC.

6.6 Обновление программного обеспечения из начального загрузчика (ESBC-15R/31/3200/3250/3350)

Программное обеспечение пограничного контроллера сессий можно обновить из начального загрузчика следующим образом:

1. Остановите загрузку после окончания инициализации пограничного контроллера сессий загрузчиком U-Boot, нажав клавишу ***<Esc>***.

```
Configuring PoE...
distribution 1 dest_threshold 0xa drop_timer 0x0
Configuring POE in bypass mode
NAE configuration done!
initializing port 0, type 2.
initializing port 1, type 2.
SMC Endian Test:b81fb81f
nae-0, nae-1
  =======Skip: Load SYS UCORE for old 8xxB1/3xxB0 revision on default.
Hit any key to stop autoboot:  2
```

2. Укажите IP-адрес TFTP-сервера:

```
BRCM.XLP316Lite Rev B0.u-boot# serverip 10.100.100.1
```

3. Укажите IP-адрес пограничного контроллера сессий:

```
BRCM.XLP316Lite Rev B0.u-boot# ipaddr 10.100.100.2
```

4. Укажите имя файла программного обеспечения на TFTP-сервере:

```
BRCM.XLP316Lite Rev B0.u-boot# firmware_file firmware
```

5. Можно сохранить окружение командой *saveenv* для будущих обновлений.

6. Запустите процедуру обновления программного обеспечения:

```
BRCM.XLP316Lite Rev B0.u-boot# run tftp_update_image1
Using nae-0-3 device
TFTP from server 10.100.100.1; our IP address is 10.100.100.2
Filename 'esbc3200/firmware'.
Load address: 0xa800000060000000
Loading: TftpStart:TftpTimeoutMsecs = 10000, TftpTimeoutCountMax = 6
#################################################################
#################################################################
#################################################################
#########################
####################################
done
Bytes transferred = 64453909 (3d77d15 hex)
Device 0: MT29F8G08ABBCAH4 ... is now current device

NAND erase: device 0 offset 0x1440000, size 0x6400000
Bad block table found at page 262080, version 0x01
Bad block table found at page 262016, version 0x01
Erasing at 0x7800000 -- 1895825408% complete..
OK

NAND write: device 0 offset 0x1440000, size 0x6400000
104857600 bytes written: OK
```

7. Установите загруженное программное обеспечение в качестве образа для запуска системы и перезагрузите ESBC:

```
BRCM.XLP316Lite Rev B0.u-boot# boot_system image1
BRCM.XLP316Lite Rev B0.u-boot# reset
```

6.7 Обновление вторичного загрузчика (U-Boot) (ESBC-15R/31/3200/3250/3350)

Вторичный загрузчик занимается инициализацией NAND и пограничного контроллера сессий. При обновлении новый файл вторичного загрузчика сохраняется на flash на месте старого.

Для просмотра текущей версии загрузочного файла, работающего на устройстве, введите команду *version* в CLI U-Boot, также версия отображается в процессе загрузки пограничного контроллера сессий:

```
BRCM.XLP316Lite Rev B0.u-boot# version
BRCM.XLP.U-Boot:1.1.0.47 (29/11/2016 - 19:00:24)
```

Процедура обновления ПО:

1. Остановите загрузку устройства после окончания инициализации пограничного контроллера сессий загрузчиком U-Boot, нажав клавишу ***<Esc>***.

```
Configuring PoE...
distribution 1 dest_threshold 0xa drop_timer 0x0
Configuring POE in bypass mode
NAE configuration done!
initializing port 0, type 2.
initializing port 1, type 2.
SMC Endian Test:b81fb81f
nae-0, nae-1
  =======Skip: Load SYS UCORE for old 8xxB1/3xxB0 revision on default.
Hit any key to stop autoboot:  2
```

2. Укажите IP-адрес TFTP-сервера:

```
BRCM.XLP316Lite Rev B0.u-boot# serverip 10.100.100.1
```

3. Укажите IP-адрес пограничного контроллера сессий:

```
BRCM.XLP316Lite Rev B0.u-boot# ipaddr 10.100.100.2
```

4. Укажите имя файла загрузчика на TFTP-сервере:

```
BRCM.XLP316Lite Rev B0.u-boot# uboot_file u-boot.bin
```

5. Можно сохранить окружение командой *saveenv* для будущих обновлений.
6. Запустите процедуру обновления программного обеспечения:

```
BRCM.XLP316LiteRevB0.u-boot# run tftp_update_uboot
Using nae-1 device
TFTP from server 10.100.100.1; our IP address is 10.100.100.2
Filename 'esbc3200/u-boot.bin'.
Load address: 0xa800000078020000
Loading: ##########################################################
done
Bytes transferred = 852648 (d02a8 hex)
SF: Detected MX25L12805D with page size 256, total 16777216 bytes
16384 KiB MX25L12805D at 0:0 is now current device
```

7. Перезагрузите пограничный контроллер сессий:

```
BRCM.XLP316Lite Rev B0.u-boot# reset
```

7 Рекомендации по безопасной настройке

- Общие рекомендации
- Настройка системы логирования событий
  - Рекомендации
  - Предупреждения
  - Пример настройки
- Настройка политики использования паролей
  - Рекомендации
  - Пример настройки
- Настройка политики AAA
  - Рекомендации
  - Предупреждения
  - Пример настройки
- Настройка удалённого управления
  - Рекомендации
  - Пример настройки
- Настройка механизмов защиты от сетевых атак
  - Рекомендации
  - Пример настройки

Рекомендации по безопасной настройке носят общий характер и подходят для большинства инсталляций. Настоящие рекомендации в значительной степени повышают безопасность эксплуатации устройства, но не являются исчерпывающими. В зависимости от схемы применения устройства необходимо настраивать и другие параметры безопасности. В некоторых специфических случаях выполнение данных рекомендаций может привести к неработоспособности сети. При настройке устройства стоит в первую очередь следовать техническим требованиям и регламентам сетей, в которых будет эксплуатироваться данное устройство.

7.1 Общие рекомендации

- Рекомендуется всегда отключать неиспользуемые физические интерфейсы с помощью команды *shutdown*. Команда подробно описана в разделе Конфигурирование и мониторинг интерфейсов справочника команд CLI.
- Рекомендуется всегда настраивать синхронизацию системных часов с доверенными источниками сетевого времени (NTP). Алгоритм настройки NTP приведён в разделе Настройка NTP руководства. Подробная информация о командах для настройки NTP приведена в разделе Управление системными часами справочника команд CLI.
- Рекомендуется отключать NTP broadcast client, включённый по умолчанию в заводской конфигурации.
- Не рекомендуется использовать команду *ip firewall disable*, отключающую межсетевое экранирование. Следует всегда назначать интерфейсам соответствующие зоны безопасности и настраивать корректные правила межсетевого экрана. Алгоритм настройки межсетевого экрана приведён в разделе Конфигурирование Firewall настоящего руководства. Подробная информация о командах для настройки межсетевого экрана приведена в разделе Управление Firewall справочника команд CLI.

> ⚠ Для передачи сигнального (SIP), медиа (RTP) трафика, обработки сообщений STUN при использовании локального STUN-сервера, а также при настройке http/https прокси, не требуется конфигурирование дополнительных правил и зон Firewall. Правила будут созданы автоматически при конфигурировании соответствующих SIP-транспортов, SIP-транков, абонентских интерфейсов, локальных STUN-серверов, http/https прокси-серверов. Подробная информация о настройке сервисов VoIP находится в разделе Управление ESBC, о настройке http/https прокси в подразделе "Проксирование HTTP/HTTPS-трафика" раздела Управление безопасностью.

7.2 Настройка системы логирования событий

Алгоритмы настройки системы логирования событий приведены в подразделе «Настройка Syslog» раздела Мониторинг настоящего руководства.

Подробная информация о командах для настройки системы логирования событий приведена в разделе Управление SYSLOG справочника команд CLI.

7.2.1 Рекомендации

- Рекомендуется настроить хранение сообщений о событиях в файл syslog на устройстве и передачу этих событий на внешний syslog-сервер.
- Рекомендуется ограничивать размер syslog-файла на устройстве.
- Рекомендуется настраивать ротацию syslog-файлов на устройстве.
- Рекомендуется включать нумерацию сообщений syslog.

7.2.2 Предупреждения

- Данные, хранящиеся в файловой системе **tmpsys:syslog**, не сохраняются при перезагрузке устройства. Этот тип файловой системы рекомендуется использовать для хранения оперативных логов.
- Не рекомендуется использовать файловую систему **flash:syslog** для хранения логов, так как это может привести к преждевременному выходу из строя устройства ESBC-15R/31/3200/3250/3350.
- Не рекомендуется включать отправку отладочных сообщений в console на нагруженной системе, это может приводить к высокой загрузке CPU и повлиять на работоспособность.

7.2.3 Пример настройки

Задача:

Настроить хранение сообщений о событиях уровня info и выше в файл syslog на устройстве и настроить передачу этих событий на внешний syslog-сервер. Ограничить файл размером 512 Кбайт. Включить ротацию 3 файлов. Включить нумерацию сообщений syslog.

Решение:

Настраиваем хранение syslog-сообщений в файле:

```
esbc(config)# syslog file tmpsys:syslog/default
esbc((config-syslog-file)# severity info
esbc((config-syslog-file)# exit
```

Настраиваем ограничение размера и ротацию файлов:

```
esbc(config)# syslog max-files 3
esbc(config)# syslog file-size 512
```

Настраиваем передачу сообщений на внешний сервер:

```
esbc(config)# syslog host mylog
esbc(config-syslog-host)# remote-address 92.168.1.2
esbc(config-syslog-host)# transport udp
esbc(config-syslog-host)# port 514
esbc(config-syslog-host)# severity info
esbc(config-syslog-host)# exit
```

Включаем нумерацию сообщений syslog:

```
esbc(config)# syslog sequence-numbers
```

7.3 Настройка политики использования паролей

Алгоритмы настройки политики AAA приведены в разделе Настройка AAA настоящего руководства.

Подробная информация о командах для настройки политики использования паролей приведена в разделе Настройка AAA справочника команд CLI.

7.3.1 Рекомендации

- Рекомендуется всегда включать требования на смену пароля по умолчанию пользователя admin.
- Рекомендуется ограничивать время жизни паролей и запрещать повторно использовать, как минимум, предыдущий пароль.
- Рекомендуется выставлять требования минимальной длины пароля больше 8 символов.
- Рекомендуется выставлять требования на использование строчных и прописных букв, цифр и спецсимволов.

7.3.2 Пример настройки

Задача:

- Настроить парольную политику с обязательным требованием смены пароля по умолчанию, временем действия пароля 1 месяц и запретом на использование 12 последних паролей.
- Задать минимальную длину пароля 16 символов, максимальную — 24 символа.
- Пароль должен содержать не менее 3 прописных букв, не менее 5 строчных букв, не менее 4 цифр и не менее 2 спецсимволов. Пароль в обязательном порядке должен содержать все 4 типа символов.

Решение:

Включаем запрос на смену пароля по умолчанию для пользователя admin:

```
esbc(config)# security passwords default-expired
```

Устанавливаем время жизни пароля 30 дней и запрет на использование предыдущих 12 паролей:

```
esbc(config)# security passwords lifetime 30
esbc(config)# security passwords history 12
```

Устанавливаем ограничения на длину пароля:

```
esbc(config)# security passwords min-length 16
esbc(config)# security passwords max-length 24
```

Устанавливаем ограничения по минимальному количеству символов соответствующих типов:

```
esbc(config)# security passwords upper-case 3
esbc(config)# security passwords lower-case 5
esbc(config)# security passwords special-case 2
esbc(config)# security passwords numeric-count 4
esbc(config)# security passwords symbol-types 4
```

7.4 Настройка политики AAA

Алгоритмы настройки политики AAA приведены в разделе Настройка AAA настоящего руководства.

Подробная информация о командах для настройки политики AAA приведена в разделе Настройка AAA справочника команд CLI.

7.4.1 Рекомендации

- Рекомендуется использовать ролевую модель доступа на устройство.
- Рекомендуется использовать персональные учетные записи для аутентификации на устройстве.
- Рекомендуется включать логирование вводимых пользователем команд.
- Рекомендуется использовать несколько методов аутентификации для входа на устройства через консоль, удалённого входа на устройства и повышения привилегий. Оптимальной считается комбинация из аутентификации по одному из протоколов RADIUS/TACACS/LDAP и локальной аутентификации.
- Рекомендуется понизить уровень привилегий встроенной учётной записи **admin**.
- Рекомендуется настроить логирование изменений локальных учётных записей.
- Рекомендуется настроить логирование изменений политики AAA.

7.4.2 Предупреждения

- Встроенную учётную запись **admin** удалить нельзя, только отключить авторизацию для неё командой **no admin login enable**.
- Команда **no username admin** не удаляет пользователя **admin**, сбрасывает его конфигурацию в значения по умолчанию. После применения этой команды пользователь **admin** не будет отображаться в конфигурации.
- Команда **no password** для пользователя **admin** также не удаляет пароль пользователя **admin**, а сбрасывает его в значение по умолчанию. После применения этой команды пароль пользователя **admin** перестаёт отображаться в конфигурации и становится ‘password'.
- Перед отключением авторизации для пользователя **admin** в конфигурацию устройства необходимо настроить пользователя с уровнем привилегий 15 или задать ENABLE-пароль для уровня привилегий 15.

7.4.3 Пример настройки

Задача:

Настроить политику AAA:

- Для удалённого входа по протоколу SSH использовать аутентификации через RADIUS.
- Для входа через локальную консоль использовать аутентификации через RADIUS, в случае отсутствия связи с RADIUS-серверами использовать локальную аутентификацию.
- Использовать ENABLE-пароль, заданный через RADIUS, в случае отсутствия связи с RADIUS-серверами использовать локальный ENABLE-пароль.
- Установить пользователю admin пониженный уровень привилегий.
- Настроить логирование изменений локальных учётных записей.
- Настроить логирование изменений политик AAA.
- Настроить логирование вводимых команд.

Решение:

Создаем локального пользователя **local-operator** с уровнем привилегий 8:

```
esbc(config)# username local-operator
esbc(config-user)# password Pa$$w0rd1
esbc(config-user)# privilege 8
esbc(config-user)# exit
```

Задаём локальный ENABLE-пароль:

```
esbc(config)# enable password $6e5c4r3e2t!
```

Далее необходимо отключить авторизацию у пользователя admin:

```
esbc(config)# no admin login enable
```

Настраиваем связь с двумя RADIUS-серверами, основным 192.168.1.11 и резервным 192.168.2.12:

```
esbc(config)# radius-server host 192.168.1.11
esbc(config-radius-server)# key ascii-text encrypted 8CB5107EA7005AFF
esbc(config-radius-server)# priority 100
esbc(config-radius-server)# exit
esbc(config)# radius-server host 192.168.2.12
esbc(config-radius-server)# key ascii-text encrypted 8CB5107EA7005AFF
esbc(config-radius-server)# priority 150
esbc(config-radius-server)# exit
```

Настраиваем политику AAA:

```
esbc(config)# aaa authentication login CONSOLE radius local
esbc(config)# aaa authentication login SSH radius
esbc(config)# aaa authentication enable default radius enable
esbc(config)# aaa authentication mode break
esbc(config)# line console
esbc(config-line-console)# login authentication CONSOLE
esbc(config-line-console)# exit
esbc(config)# line ssh
esbc(config-line-ssh)# login authentication SSH
esbc(config-line-ssh)# exit
```

Настраиваем логирование:

```
esbc(config)# logging userinfo
esbc(config)# logging aaa
esbc(config)# syslog cli-commands
```

7.5 Настройка удалённого управления

Подробная информация о командах настройки удалённого доступа приведена в разделе Настройка доступа SSH, Telnet справочника команд CLI.

7.5.1 Рекомендации

- Не рекомендуется включать удалённое управление по протоколу Telnet.
- Рекомендуется использовать криптостойкие алгоритмы аутентификации sha2-512 и отключить все остальные.
- Рекомендуется использовать криптостойкие алгоритмы шифрования aes256ctr и отключить все остальные.
- Рекомендуется использовать криптостойкий алгоритм обмена ключами шифрования dh-group-exchange-sha256 и отключить все остальные.
- Рекомендуется использовать криптостойкий алгоритм верификации Host-Key для SSH rsa и отключить все остальные.
- Рекомендуется разрешить доступ к удалённому управлению устройством только с определённых IP-адресов.
- Перед началом эксплуатации рекомендуется перегенерировать ключи шифрования.

7.5.2 Пример настройки

Задача:

Сгенерировать новые ключи шифрования. Использовать криптостойкие алгоритмы.

Решение:

Отключаем устаревшие и не криптостойкие алгоритмы:

```
esbc(config)# ip ssh server
esbc(config)# ip ssh authentication algorithm md5 disable
esbc(config)# ip ssh authentication algorithm md5-96 disable
esbc(config)# ip ssh authentication algorithm ripemd160 disable
esbc(config)# ip ssh authentication algorithm sha1 disable
esbc(config)# ip ssh authentication algorithm sha1-96 disable
esbc(config)# ip ssh authentication algorithm sha2-256 disable
esbc(config)# ip ssh encryption algorithm 3des disable
esbc(config)# ip ssh encryption algorithm aes128 disable
esbc(config)# ip ssh encryption algorithm aes128ctr disable
esbc(config)# ip ssh encryption algorithm aes192 disable
esbc(config)# ip ssh encryption algorithm aes192ctr disable
esbc(config)# ip ssh encryption algorithm aes256 disable
esbc(config)# ip ssh encryption algorithm arcfour disable
esbc(config)# ip ssh encryption algorithm arcfour128 disable
esbc(config)# ip ssh encryption algorithm arcfour256 disable
esbc(config)# ip ssh encryption algorithm blowfish disable
esbc(config)# ip ssh encryption algorithm cast128 disable
esbc(config)# ip ssh key-exchange algorithm dh-group-exchange-sha1 disable
esbc(config)# ip ssh key-exchange algorithm dh-group1-sha1 disable
esbc(config)# ip ssh key-exchange algorithm dh-group14-sha1 disable
esbc(config)# ip ssh key-exchange algorithm ecdh-sha2-nistp256 disable
esbc(config)# ip ssh key-exchange algorithm ecdh-sha2-nistp384 disable
esbc(config)# ip ssh key-exchange algorithm ecdh-sha2-nistp521 disable
esbc(config)# ip ssh host-key algorithm dsa disable
esbc(config)# ip ssh host-key algorithm ecdsa256 disable
esbc(config)# ip ssh host-key algorithm ecdsa384 disable
esbc(config)# ip ssh host-key algorithm ecdsa521 disable
esbc(config)# ip ssh host-key algorithm ed25519 disable
```

Генерируем новые ключи шифрования:

```
esbc# update ssh-host-key rsa 2048
```

7.6 Настройка механизмов защиты от сетевых атак

Алгоритмы настройки механизмов защиты от сетевых атак приведены в разделе Настройка логирования и защиты от сетевых атак.

Подробная информация о командах для настройки политики использования паролей приведена в разделе Управление логированием и защитой от сетевых атак справочника команд CLI.

7.6.1 Рекомендации

- Рекомендуется всегда включать защиту от ip spoofing.
- Рекомендуется всегда включать защиту от TCP-пакетов с неправильно выставленными флагами.
- Рекомендуется всегда включать защиту от фрагментированных TCP-пакетов с выставленным флагом SYN.
- Рекомендуется всегда включать защиту от фрагментированных ICMP-пакетов.
- Рекомендуется всегда включать защиту ICMP-пакетов большого размера.

- Рекомендуется всегда включать защиту от незарегистрированных IP-протоколов.
- Рекомендуется включать логирование механизма защиты от сетевых атак.

7.6.2 Пример настройки

Задача:

Настроить механизм защиты от сетевых атак в соответствии с рекомендациями.

Решение:

Включаем защиту от ip spoofing и логирование механизма защиты:

```
esbc(config)# ip firewall screen spy-blocking spoofing
esbc(config)# logging firewall screen spy-blocking spoofing
```

Включаем защиту от TCP-пакетов с неправильно выставленными флагами и логирование механизма защиты:

```
esbc(config)# ip firewall screen spy-blocking syn-fin
esbc(config)# logging firewall screen spy-blocking syn-fin
esbc(config)# ip firewall screen spy-blocking fin-no-ack
esbc(config)# logging firewall screen spy-blocking fin-no-ack
esbc(config)# ip firewall screen spy-blocking tcp-no-flag
esbc(config)# logging firewall screen spy-blocking tcp-no-flag
esbc(config)# ip firewall screen spy-blocking tcp-all-flags
esbc(config)# logging firewall screen spy-blocking tcp-all-flags
```

Включаем защиту от фрагментированных ICMP-пакетов и логирование механизма защиты:

```
esbc(config)# ip firewall screen suspicious-packets icmp-fragment
esbc(config)# logging firewall screen suspicious-packets icmp-fragment
```

Включаем защиту от ICMP-пакетов большого размера и логирование механизма защиты:

```
esbc(config)# ip firewall screen suspicious-packets large-icmp
esbc(config)# logging firewall screen suspicious-packets large-icmp
```

Включаем защиту от незарегистрированных IP-протоколов и логирование механизма защиты:

```
esbc(config)# ip firewall screen suspicious-packets unknown-protocols
esbc(config)# logging firewall screen suspicious-packets unknown-protocols
```

8 Примеры подключения ESBC к сети передачи данных

> ⚠ В данном разделе приведены примеры физического подключения ESBC к сети передачи данных. В примерах указан ESBC-3200, но схемы применимы и к ESBC-15R/31/3250/3350/vESBC.

После подключения и настройки сетевых интерфейсов (терминации IP-адресов), можно использовать эти интерфейсы для организации SIP-trunk/User-interface между сетями NET 1 и NET 2, в качестве которых, например, могут выступать публичная сеть Internet и локальная сеть предприятия.

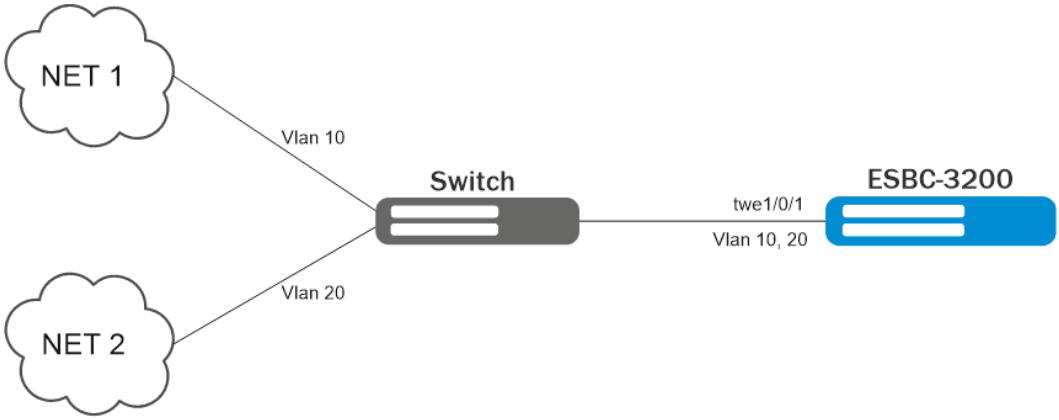
Команды и примеры настройки интерфейсов ESBC приведены в разделах Управление интерфейсами и Управление функциями второго уровня (L2) Руководства по эксплуатации, а также в разделах Управление L2-функциями и Конфигурирование и мониторинг интерфейсов Справочника команд CLI.

- Подключение к разным сетям с использованием двух сетевых интерфейсов
- Подключение к сети с использованием одного сетевого интерфейса
- Подключение к сети с использованием нескольких сетевых интерфейсов (резервирование линков)
  - Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал
  - Использование моста (Bridge) для терминации на уровне L3
- Подключение к нескольким коммутаторам с использованием нескольких сетевых интерфейсов (резервирование линков и узлов сети)
  - Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал
  - Использование моста (Bridge) для терминации на уровне L3
- Использование кластера

8.1 Подключение к разным сетям с использованием двух сетевых интерфейсов

При подключении к сети с использованием двух сетевых интерфейсов в разных сетях, следует перевести режим работы интерфейсов twe1/0/1 и twe1/0/2 в L3 (*mode routerport*) и назначить на них соответствующие IP-адреса. При использовании VLAN требуется сконфигурировать соответствующие саб-интерфейсы, например twe1/0/1.10 и twe1/0/1.20 и назначить на них соответствующие IP-адреса.

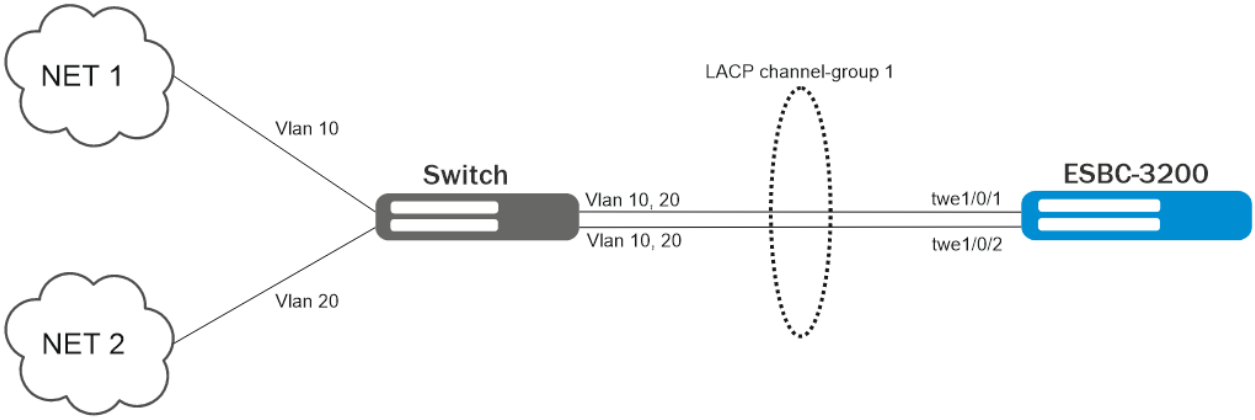
8.2 Подключение к сети с использованием одного сетевого интерфейса



При подключении к сети с использованием одного сетевого интерфейса ESBC-3200, следует перевести режим работы интерфейса twe1/0/1 в L3 (*mode routerport*). Для терминации VLAN 10 и VLAN 20 требуется сконфигурировать два саб-интерфейса twe1/0/1.10 и twe1/0/1.20 и назначить на них соответствующие IP-адреса.

На порту коммутатора (Switch) VLAN 10 и VLAN 20 необходимо передавать с тегами (mode trunk).

8.3 Подключение к сети с использованием нескольких сетевых интерфейсов (резервирование линков)



8.3.1 Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал

Для агрегации интерфейсов twe1/0/1 и twe1/0/2 необходимо включить их в одну группу (channel-group 1). Для терминации на уровне L3 следует использовать *interface port-channel 1*.

Терминация VLAN 10 и VLAN 20 осуществляется путем конфигурации двух саб-интерфейсов: *port-channel 1.10* и *port-channel 1.20*.

На коммутаторе также необходимо настроить протокол LACP.

8.3.2 Использование моста (Bridge) для терминации на уровне L3

Для терминации на L3 используется интерфейс bridge. Пример настройки:

```
esbc# configure
esbc(config)# bridge 10
esbc(config-bridge)# vlan 10
esbc(config-bridge)# ip address 192.168.1.1/24
esbc(config-bridge)# exit
esbc(config)# bridge 20
esbc(config-bridge)# vlan 20
esbc(config-bridge)# ip address 192.168.2.1/24
esbc(config-bridge)# exit
```

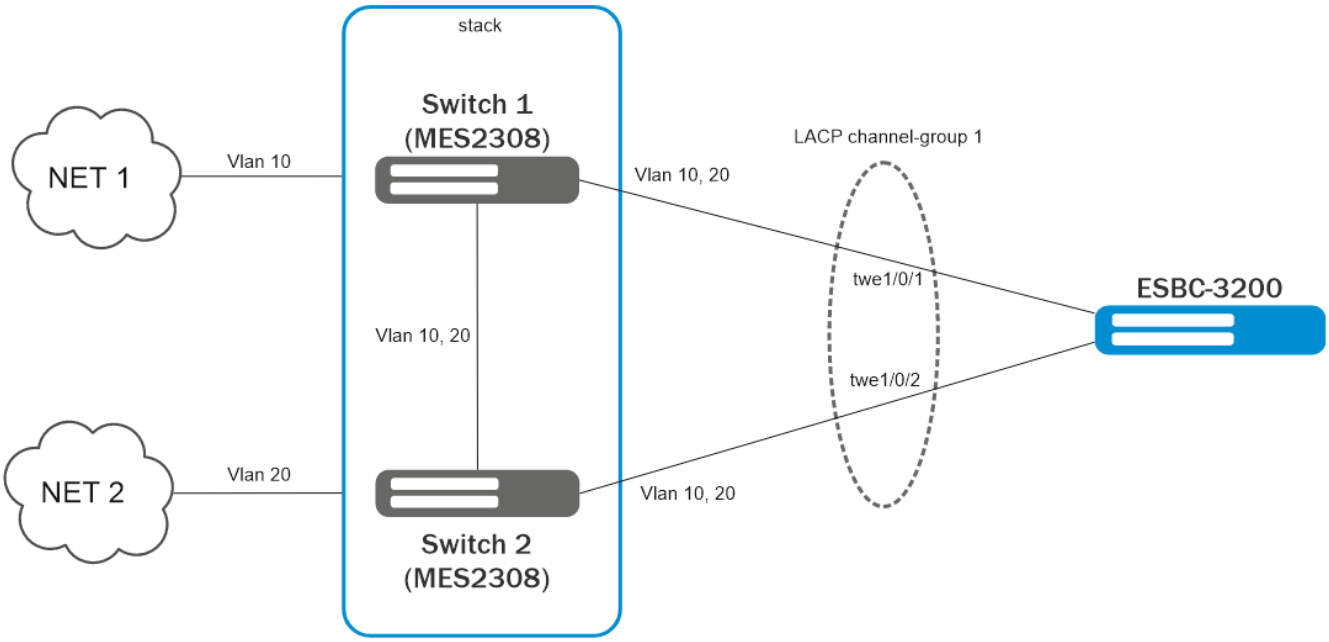
На интерфейсах twe1/0/1 и twe1/0/2 следует использовать режим *switchport* и добавить VLAN 10 и 20:

```
esbc# configure
esbc(config)# interface twentyfivegigabitethernet 1/0/1-2
esbc(config-if-twe)#  mode switchport
esbc(config-if-twe)#  switchport mode trunk
esbc(config-if-twe)#  switchport trunk allowed vlan add 10,20
esbc(config-if-twe)#  exit
```

При использовании интерфейсов в режиме **switchport** необходимо дополнительно настроить протокол семейства STP для предотвращения образования петель.

Наиболее предпочтительным является подключение, описанное в примере Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал, т. к. использование моста (Bridge) может увеличить нагрузку на устройство и приводить к образованию петель коммутации.

8.4 Подключение к нескольким коммутаторам с использованием нескольких сетевых интерфейсов (резервирование линков и узлов сети)



8.4.1 Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал

Подключение и настройка осуществляется аналогично примеру Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал.

Обязательным требованием для реализации данного подключения является поддержка коммутаторами (Switch 1 и Switch 2) функции стекирования и/или VPC.

8.4.2 Использование моста (Bridge) для терминации на уровне L3

В случае если коммутаторы не поддерживают функции стекирования и VPC, то допускается подключение, описанное в примере Использование моста (Bridge) для терминации на уровне L3, но требуется дополнительная настройка протокола STP таким образом, чтобы в топологии STP был заблокирован один из интерфейсов ESBC с целью исключения прохождения транзитного broadcast-трафика.

В данном примере при обрыве линка между Switch 1 и Switch 2, транзитный трафик в любом случае будет проходить через ESBC, что может повлиять на производительность.

> Наиболее предпочтительным является подключение, описанное в примере Использование протокола LACP для агрегирования сетевых интерфейсов в один логический канал, т. к. использование моста (Bridge) может увеличить нагрузку на устройство и приводить к образованию петель коммутации.

8.5 Использование кластера

Пример конфигурации кластера приведен в разделе Управление кластеризацией данного руководства по эксплуатации.

9 Управление ESBC

- Общие сведения
- Настройка абонентских интерфейсов
  - Локальная обработка регистрации
    - Минимальное время регистрации (Expires) на сервере
    - Максимальное время регистрации (Expires) абонента
    - Одновременное использование минимального времени регистрации на сервере и максимального времени регистрации абонента
  - Сброс регистрации абонента на регистраторе
  - Инициация перерегистрации абонента через SIP-NOTIFY
- Настройка SIP-транков
  - Динамический режим транка
    - DNS
    - RADIUS
- Настройка транковых групп
  - Логика работы транковой группы для распределения вызовов на транки, входящие в ее состав
- Настройка SIP-транспортов
- Настройка медиаресурсов
- Использование тегов для маршрутизации и модификации SIP-сообщений
- Настройка таблиц маршрутизации
  - Балансировка правил таблицы маршрутизации
    - Пример балансировки правил по алгоритму round robin
    - Пример балансировки правил по алгоритму weighted round robin
  - Код отбоя для правила Reject
  - Смена таблиц маршрутизации
  - Настройка "закрепления" регистрации
  - Использование условий в таблице маршрутизации
- Настройка модификаторов
  - Общие модификаторы
    - Модификатор замены номера (replace)
    - Модификатор ротации номера (number-rotation)
    - Модификатор добавления тега (add-tag)
    - Модификатор удаления тега (remove-tag)
    - Использование условий в общих модификаторах
  - Модификаторы SIP
    - Модификатор добавления заголовка (add)
    - Модификатор передачи заголовка (transit)
    - Модификатор удаления заголовка (no-transit)
    - Модификатор транзита и замены заголовка (replace)
    - Модификатор копирования (copy)
    - Использование системных переменных
    - Использование условий в SIP-модификаторах
- Настройка SIP-профилей
  - Контроль доступности направления
  - Список причин отбоя для перехода на следующее направление
  - Поведение при перенаправлении
  - Игнорирование OPTIONS
  - Таймеры SIP-сессий (RFC 4028)
  - Транзит сообщений ISUP для работы в режиме SIP-T/SIP-I
  - Опционирование обработки заголовка Contact в запросах и ответах SIP
- Настройка медиапрофилей
  - Управление типом медиаданных и кодеками

        - Примеры использования медиапрофиля для управления кодеками и типами медиаданных в режиме проксирования
    - Транскодирование
        - Примеры использования медиапрофилей для управления кодеками в режиме транскодирования
    - Таймаут ожидания RTP-пакетов
        - Таймаут ожидания RTP-пакетов после получения Comfort Noise
        - Таймаут ожидания RTCP-пакетов
    - Локальная обработка RTCP
    - SRTP
    - Контроль источника RTP
    - Поддержка RFC5168 (PFU)
    - Настройка Media Offroad
- Управление безопасностью системы
    - Настройка профилей безопасности
    - Общий принцип работы модуля fail2ban
    - Фильтрация SIP-флуда
        - Фильтрация клиентских приложений
    - Блокировка по AOR/User-Agent
    - Объединение ошибок по IP-адресу
    - Защита от SIP-spoofing атак
    - Настройка временных периодов
- Настройка AAA
    - Настройка аутентификации абонентов через RADIUS
    - Настройка локальной аутентификации запросов
    - Настройка клиентской регистрации транка
- Настройка NAT
    - Настройка NAT comedia-mode
    - Настройка Public IP
    - Настройка STUN
        - Настройка внешнего STUN-сервера
        - Настройка локального STUN-сервера
- Настройка QoS
- Контроль трафика
    - Контроль входящего трафика
    - Контроль исходящего трафика
- Мониторинг
    - Активные вызовы
    - Расширенный мониторинг вызовов (call-flow)
    - Черный список
    - Белый список
    - Состояние транков
    - Список зарегистрированных абонентов
    - Статистика SIP
    - Мониторинг параметров трафика (KPI)
    - Утилита Sngrep
- Аварии
    - Отправка аварийных SNMP-трапов
- Настройка CDR
- Работа с логами
- Изменение количества модулей
- Настройка VPN (PPTP и L2TP over IPSec)
    - Пример настройки PPTP-сервера для подключения SIP-транков
    - Пример настройки PPTP-клиента для подключения SIP-транков

- Примеры настройки ESBC
  - Настройка для SIP-абонентов
  - Настройка для SIP-транков
  - Настройка для SIP-абонентов, использующих WebRTC
  - Пример настройки балансировки сессий для резервирования направлений

9.1 Общие сведения

В данном разделе содержится описание функций пограничного контроллера сессий ESBC и примеры их настройки для обеспечения передачи SIP-сигнализации и медиапотоков RTP между разными направлениями.

Переход в режим конфигурирования функционала ESBC осуществляется следующими командами CLI:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)#
```

Максимальное количество объектов конфигурации ESBC каждого типа:

Объект	Количество
sip transport	500
trunk	500
user-interface	500
trunk-group	250
sip profile	500
route table	500
rule	32000 на всю систему
condition(route-table)	8 на правило rule
media profile	1000
media resource	1000
mod-table	500
mod	32000 на всю систему
condition(mod-table)	8 на модификатор mod
prefix-group	2000
prefix(number)	250000 на всю систему

Объект	Количество
cause-list	64
crypto profile	64
flood filter	250
security-profile	500
aaa profile	64
radius profile	64
radius server	8 на radius profile
credential profile	64
number	2500 на credential profile
stun server external	64
stun server local	64

Не рекомендуется использовать максимальное количество объектов конфигурации одновременно, это может повлиять на работоспособность системы.

Расчет максимального количества контактов зарегистрированных абонентов для vESBC:

Объем оперативной памяти (RAM) vESBC, GB	Количество контактов
3	4500
≥4	20000*(объем RAM - 3)

ESBC-3200 поддерживает до 420000 зарегистрированных абонентов.

9.2 Настройка абонентских интерфейсов

Абонентский интерфейс представляет собой направление для приёма и маршрутизации запросов SIP-абонентов. В конфигурации не задаётся адрес и порт удалённой стороны, для аутентификации используется механизм SIP-регистрации. Регистрация на вышестоящем сервере осуществляется через связанный SIP-транк.

Для создания абонентского интерфейса необходимо настроить:

- SIP-транспорт;
- Медиаресурсы;
- Таблицу маршрутизации.

Эти настройки являются обязательными. Описание конфигурирования и базовой схемы применения представлено в разделе Примеры настройки ESBC.

Помимо этого абонентский интерфейс содержит набор следующих настроек:

- Таблица модификаций (для входящих и исходящих сообщений);
- SIP-профиль;
- Медиапрофиль;
- Профиль безопасности;
- Режим работы абонентов за NAT;
- Public IP;
- QoS;
- Контроль входящего и исходящего трафика;
- Локальная обработка регистрации;
- AAA профиль;
- STUN-сервер;
- SIP-домен. При настройке домен будет использоваться в host-part URI в заголовках To и From для исходящих сообщений. Во входящих сообщениях будет осуществляться проверка домена в заголовке From;
- Опция "Разрешить вызовы без регистрации". Разрешает принимать входящие сообщения INVITE от незарегистрированных абонентов;
- Сброс регистрации абонента на регистраторе;
- Инициация перерегистрации абонента через SIP-NOTIFY.

По умолчанию вызовы с абонентского интерфейса без предварительной регистрации запрещены.

С целью повышения безопасности, входящие запросы OPTIONS, поступающие в абонентский интерфейс, будут игнорироваться. Для изменения поведения см. раздел Игнорирование OPTIONS.

Подробное описание параметров всех настроек можно найти в разделе Настройки абонентского интерфейса справочника команд CLI.

9.2.1 Локальная обработка регистрации

Использование локальной обработки регистрации позволяет снизить нагрузку на сервер регистрации за счет локального ответа на повторный запрос регистрации.

Данная функция позволяет управлять параметром заголовка Expires при регистрации абонента через ESBC, что позволяет снизить нагрузку на сервер регистрации и дает возможность контролировать доступность зарегистрированного абонента.

По умолчанию локальная обработка регистрации на абонентском интерфейсе отключена.

Подробное описание команд для настройки локальной обработки регистраций на ESBC представлено в настройках user-interface в CLI.

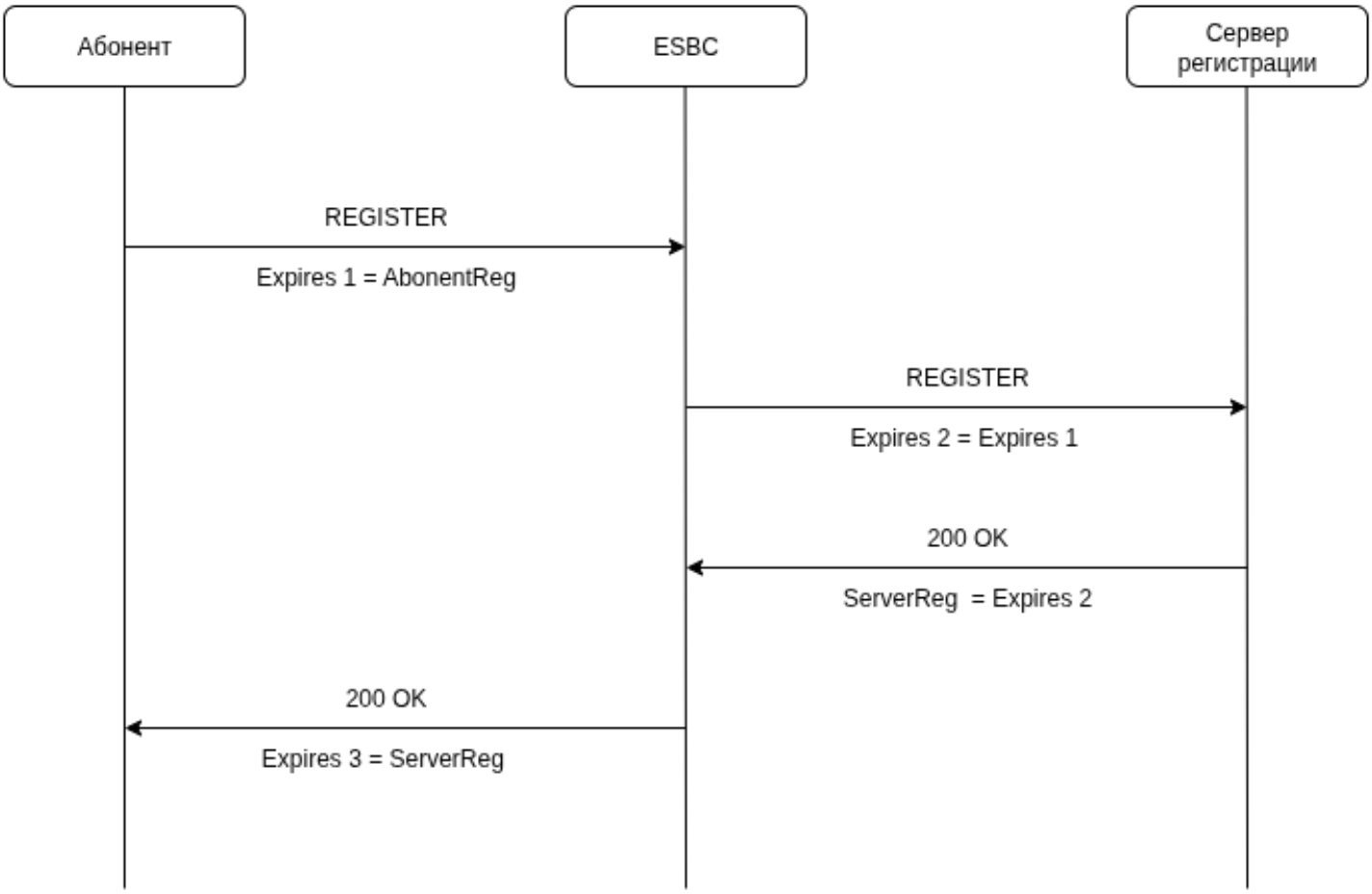
При записи информации о зарегистрированных абонентах на ESBC всегда используется Expires, пришедший с сервера регистрации.

Локальная обработка повторной регистрации разрешена, если соблюдаются все нижеперечисленные требования:

1. Абонент уже зарегистрирован;
2. Контакты абонента не изменились;
3. Адрес и порт абонента не изменились;
4. Абонент не регистрирует новые контакты.

Логика первичной регистрации:

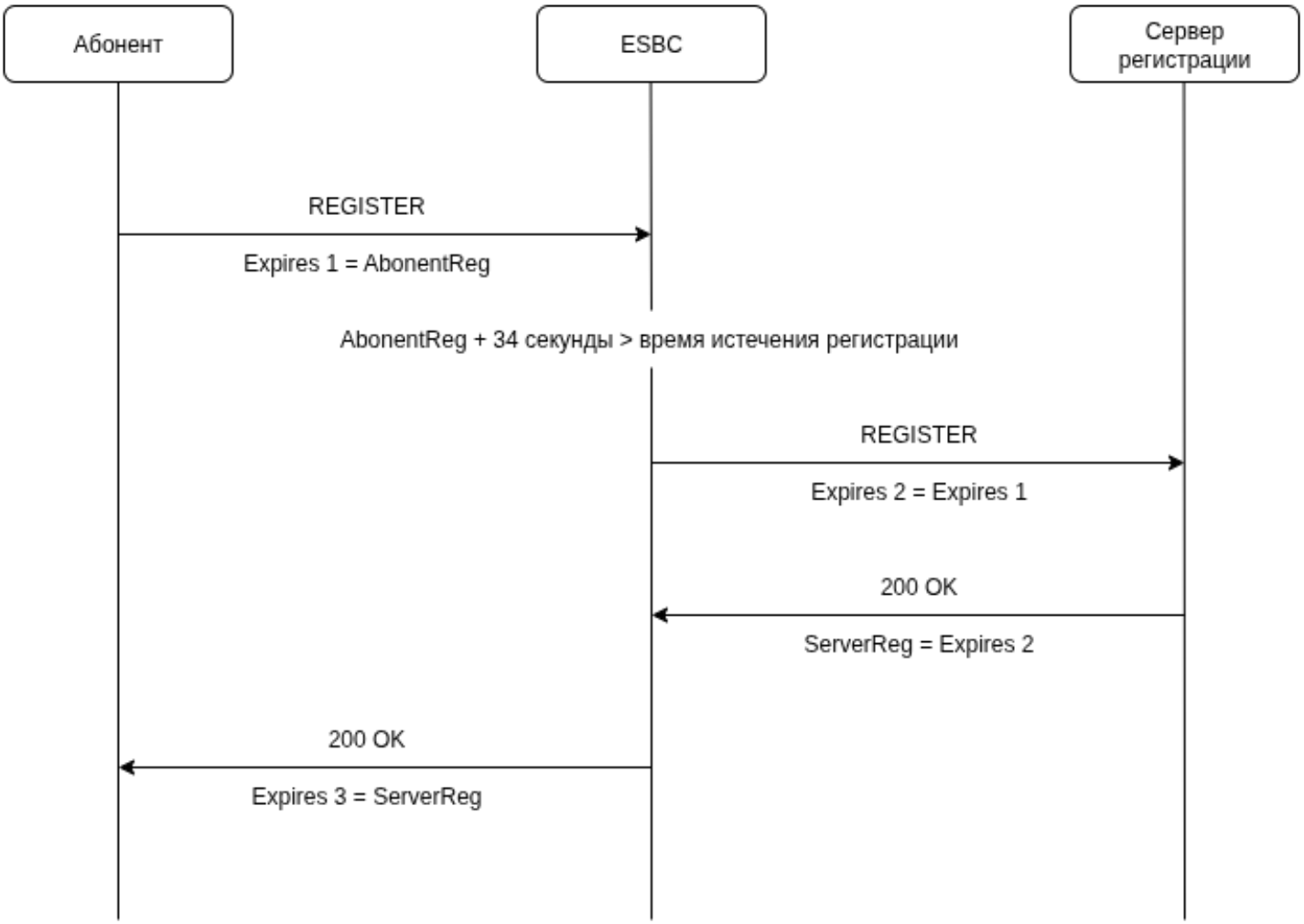
1. В сторону сервера регистрации отправляем REGISTER с Expires, пришедшим от абонента;
2. В сторону абонента отправляем 200 OK с Expires из ответа от сервера регистраций.



AbonentReg - Expires пришедший от абонента
ServerReg - Expires пришедший от сервера регистрации

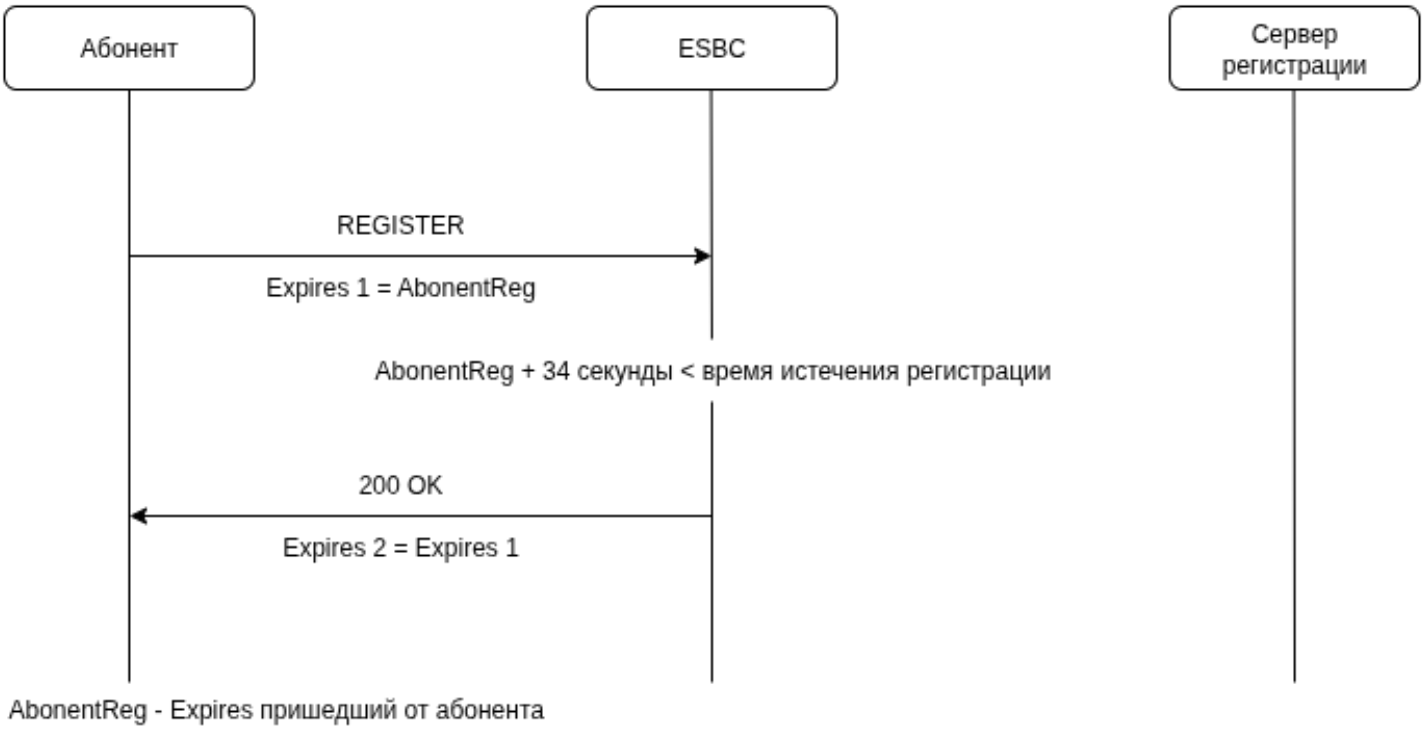
Логика работы локальной обработки повторной регистрации для зарегистрированного абонента:

1. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды больше, чем оставшееся время регистрации (Expires) на ESBC, то отправляем REGISTER на сервер регистрации со значением Expires, описанным пунктом выше.



AbonentReg - Expires пришедший от абонента
ServerReg - Expires пришедший от сервера регистрации

2. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды меньше, чем оставшееся время регистрации (Expires) на ESBC, то регистрация будет обработана локально путем отправки абоненту 200 OK с Expires, пришедшим от него же самого.

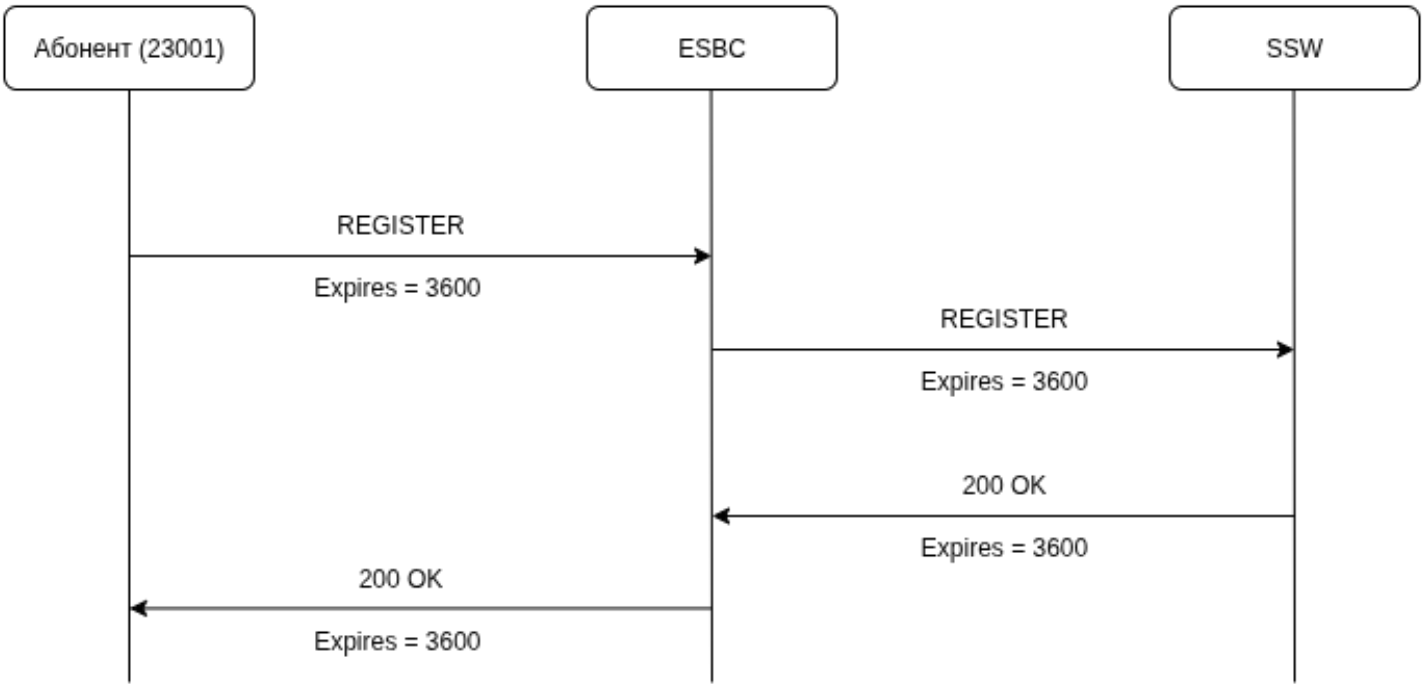


При разрегистрации абонента логика локальной регистрации не используется.

Пример настройки локальной обработки регистраций в конфигурации абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение локальной обработки регистраций:
vesbc(config-esbc-user-interface-sip)# registration simulation enable

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

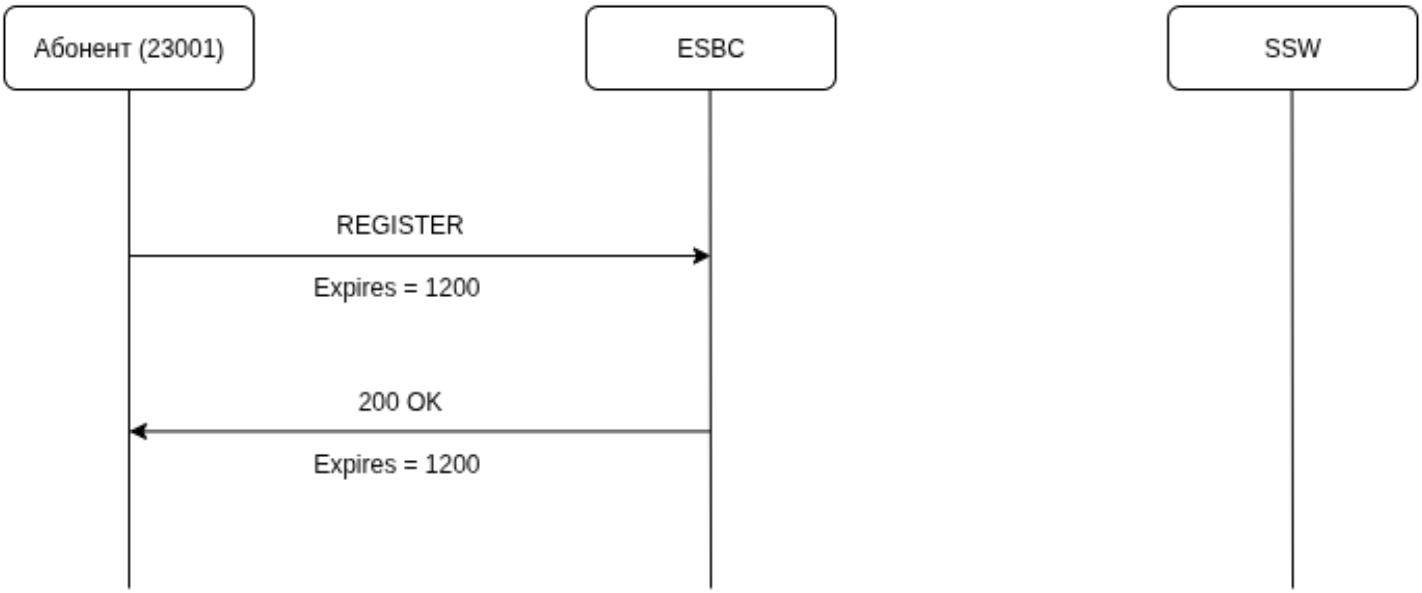
На абонентский интерфейс ESBC приходит запрос REGISTER от абонента 23001 с Expires 3600, который пересылается на сервер регистрации SSW. После получения 200 OK от сервера регистрации, ESBC отправляет 200 OK абоненту, содержащий Expires 3600:



На ESBC создается запись зарегистрированного абонента с Expires 3600:

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User        Trunk name   IP address of     OUT
Trunk contact          Status        Expires
                            user              interface                registrar
                                              name
-------------------------   ---------------   ----------  -----------  ---------------
-------------------------  ------------  ---------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT     SSW          192.168.126.212   <sip:2
3001@192.168.126.21  Uncontrolled  3554/3600
0:12300;transport=UDP>                                                                    7:5071;
line=9e538107d8068

2246c8eed958f91da89>
```

Спустя время от абонента приходит запрос REGISTER с Expires 1200 для перерегистрации. Так как оставшееся время регистрации (1523) больше, чем пришедший Expires (1200), то запрос будет обработан локально с отправкой 200 OK абоненту, содержащий Expires 1200:



На ESBC запись зарегистрированного абонента не изменяется:

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact            IP address of     User        Trunk name  IP address of    OUT Trunk
contact          Status        Expires
                           user              interface               registrar
                                             name
-------------------------  ---------------   ----------  ----------  ---------------
-------------------------  ------------  ---------
<sip:23001@192.168.126.21  192.168.126.210   ABONENT     SSW         192.168.126.212  <sip:2300
1@192.168.126.21  Uncontrolled  1523/3600
0:12300;transport=UDP>                                                                 7:5071;li
ne=9e538107d8068

2246c8eed958f91da89>
```

Минимальное время регистрации (Expires) на сервере

Функция настройки параметра заголовка Expires, передаваемого в сообщении REGISTER в сторону регистратора, позволяет снизить нагрузку на сервер, путем уменьшения количества запросов REGISTER при большом количестве зарегистрированных абонентов.

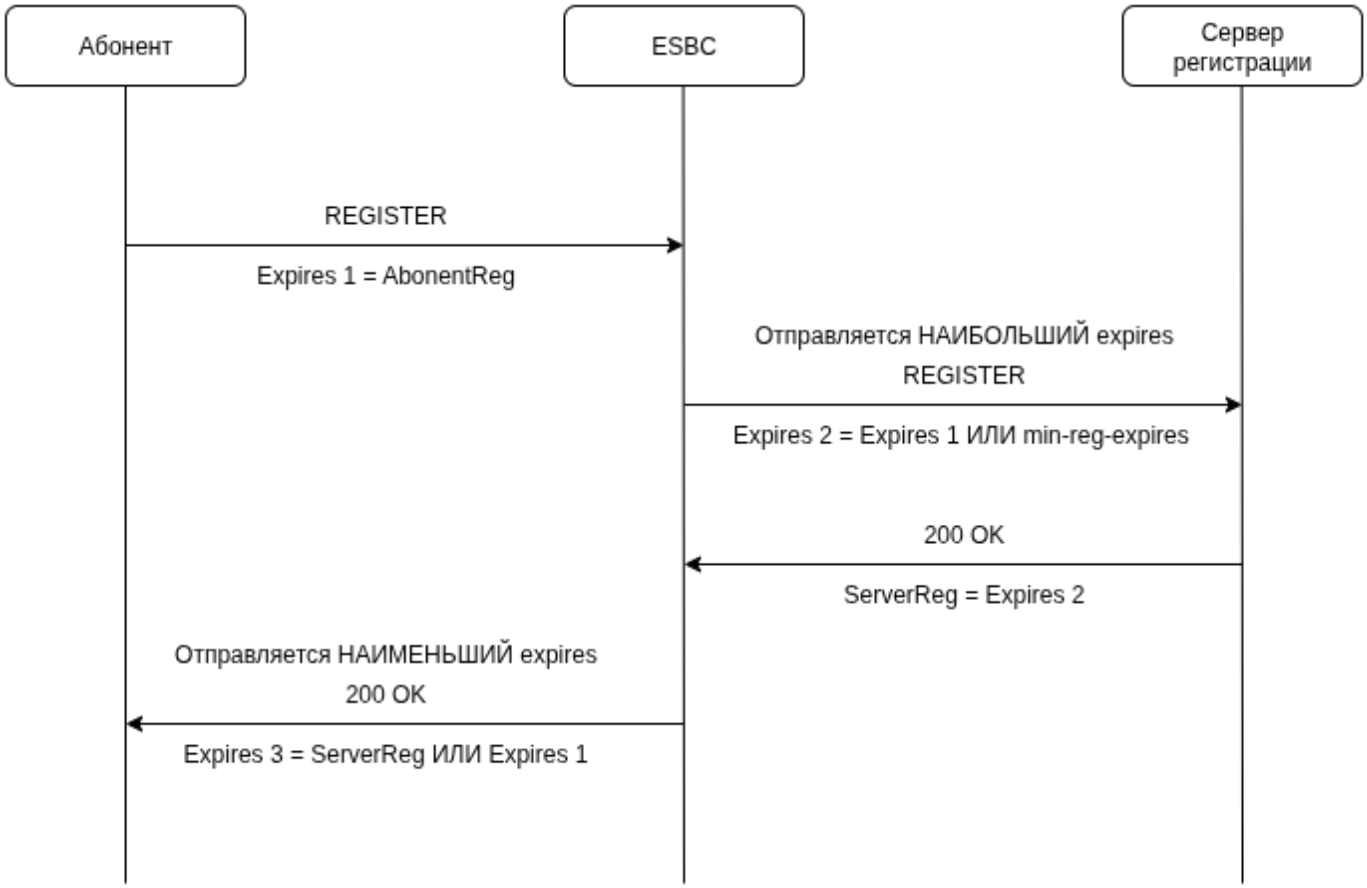
> ⚠ Работает только при включенной локальной обработке регистраций.

Подробное описание команд для настройки минимального времени регистрации на сервере при включенной локальной обработкой регистраций на ESBC представлено в настройках user-interface в CLI.

Может работать вместе с максимальным временем регистрации абонента. Подробнее в разделе ниже.

Логика работы первичной регистрации при использовании минимального времени регистрации на сервере:

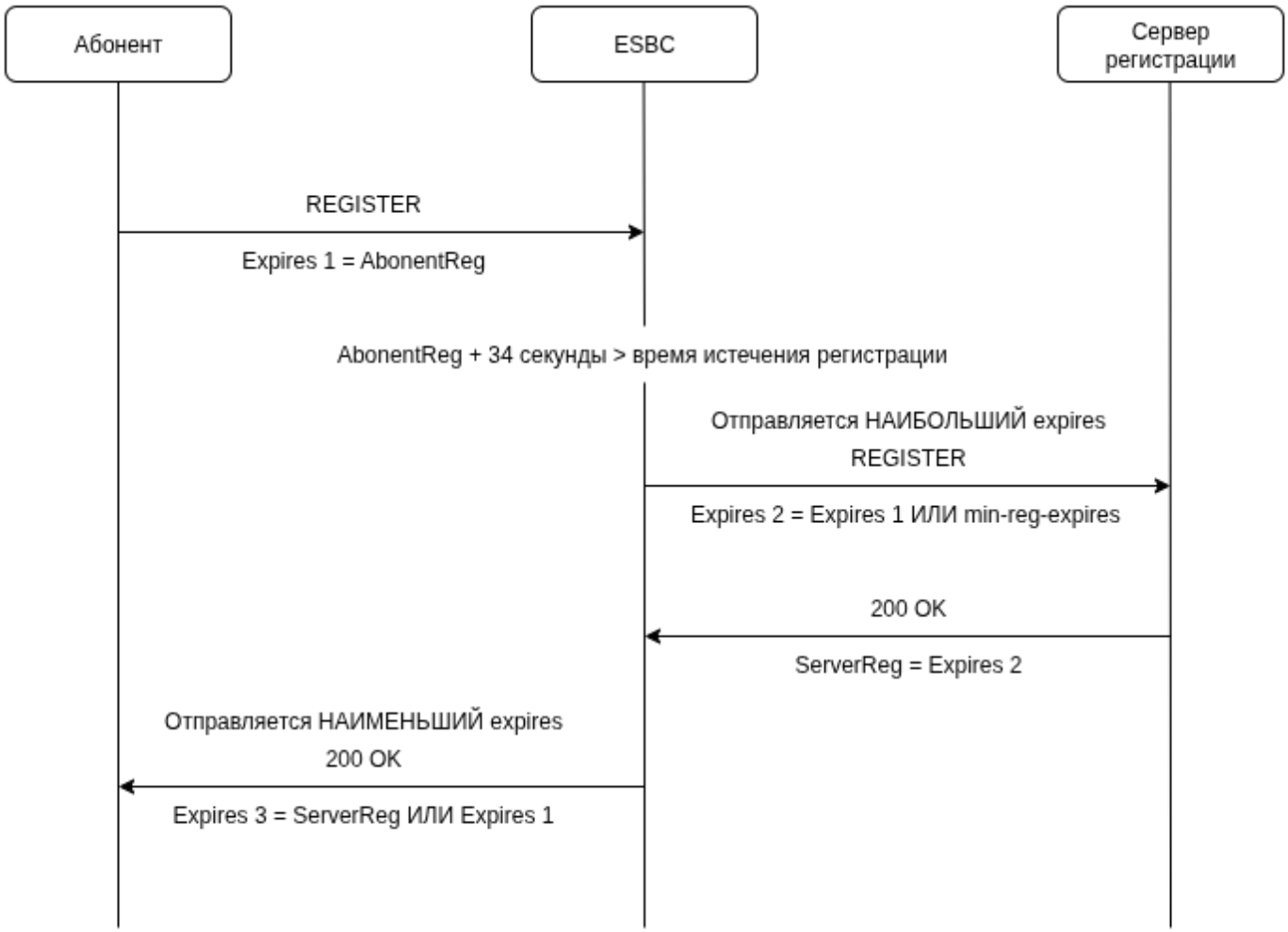
1. В сторону сервера регистрации отправляем REGISTER с наибольшим Expires, пришедшим от абонента, и минимальным временем регистрации на сервере;
2. В сторону абонента отправляем 200 OK с наименьшим Expires из ответа от сервера регистраций и пришедшего от абонента.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации

Логика работы локальной обработки повторной регистрации при использовании минимального времени регистрации на сервере для зарегистрированного абонента:

1. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды больше, чем оставшееся время регистрации (Expires) на ESBC, то отправляем REGISTER на сервер регистрации со значением Expires, описанным пунктом выше.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации

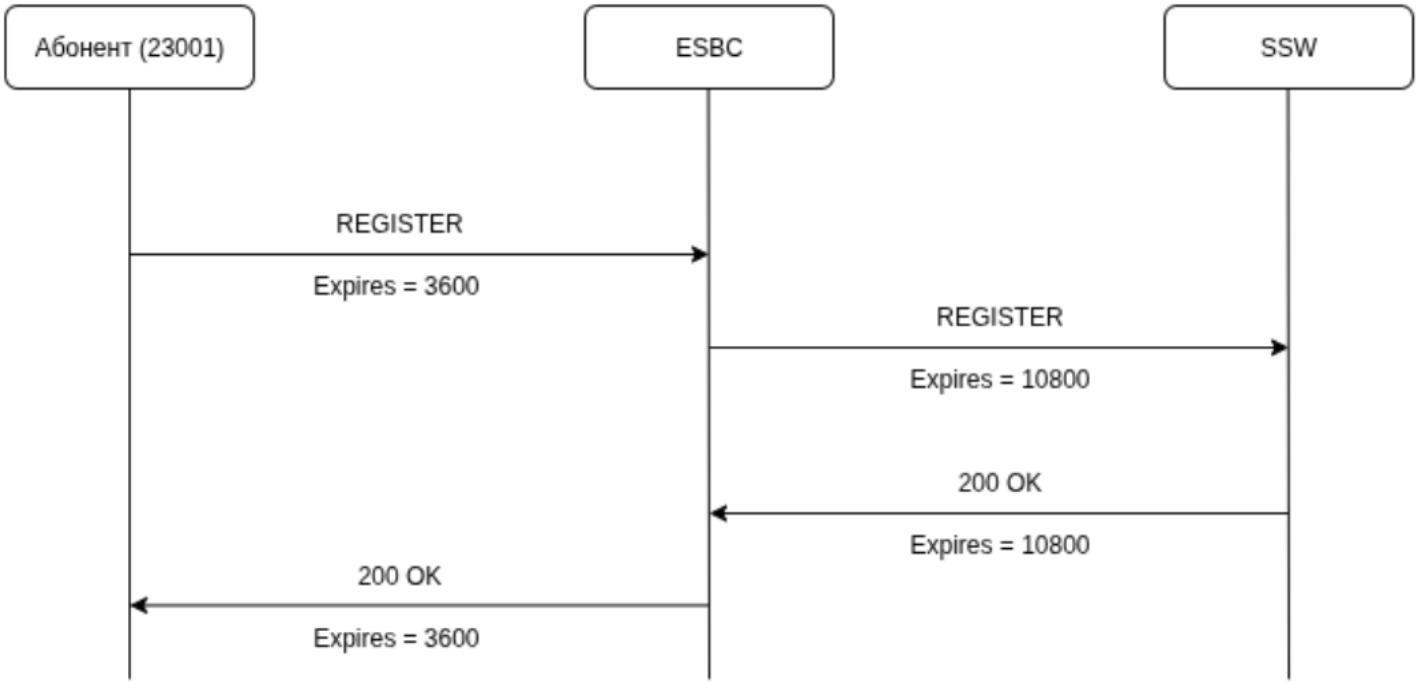
2. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды меньше, чем оставшееся время регистрации (Expires) на ESBC, то регистрация будет обработана локально путем отправки абоненту 200 OK с Expires, пришедшим от него же самого.

⚠ Если абонент не перерегистрируется согласно тому Expires, который он получил при регистрации от ESBC, то такой абонент будет считаться недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.
Абонент станет доступен только после перерегистрации.

Пример настройки локальной обработки регистраций с использованием минимального времени регистрации на сервере в конфигурации абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение локальной обработки регистраций:
vesbc(config-esbc-user-interface-sip)# registration simulation enable

#Использование минимального времени регистрации на сервере:
vesbc(config-esbc-user-interface-sip)# registration simulation min-reg-expires 10800

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

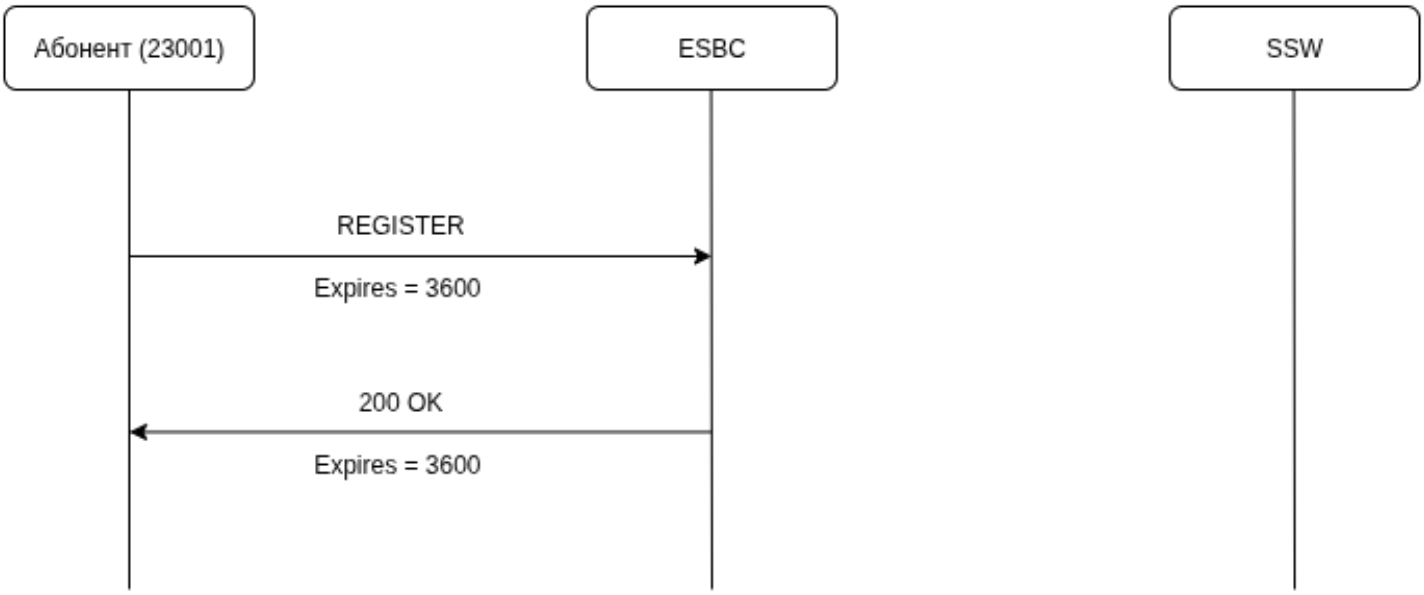
На абонентский интерфейс ESBC приходит запрос REGISTER от абонента 23001 с Expires 3600, который пересылается на сервер регистрации SSW со значением Expires 10800. После получения 200 OK от сервера регистрации со значением Expires 10800, ESBC отправляет 200 OK абоненту, содержащий Expires 3600:



На ESBC создается запись зарегистрированного абонента с Expires 10800:

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User        Trunk name   IP address of    OUT
Trunk contact          Status     Expires
                            user              interface                registrar
                                              name
-------------------------   ---------------   ---------   ----------   ---------------
-------------------------  ---------  -----------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT     SSW          192.168.126.212  <sip:23
001@192.168.126.21  Available  10693/10800
0:12300;transport=UDP>                                                                   7:5071;
line=9e538107d8068

2246c8eed958f91da89>
```

Спустя время приходит запрос REGISTER с Expires 3600 для перерегистрации. Так как оставшееся время регистрации (Expires) (8978) больше, чем пришедший Expires (3600), то запрос будет обработан локально с отправкой 200 OK абоненту, содержащий Expires 3600:




```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User         Trunk name   IP address of    OUT
Trunk contact          Status     Expires
                            user              interface                  registrar
                                              name
-------------------------   ---------------   ----------   ----------   ---------------
-------------------------  ---------  -----------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT      SSW          192.168.126.212  <sip:2
3001@192.168.126.21  Available  8978/10800
0:12300;transport=UDP>                                                                   7:5071;
line=9e538107d8068

2246c8eed958f91da89>
```

Если спустя время запрос REGISTER для перерегистрации не поступил, то абонент становится недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User        Trunk name    IP address of   OUT
Trunk contact          Status         Expires
                            user              interface                  registrar
                                              name
-------------------------   ---------------   ---------   -----------  ---------------
-------------------------  -------------  ---------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT     SSW          192.168.126.212  <sip:23
001@192.168.126.21  Not available  322/10800
0:12300;transport=UDP>                                                                   7:5071;
line=9e538107d8068

2246c8eed958f91da89>
```

Максимальное время регистрации (Expires) абонента

Функция настройки параметра заголовка Expires, передаваемого в сообщении 200 OK на регистрацию в сторону абонента, позволяет управлять частотой обновления регистрации абонента (без дополнительной нагрузки на сервер регистрации), тем самым позволяет более гибко контролировать доступность абонента.

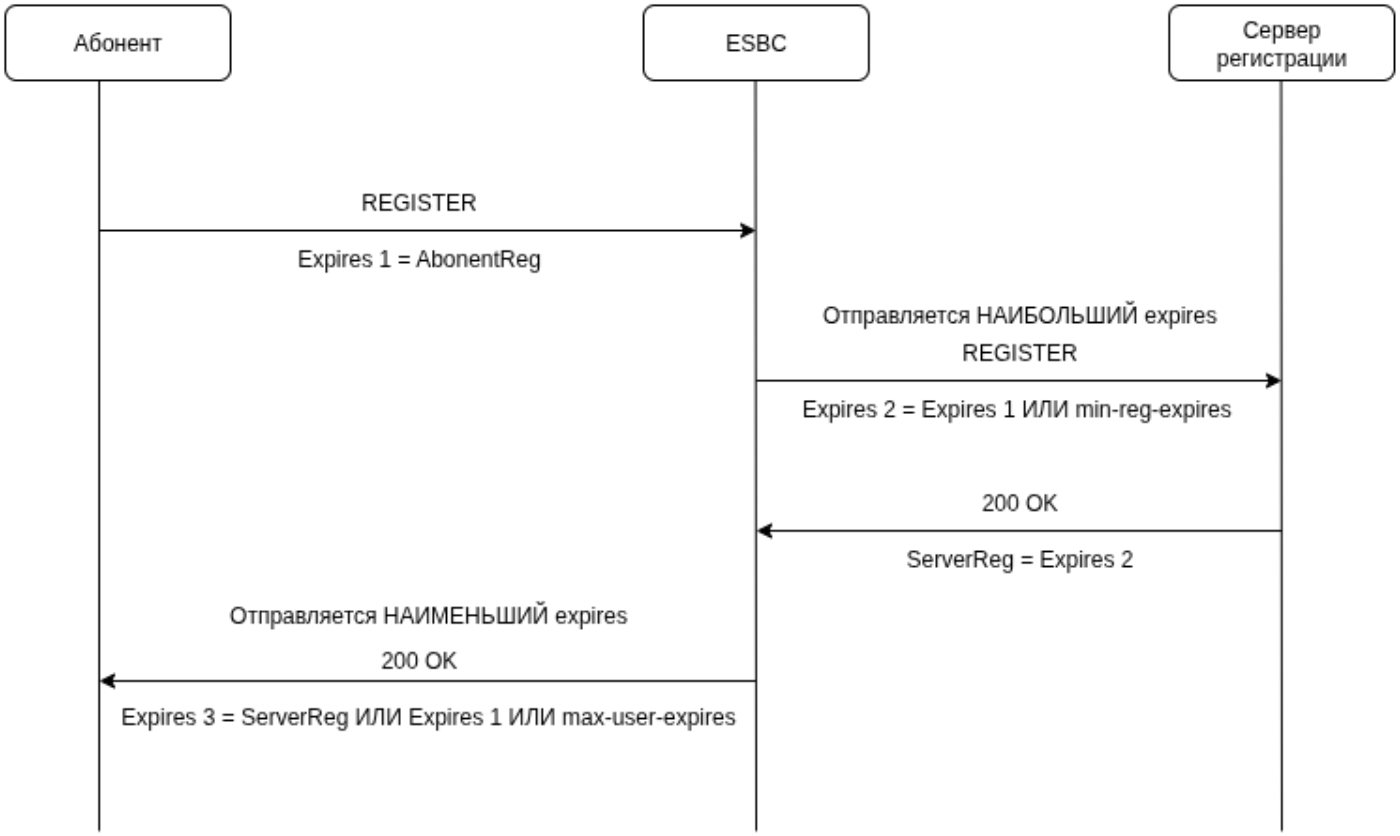
> ⚠ Работает только при включенной локальной обработке регистраций.

Подробное описание команд для настройки максимального времени регистрации абонента при включенной локальной обработке регистраций на ESBC представлено в настройках user-interface в CLI.

Может работать вместе с минимальным временем регистрации на сервере. Подробнее в разделе ниже.

Логика работы первичной регистрации при использовании максимального времени регистрации абонента:

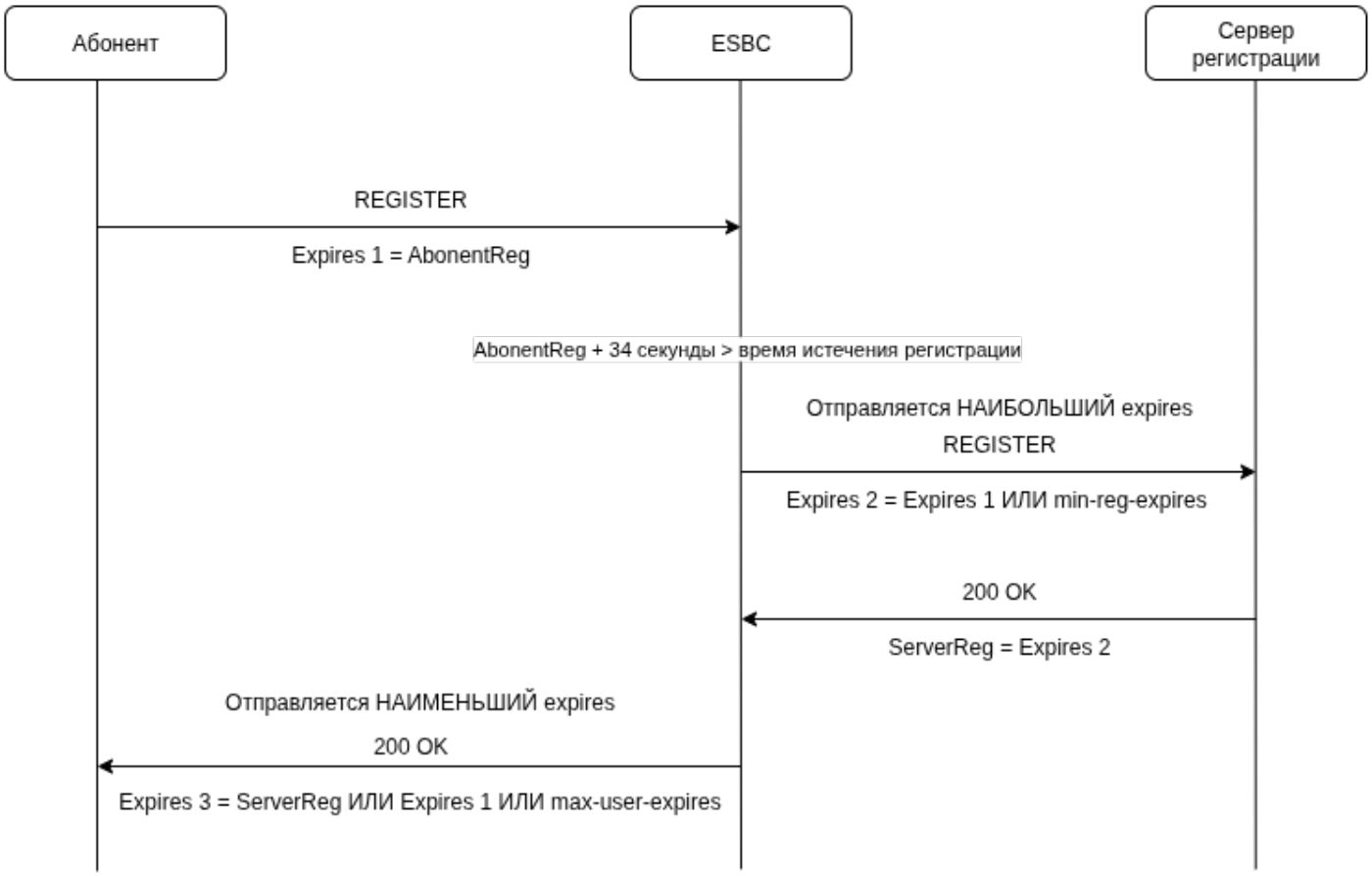
1. В сторону сервера регистрации отправляем REGISTER с Expires, пришедшим от абонента;
2. В сторону абонента отправляем 200 OK с минимальным Expires из ответа от сервера регистраций и настроенным максимальным временем регистрации абонента.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации
max-user-expires - максимальное время регистрации абонента, настроенное в локальной обработке регистрация на ESBC

Логика работы локальной обработки повторной регистрации при использовании максимального времени регистрации абонента для зарегистрированного абонента:

1. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды больше, чем оставшееся время регистрации (Expires) на ESBC, то отправляем REGISTER на сервер регистрации со значением Expires, описанным пунктом выше.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации
max-user-expires - максимальное время регистрации абонента, настроенное в локальной обработке регистрация на ESBC

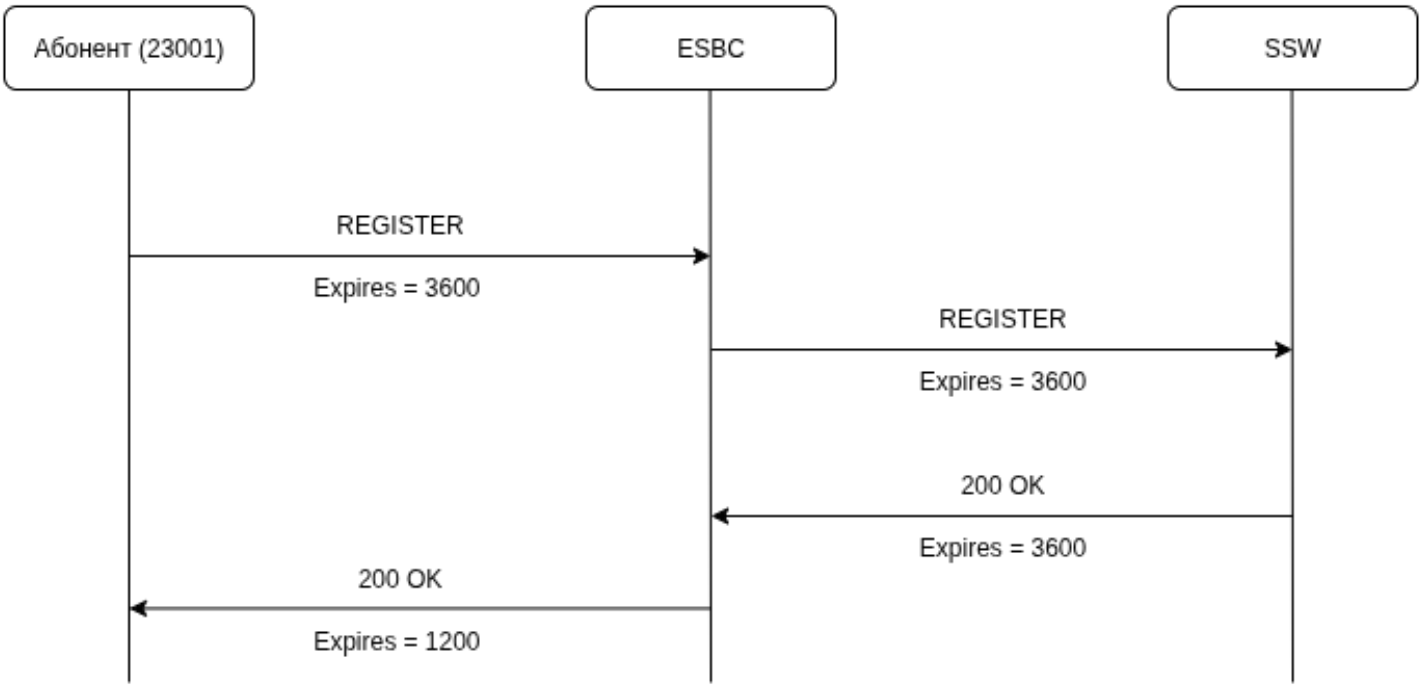
2. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды меньше, чем оставшееся время регистрации (Expires) на ESBC, то регистрация будет обработана локально путем отправки абоненту 200 OK с Expires, пришедшим от него же самого.

⚠ Если абонент не перерегистрируется согласно тому Expires, который он получил при регистрации от ESBC, то такой абонент будет считаться недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.
Абонент станет доступен только после перерегистрации.

Пример настройки локальной обработки регистраций с использованием максимального времени регистрации абонента в конфигурации абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение локальной обработки регистраций:
vesbc(config-esbc-user-interface-sip)# registration simulation enable

#Использование максимального времени регистрации абонента:
vesbc(config-esbc-user-interface-sip)# registration simulation max-user-expires 1200

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

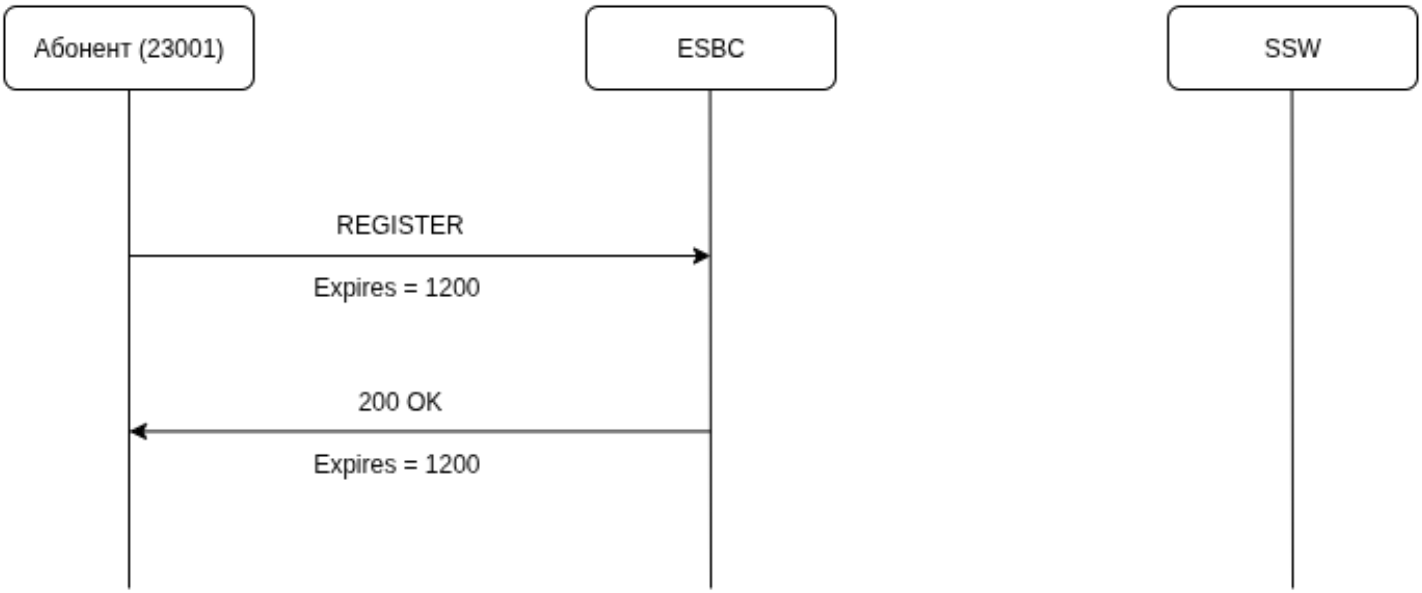
На абонентский интерфейс ESBC приходит запрос REGISTER от абонента 23001 с Expires 3600, который пересылается на сервер регистрации SSW со значением Expires 3600. После получения 200 OK от сервера регистрации со значением Expires 3600, ESBC отправляет 200 OK абоненту, содержащему Expires 1200:



На ESBC создается запись зарегистрированного абонента с Expires 3600:

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact            IP address of     User        Trunk name  IP address of    OUT Trunk
contact          Status     Expires
                           user              interface               registrar
                                             name
------------------------  ---------------   ----------  ----------  ---------------
-------------------------  ---------  ---------
<sip:23001@192.168.126.21  192.168.126.210   ABONENT     SSW         192.168.126.212  <sip:2300
1@192.168.126.21  Available  3550/3600
0:12300;transport=UDP>                                                                 7:5071;li
ne=9e538107d8068

2246c8eed958f91da89>
```

Спустя время приходит запрос REGISTER с Expires 1200 для перерегистрации. Так как оставшееся время регистрации (Expires) больше (3501), чем пришедший Expires (1200), то запрос будет обработан локально с отправкой 200 OK абоненту, содержащий Expires 1200:




```
vesbc#
 vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact            IP address of    User        Trunk name  IP address of    OUT Trunk
contact          Status      Expires
                           user             interface               registrar
                                            name
------------------------  ---------------  ----------  ----------  ---------------
-------------------------  ---------   ---------
<sip:23001@192.168.126.21  192.168.126.210  ABONENT     SSW         192.168.126.212  <sip:23001@
192.168.126.21  Available   2501/3600
0:12300;transport=UDP>                                                              7:5071;lin
e=9e538107d8068

2246c8eed958f91da89>
```

Если спустя время запрос REGISTER для перерегистрации не поступил, то абонент становится недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.

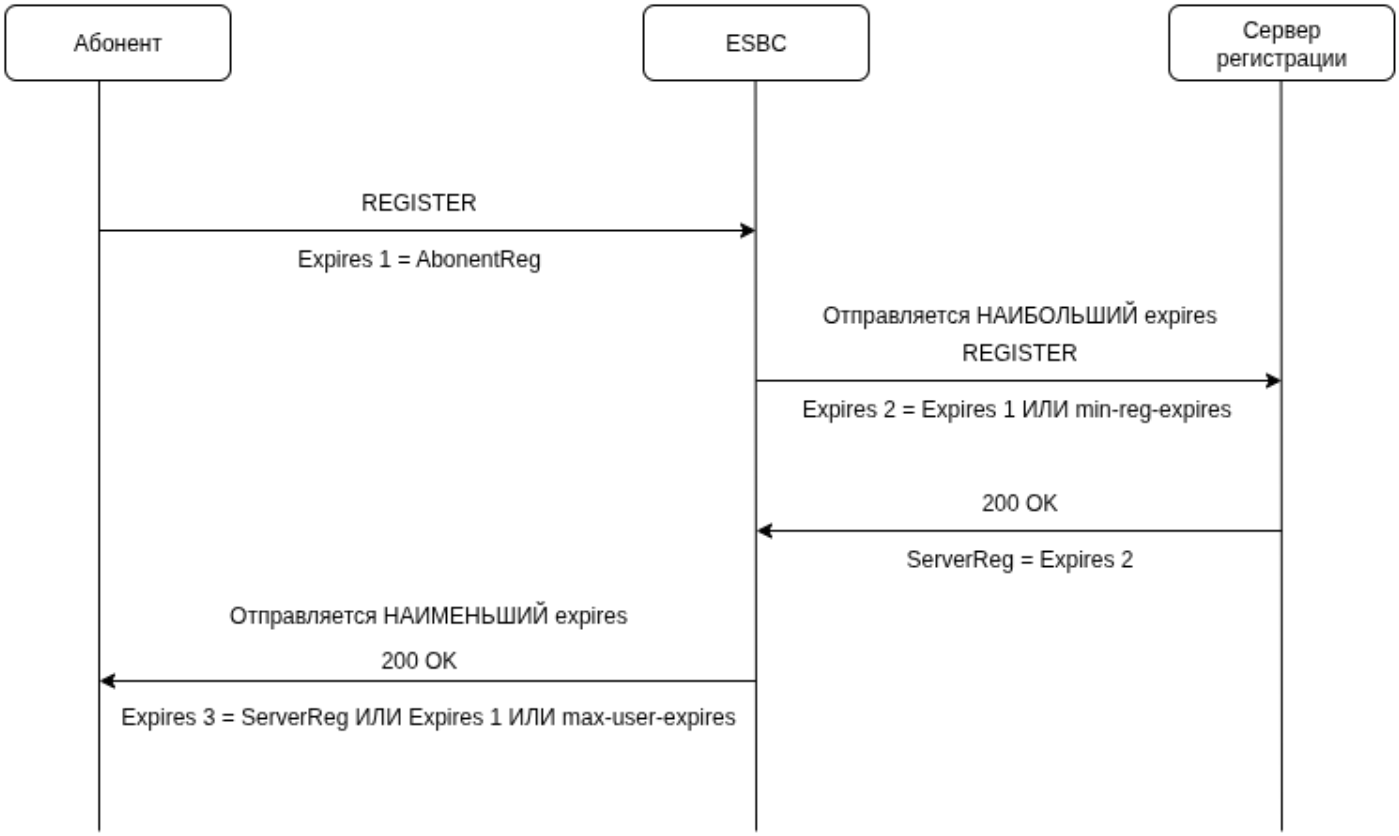
```
vesbc#
 vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT      SSW               192.168.126.212
<sip:23001@192.168.126.21   Not available   1984/3600
0:12300;transport=UDP>
7:5071;line=9e538107d8068

2246c8eed958f91da89>
```

Одновременное использование минимального времени регистрации на сервере и максимального времени регистрации абонента

⚠ Работает только при включенной локальной обработке регистраций.

Логика первичной регистрации при использовании минимального времени регистрации на сервере и максимального времени регистрации абонента:

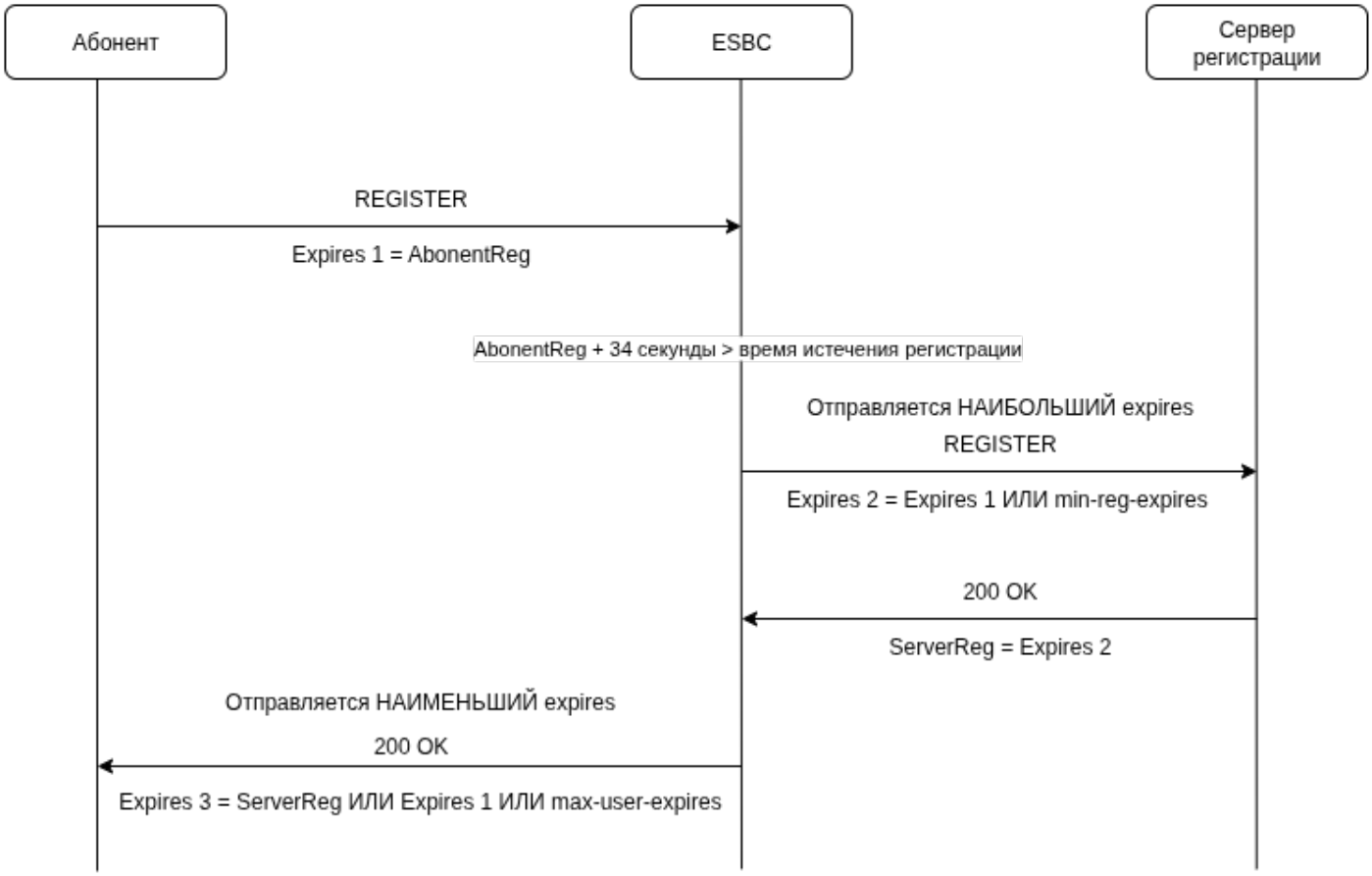
1. В сторону сервера регистрации отправляем REGISTER с наибольшим Expires, пришедшим от абонента, и минимальным временем регистрации на сервере;
2. В сторону абонента отправляем 200 OK с наименьшим Expires из ответа от сервера регистраций, пришедшего от абонента и максимальным временем регистрации абонента.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации
max-user-expires - максимальное время регистрации абонента, настроенное в локальной обработке регистрация на ESBC

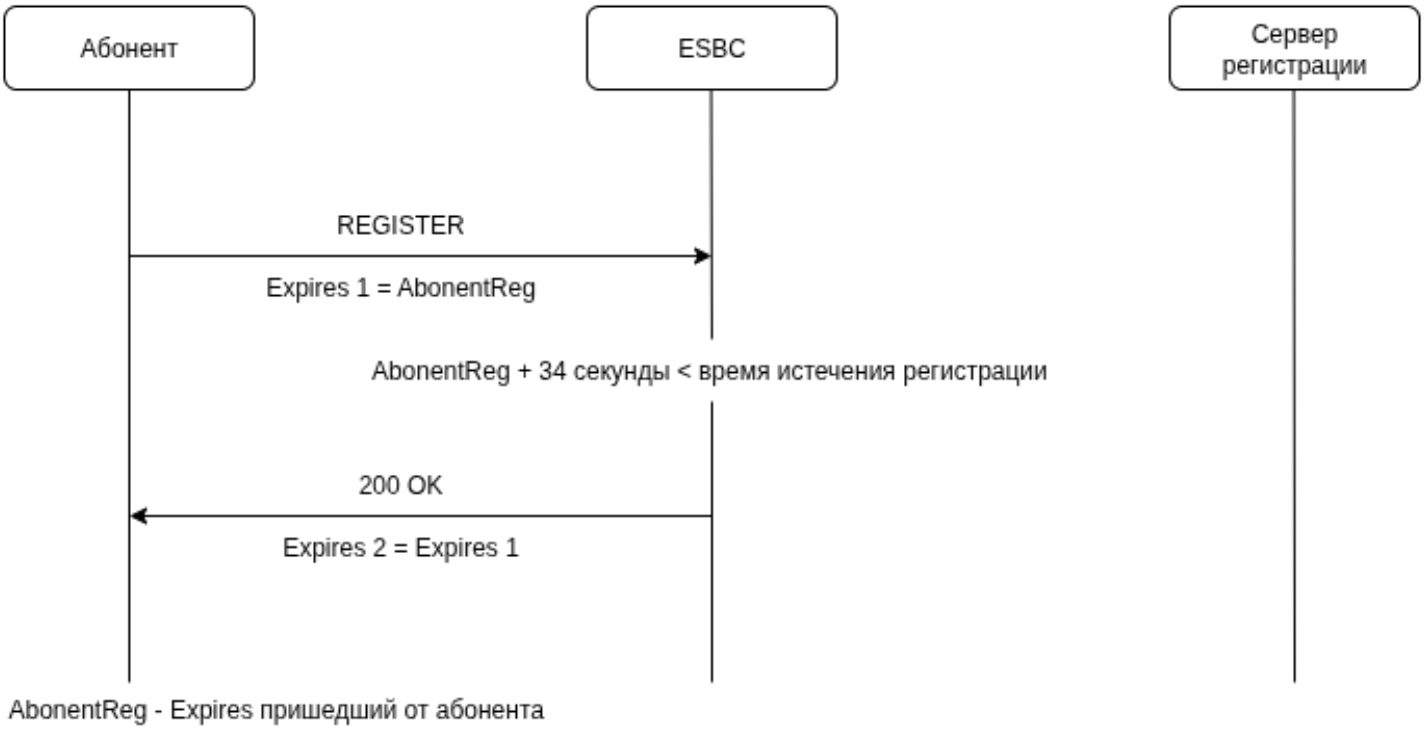
Логика работы локальной обработки повторной регистрации при использовании минимального времени регистрации на сервере и максимального времени регистрации абонента для зарегистрированного абонента:

1. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды больше, чем оставшееся время регистрации (Expires) на ESBC, то отправляем REGISTER на сервер регистрации со значением Expires, описанным пунктом выше.



AbonentReg - Expires пришедший от абонента
min-reg-expires - минимальное время регистрации на сервере, настроенное в локальной обработке регистраций на ESBC
ServerReg - Expires пришедший от сервера регистрации
max-user-expires - максимальное время регистрации абонента, настроенное в локальной обработке регистрация на ESBC

2. Если в REGISTER, полученном от абонента, Expires в поле Contact + 34 секунды меньше, чем оставшееся время регистрации (Expires) на ESBC, то регистрация будет обработана локально путем отправки абоненту 200 OK с Expires, пришедшим от него же самого.

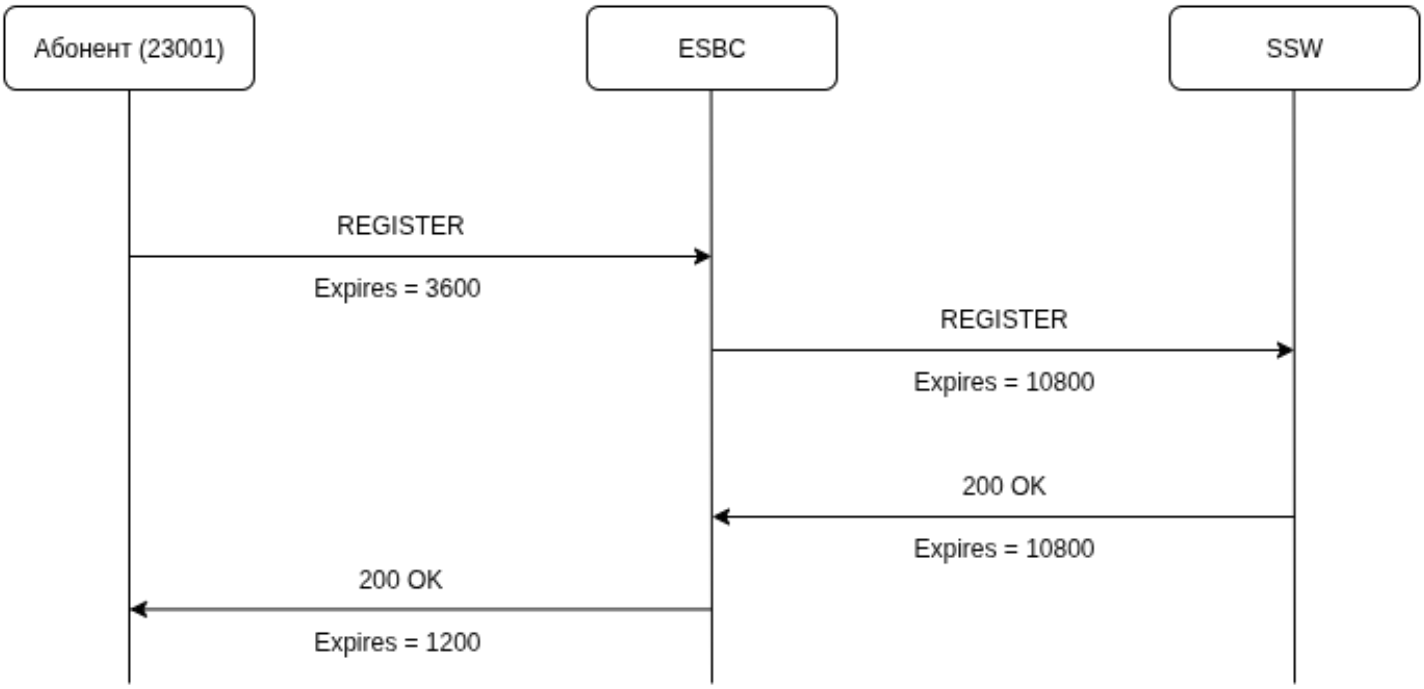


Если абонент не перерегистрируется согласно тому Expires, который он получил при регистрации от ESBC, то такой абонент будет считаться недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.
Абонент станет доступен только после перерегистрации.

Пример настройки локальной обработки регистраций с использованием минимального времени регистрации на сервере и максимального времени регистрации абонента в конфигурации абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение локальной обработки регистраций:
vesbc(config-esbc-user-interface-sip)# registration simulation enable

#Использование минимального времени регистрации на сервере:
vesbc(config-esbc-user-interface-sip)# registration simulation min-reg-expires 10800

#Использование максимального времени регистрации абонента:
vesbc(config-esbc-user-interface-sip)# registration simulation max-user-expires 1200

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

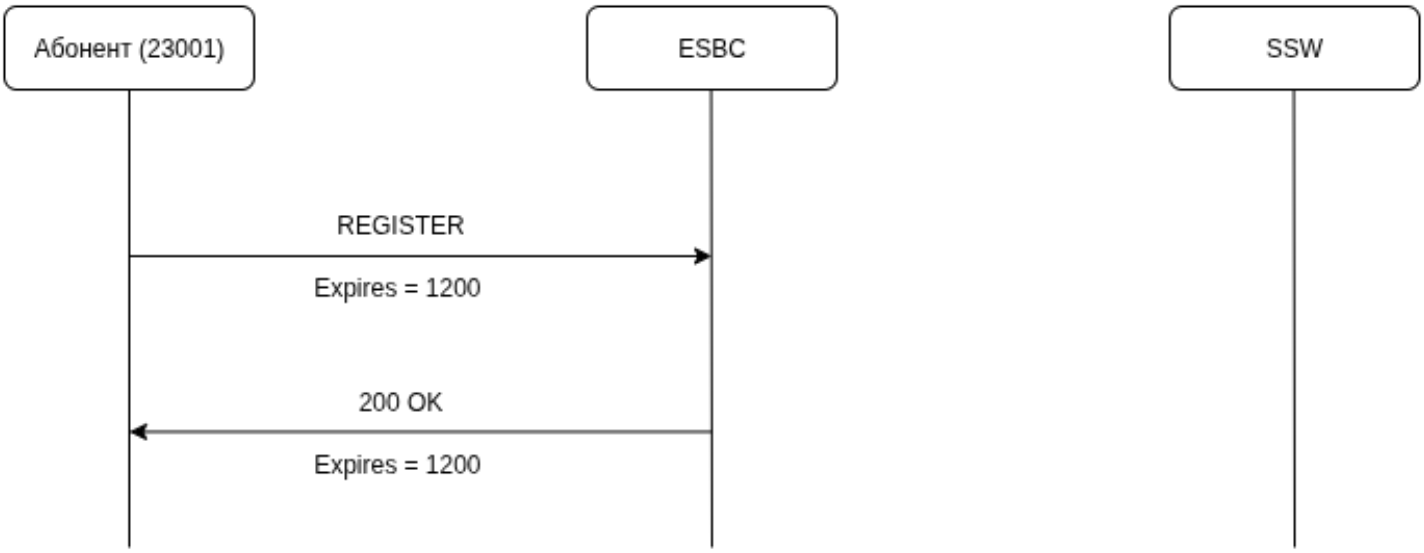
На абонентский интерфейс ESBC приходит запрос REGISTER от абонента 23001 с Expires 3600, который пересылается на сервер регистрации SSW со значением Expires 10800. После получения 200 OK от сервера регистрации со значением Expires 10800, ESBC отправляет 200 OK абоненту, содержащий Expires 1200:



На ESBC создается запись зарегистрированного абонента с Expires 10800:

```
vesbc#
vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT      SSW               192.168.126.212
<sip:23001@192.168.126.21   Available       10775/10800
0:12300;transport=UDP>
7:5071;line=9e538107d8068

2246c8eed958f91da89>
```

Спустя время приходит запрос REGISTER с Expires 1200 для перерегистрации. Так как оставшееся время регистрации (Expires) (9402) больше, чем пришедший Expires (1200), то запрос будет обработан локально с отправкой 200 OK абоненту, содержащему Expires 1200:



```
vesbc#
 vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT      SSW               192.168.126.212
<sip:23001@192.168.126.21   Available       9402/10800
0:12300;transport=UDP>
7:5071;line=9e538107d8068

2246c8eed958f91da89>
```

Если спустя время запрос REGISTER для перерегистрации не поступил, то абонент становится недоступным. Все вызовы на этого абонента будут завершаться с причиной 480 Temporarily Unavailable.

```
vesbc#
 vesbc# show esbc users sip 23001@192.168.126.217 detailed
 User AOR:       23001@192.168.126.217
 User type:      SIP
 Contact count:  1

IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:23001@192.168.126.21   192.168.126.210   ABONENT      SSW               192.168.126.212
<sip:23001@192.168.126.21   Not available   988/10800
0:12300;transport=UDP>
7:5071;line=9e538107d8068

2246c8eed958f91da89>
```

9.2.2 Сброс регистрации абонента на регистраторе

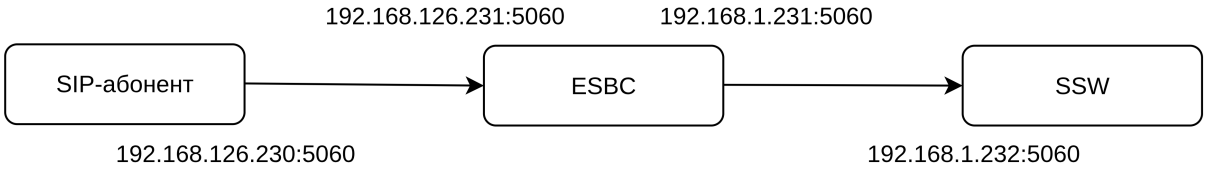
Функция сброса регистрации абонента на регистраторе используется для очистки записи о регистрации абонента на вышестоящем сервере. При включенной функции, если на ESBC будет удалена запись о регистрации абонента или он перерегистрируется на другом сервере, то в сторону регистратора будет отправлен запрос REGISTER с данными абонента и параметром Expires=0. Также запрос разрегистрации будет отправлен на регистратор в случае недоступности абонента, которая контролируется локальной обработкой регистрации.

> Функция включается на абонентском интерфейсе командой **registration forced-unregistration enable.**

Пример сброса регистрации абонента на регистраторе при ручном удалении регистрации на ESBC:

При ручном удалении регистрации абонента на ESBC будет отправлен запрос REGISTER с данными абонента и параметром Expires=0 на тот регистратор, где он зарегистрирован.

Схема:



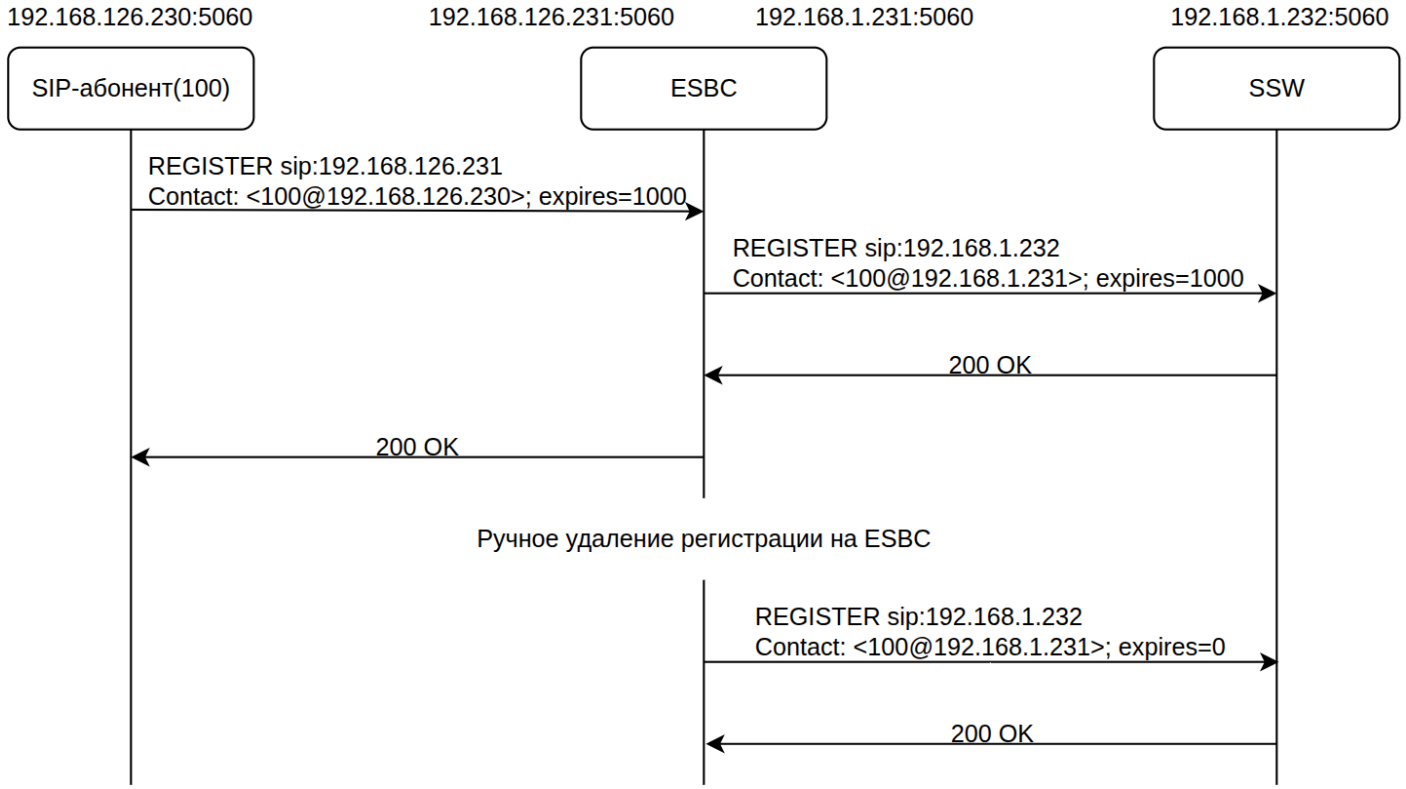
На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW.

Пример настройки абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение разрегистрации абонентов на регистраторе:
vesbc(config-esbc-user-interface-sip)# registration forced-unregistration enable

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Теперь при удалении регистраций, прошедших через этот абонентский интерфейс, в сторону SSW будут отправлены запросы REGISTER с данными удалённых абонентов с параметром Expires=0.

Схема прохождения запросов:

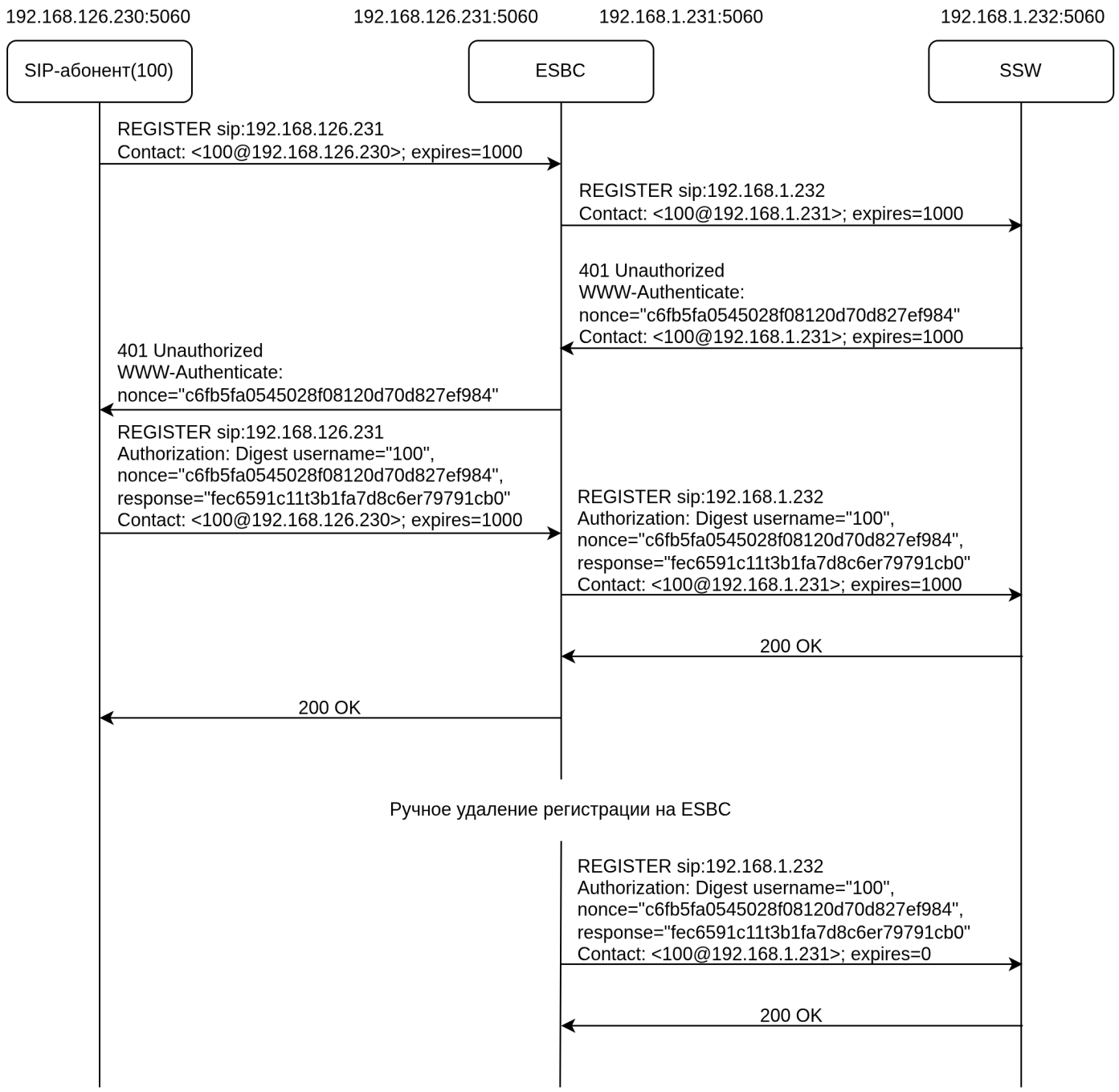


Пример ручного удаления регистрации через CLI:

```
#Вывод всех зарегистрированных абонентов:
vesbc# show esbc users sip
 Total AORs:       1
 Total Contacts:   1

User AOR                          User       Contact
                                  type       count
------------------------------    --------   --------
100@192.168.126.230               SIP        1

#Вывод детальной информации абонента с AOR 100@192.168.126.230:
vesbc# show esbc users sip 100@192.168.126.230 detailed
 User AOR:       100@192.168.126.230
 User type:      SIP
 Contact count:  1
IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:100@192.168.126.230:   192.168.126.230   ABONENT      TO_SSW            192.168.1.232
<sip:100@192.168.1.231:50   Uncontrolled    980/1000
5060>
60;line=7d296a72328c2d1d5

042ec9ad4db2bc6>

#Удаление регистрации абонента с AOR 100@192.168.126.230:
vesbc# clear esbc registration pattern-aor 100@192.168.126.230
Following records will be deleted:
-------------------------------------   -------------------------------------
100@192.168.126.230
Do you really want to clear user registrations with pattern? (y/N): y
vesbc#
```

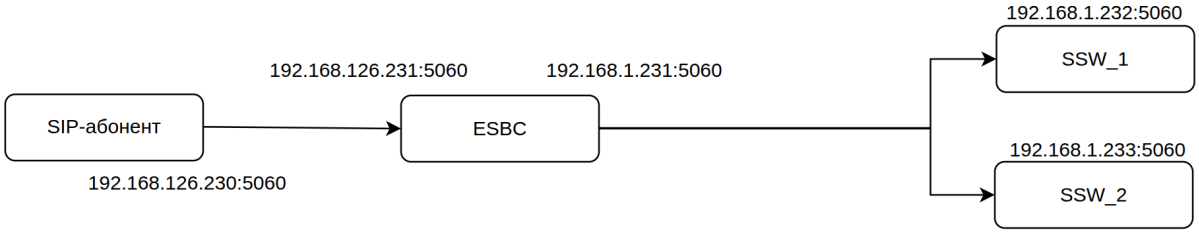
Также, если от абонента пришёл запрос регистрации с аутентификационными данными, они будут подставлены в запрос разрегистрации. В большинстве случаев сервер регистрации на такой запрос регистрации будет отправлять запрос аутентификации, но в некоторых случаях это позволит избежать 401 и 407 ответа.

Схема прохождения запросов с аутентификационными данными:



Пример сброса регистрации абонента при смене регистратора:

При регистрации абонента на другом направлении будет отправлен запрос REGISTER с данными абонента и параметром Expires=0 на тот регистратор, где он был зарегистрирован.



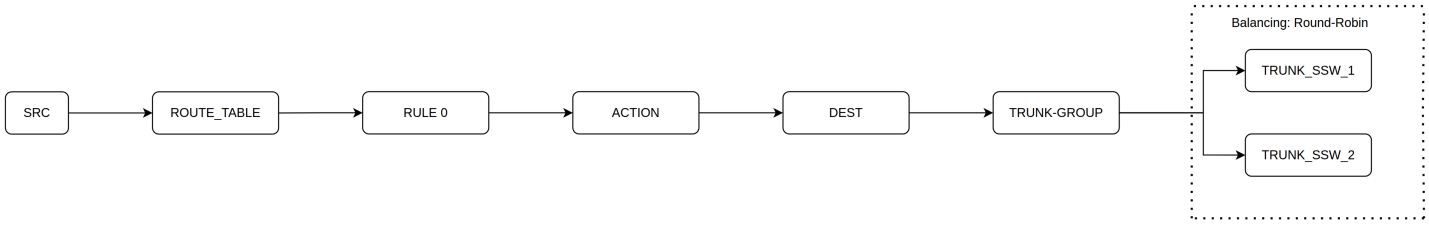
На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и

два транка, смотрящие в сторону SSW_1 и SSW_2. Транки находятся в транковой группе с балансировкой round robin. Поэтому сервер регистрации абонента будет чередоваться между SSW_1 и SSW_2.

Пример настройки абонентского интерфейса:

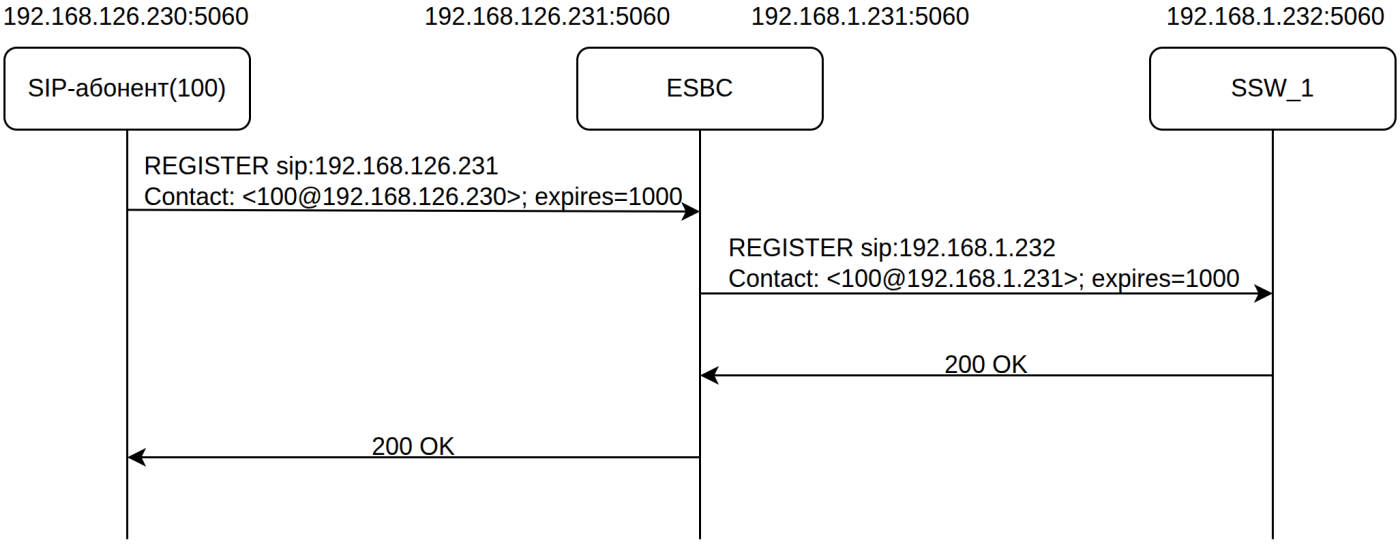
```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_TRUNK_GROUP

#Включение разрегистрации абонентов на регистраторе:
vesbc(config-esbc-user-interface-sip)# registration forced-unregistration enable

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Схема таблицы маршрутизации:

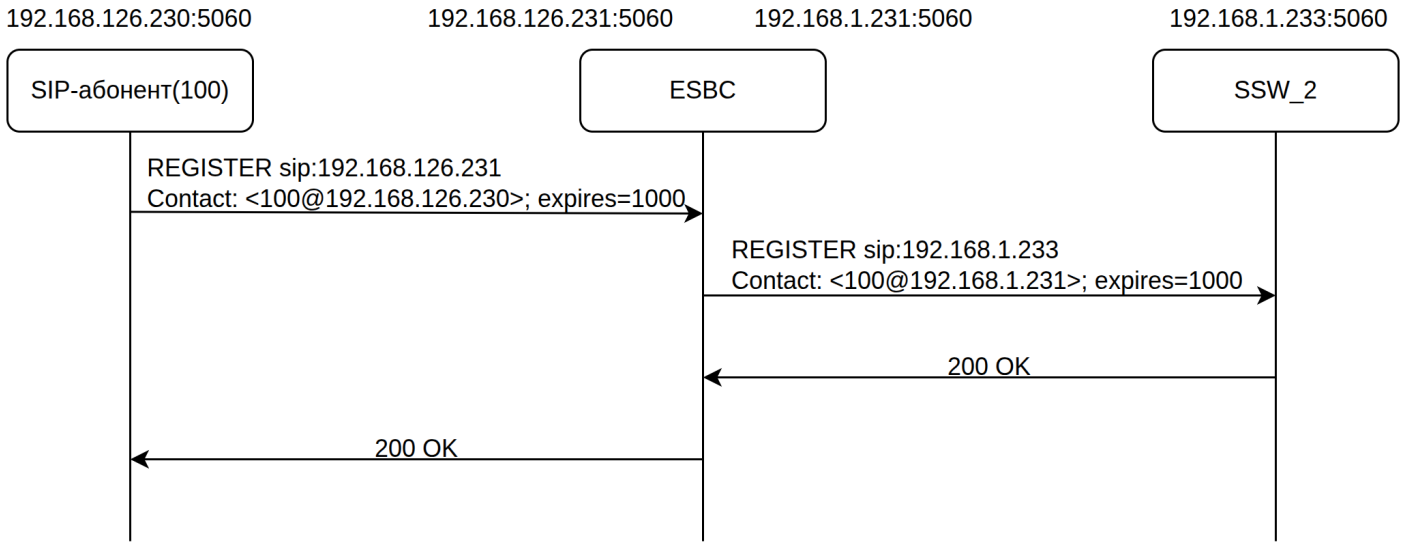


Алгоритм прохождения запросов:

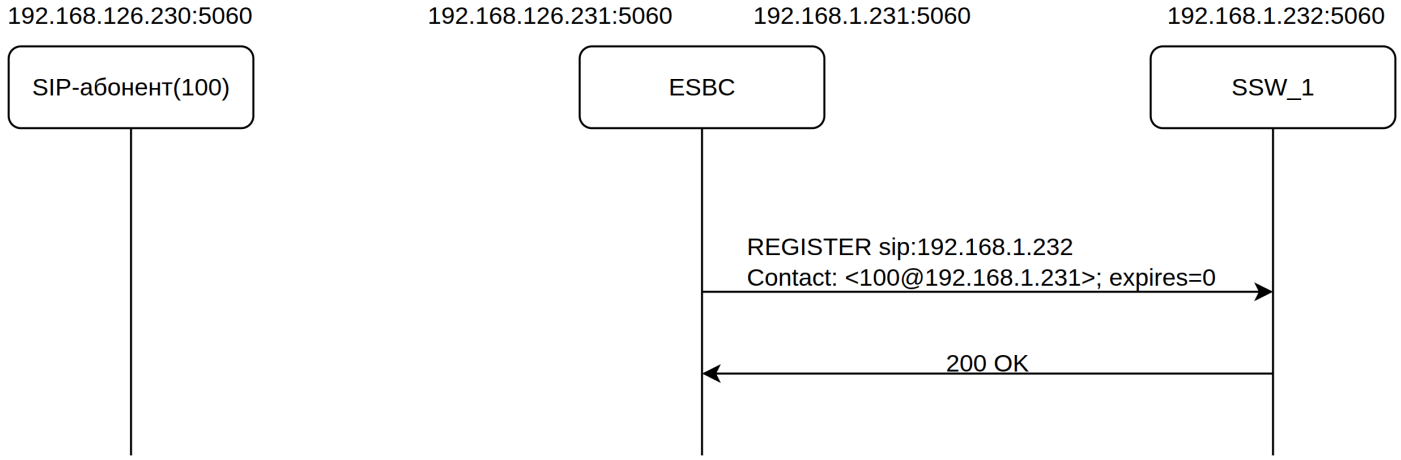
1. Абонент регистрируется на SSW_1:



2. Абонент перерегистрируется, и запрос согласно таблице маршрутизации направляется на SSW_2:



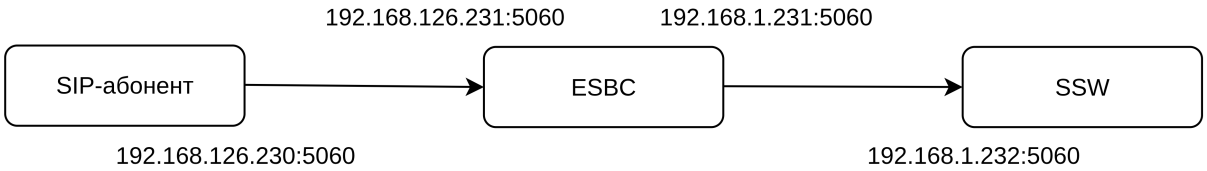
3. В сторону SSW_1 будет отправлен запрос разрегистрации, так как абонент сменил сервер регистрации на SSW_2:



Пример сброса регистрации при недоступности абонента:

При недоступности абонента будет отправлен запрос REGISTER с данными абонента и параметром Expires=0 на тот регистратор, где он зарегистрирован. Контроль доступности абонента работает при включении локальной обработки регистраций.

Схема:



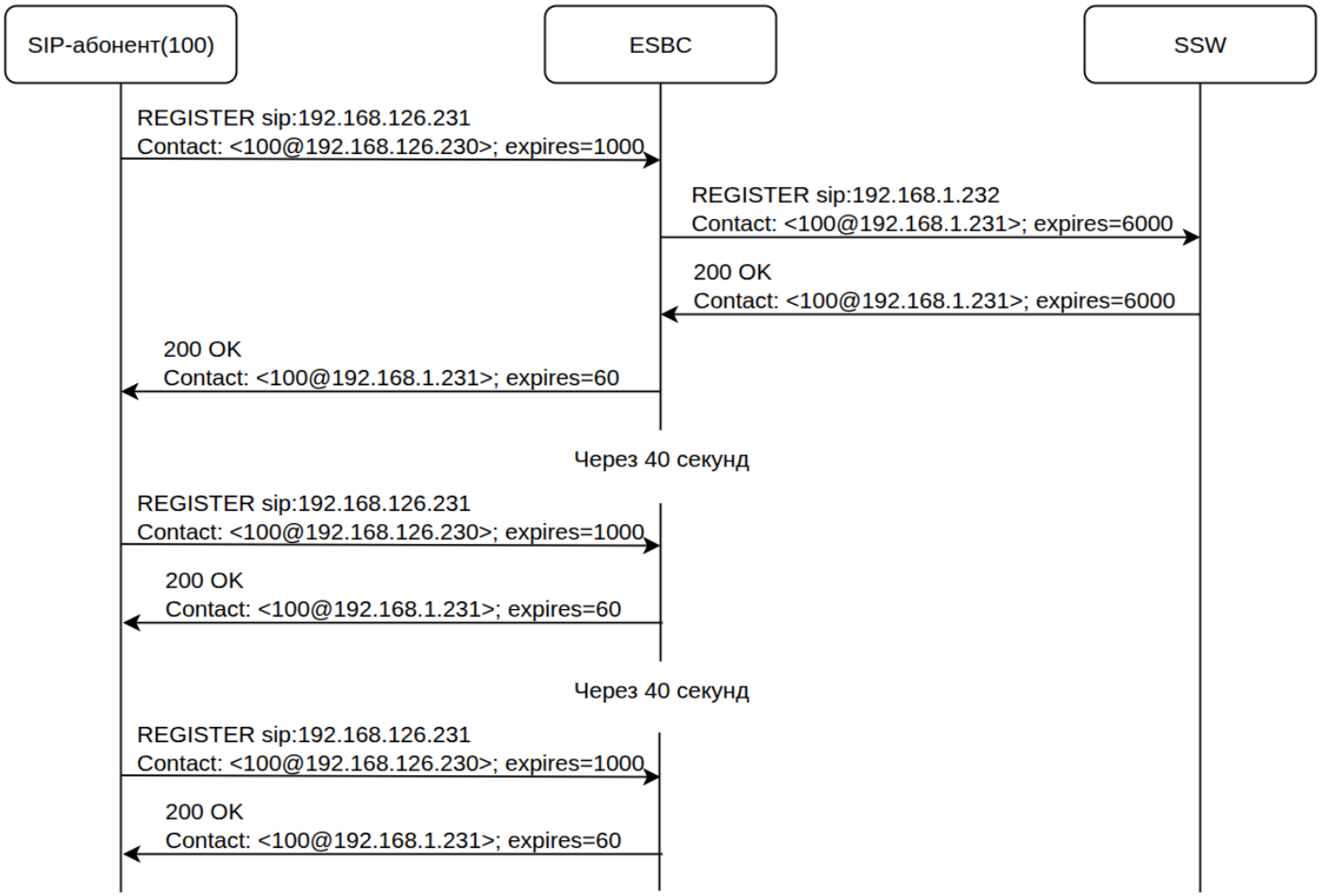
На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW.

Пример настройки абонентского интерфейса:

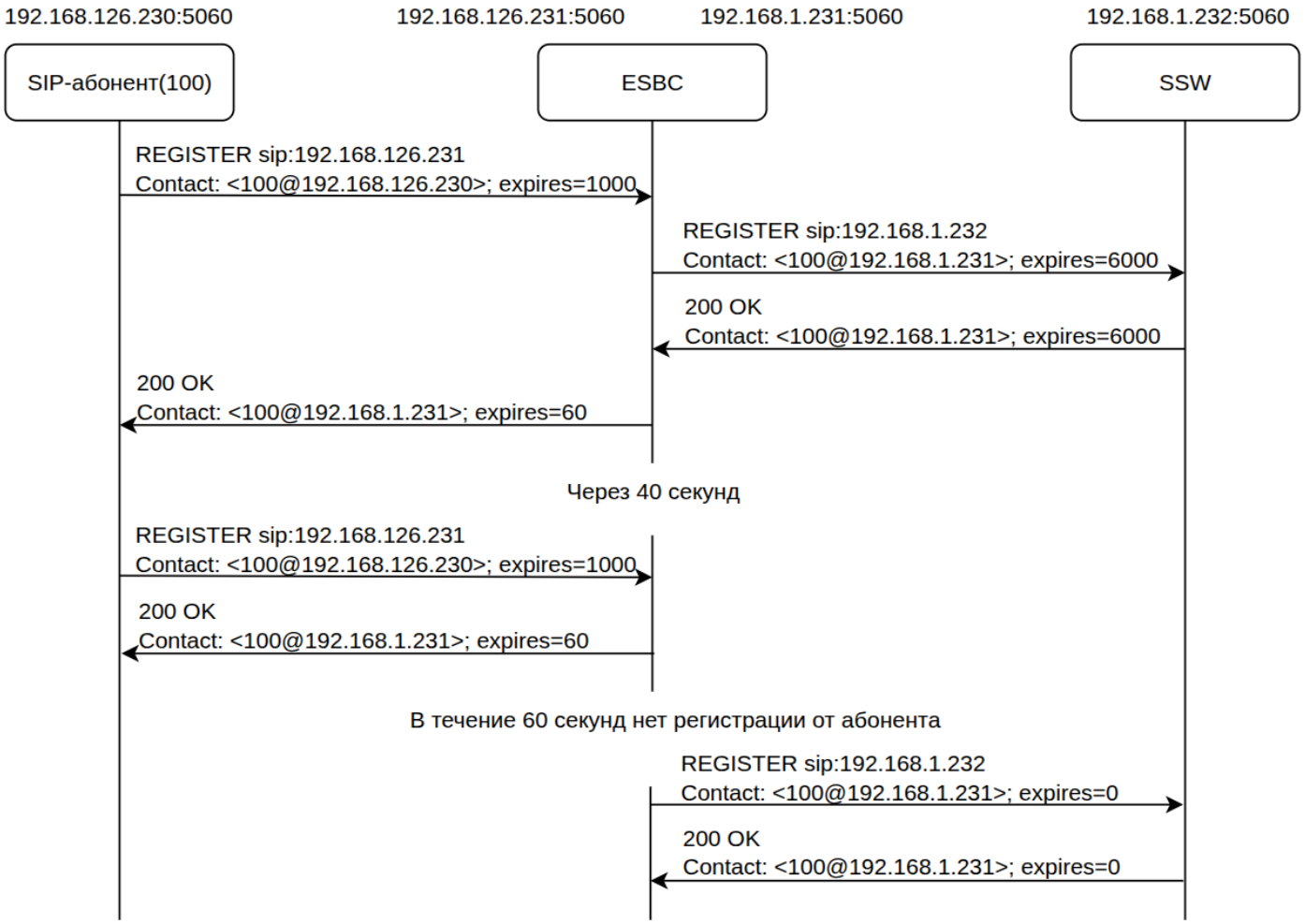
```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение разрегистрации абонентов на регистраторе:
vesbc(config-esbc-user-interface-sip)# registration forced-unregistration enable

#Включение локальной обработки регистраций:
vesbc(config-esbc-user-interface-sip)# registration simulation enable

#Использование минимального времени регистрации на сервере:
vesbc(config-esbc-user-interface-sip)# registration simulation min-reg-expires 6000

#Использование максимального времени регистрации абонента:
vesbc(config-esbc-user-interface-sip)# registration simulation max-user-expires 60

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Алгоритм работы:

1. Абонент отправляет запрос регистрации на ESBC со своим временем регистрации. Если оно меньше 6000, то ESBC автоматически подставит Expires=6000 в отправленный на SSW запрос REGISTER. А в ответе 200 OK, который приходит к абоненту от ESBC, будет подставлен Expires=60, из-за чего абонент будет чаще перерегистрироваться.



2. Если от абонента не пришла регистрация в течение 60 секунд, такой абонент считается недоступным, и отправляется разрегистрация на SSW.



9.2.3 Инициация перерегистрации абонента через SIP-NOTIFY

SIP-телефоны и другие пользовательские агенты могут управляться удалённо с помощью SIP-метода NOTIFY. Для управления устройством применяются специальные события **«Event»**, которые позволяют серверу инициировать действия на SIP-телефоне.

ESBC поддерживает функцию отправки управляющих NOTIFY для инициации обновления регистрации абонента при недоступности транка или ручном удалении регистрации.

Примеры инициации перерегистрации абонента через передачу SIP-NOTIFY.

Для инициации обновления регистрации абонента используется SIP-метод NOTIFY. Для разных устройств могут быть использованы свои значения заголовка **«Event»**.

Для ввода значения заголовка **«Event»** с использованием пробела используйте двойные кавычки, например:

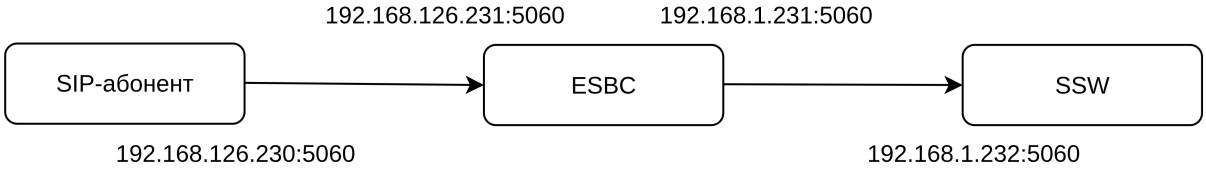
```
vesbc(config-esbc-user-interface-sip)# sip notify-unavailable event 'reg;id="software
client"'
```

Передача SIP-сообщения NOTIFY в случае недоступности сервера регистрации:

В случае недоступности сервера регистрации зарегистрированному абоненту будет отправлено SIP-сообщение NOTIFY с назначенным значением заголовка **«Event»**.

> ⚠ По умолчанию установлено значение: "**check-sync**".

Схема:



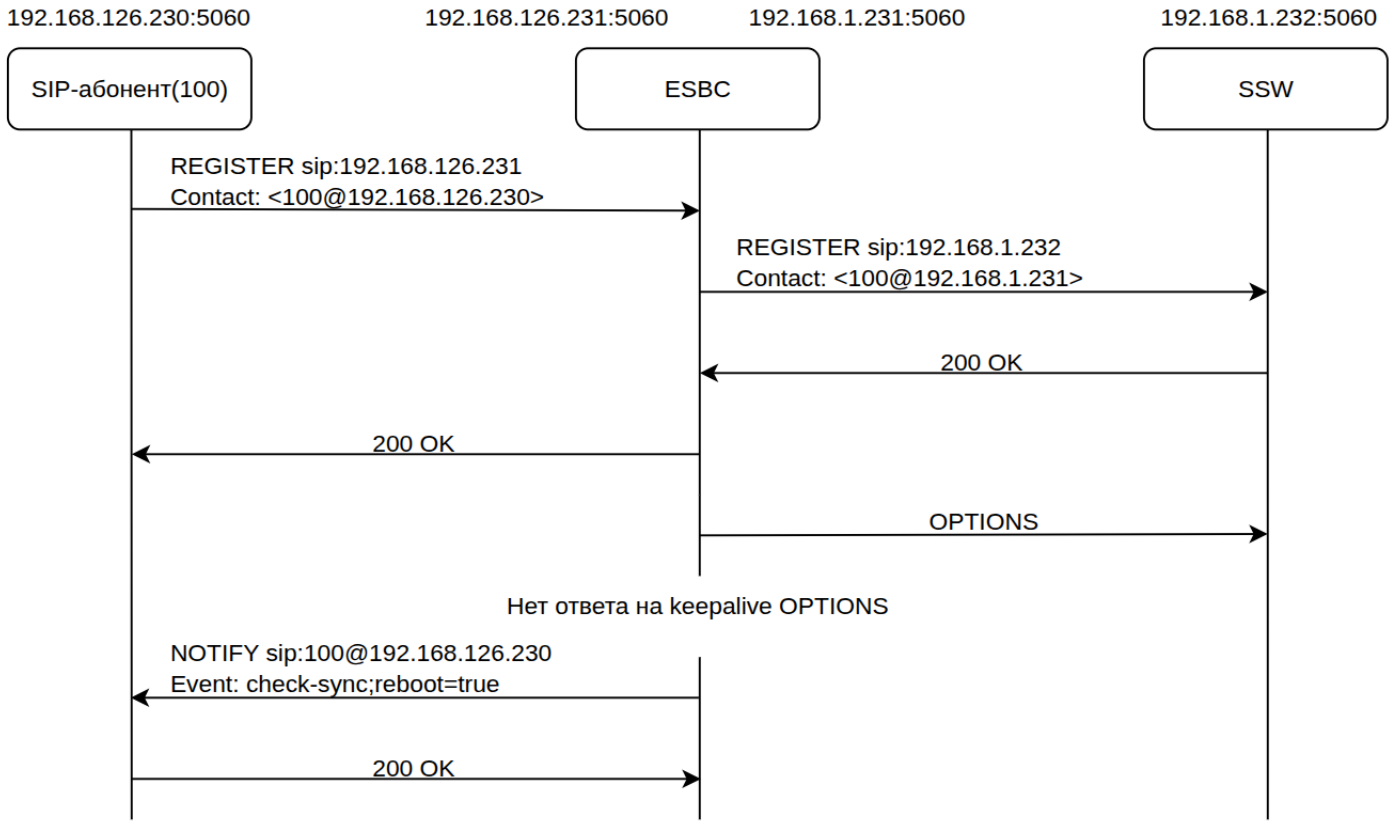
На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW. На транке настроен SIP-профиль с контролем доступности направления.

Пример настройки абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение отправки сообщения NOTIFY с заголовком "Event: check-sync;reboot=true" в случае
недоступности сервера регистрации:
vesbc(config-esbc-user-interface-sip)# sip notify-unavailable enable event
vesbc(config-esbc-user-interface-sip)# sip notify-unavailable event "check-sync;reboot=true"

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Теперь при недоступности сервера SSW зарегистрированному абоненту будет отправлен NOTIFY с заголовком **«Event: check-sync;reboot=true»**.

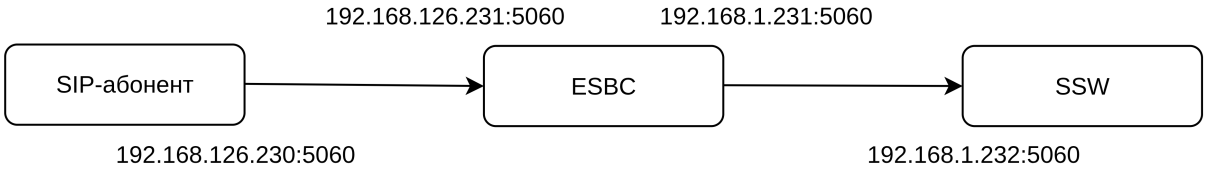
Схема прохождения запросов:



Передача сообщения NOTIFY при ручном удалении регистрации абонента:

При ручном удалении регистрации абонента для отправки SIP-сообщения NOTIFY с определённым значением заголовка «**Event**» нужно ввести его значение.

Схема:

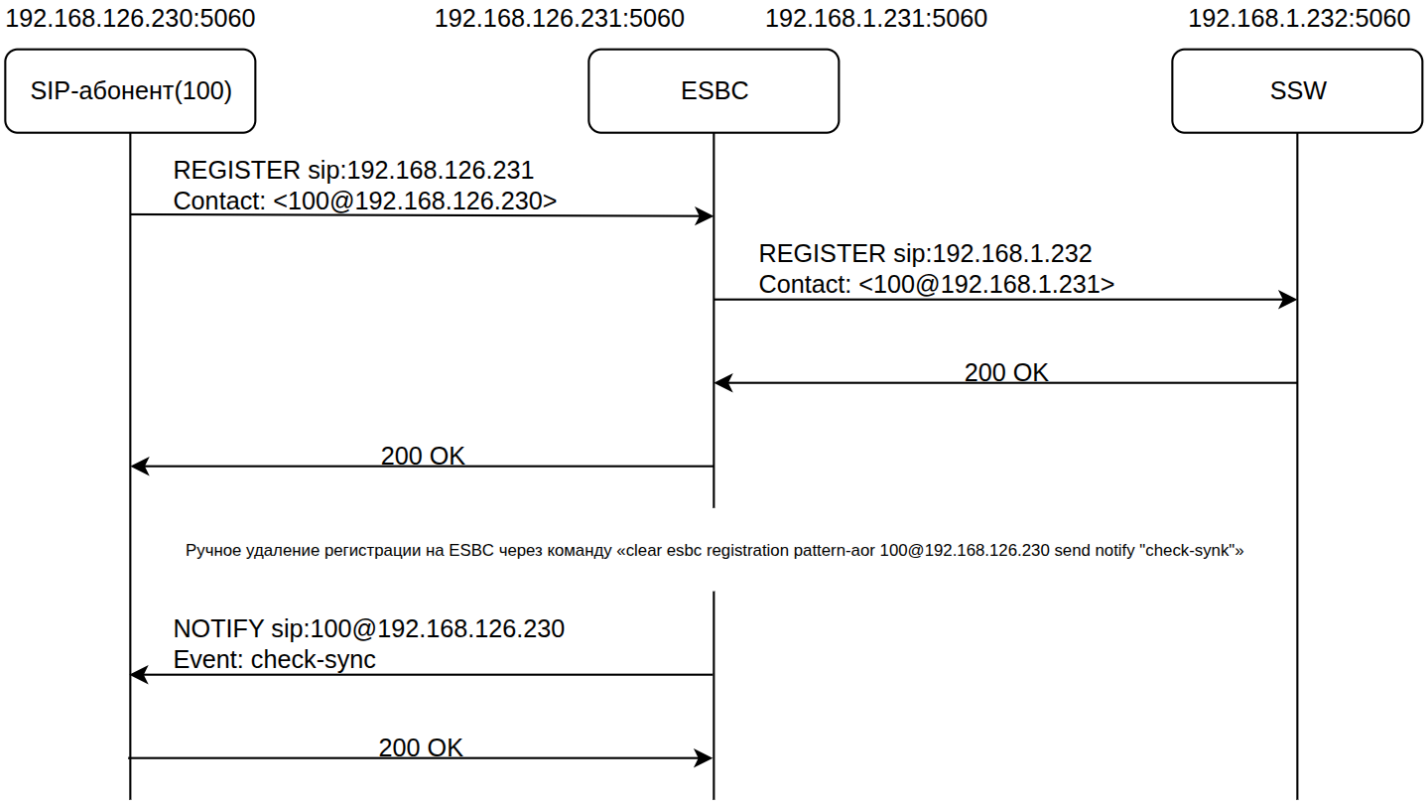


На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW.

Пример ручного удаления регистрации с отправкой NOTIFY через CLI:

```
#Вывод всех зарегистрированных абонентов:
vesbc# show esbc users sip
 Total AORs:       1
 Total Contacts:   1

User AOR                          User       Contact
                                  type       count
------------------------------    --------   --------
100@192.168.126.230               SIP        1

#Вывод детальной информации абонента с AOR 100@192.168.126.230:
vesbc# show esbc users sip 100@192.168.126.230 detailed
 User AOR:       100@192.168.126.230
 User type:      SIP
 Contact count:  1
IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:100@192.168.126.230:   192.168.126.230   ABONENT      TO_SSW            192.168.1.232
<sip:100@192.168.1.231:50   Uncontrolled    2950/3000
5060>
60;line=6d29aa7232hg789d5

0412dwad4ds2bc6>

#Удаление регистрации абонента с AOR 100@192.168.126.230 с отправкой NOTIFY с заголовком
"Event: check-sync":
vesbc# clear esbc registration pattern-aor 100@192.168.126.230 send notify "check-sync"
Following records will be deleted:
-------------------------------------   -------------------------------------
100@192.168.126.230
Do you really want to clear user registrations with pattern? (y/N): y
vesbc#
```

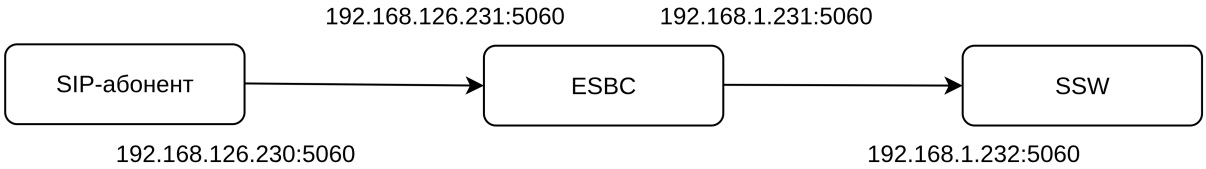
Схема прохождения запросов:



Передача сообщения NOTIFY для устройств Cisco в случае недоступности сервера регистрации:

В случае недоступности сервера регистрации зарегистрированному абоненту Cisco будет отправлено SIP-сообщение NOTIFY с заголовком «**Event: service-control**» и телом с сообщением «**action=restart**» или «**action=reset**».

Схема:



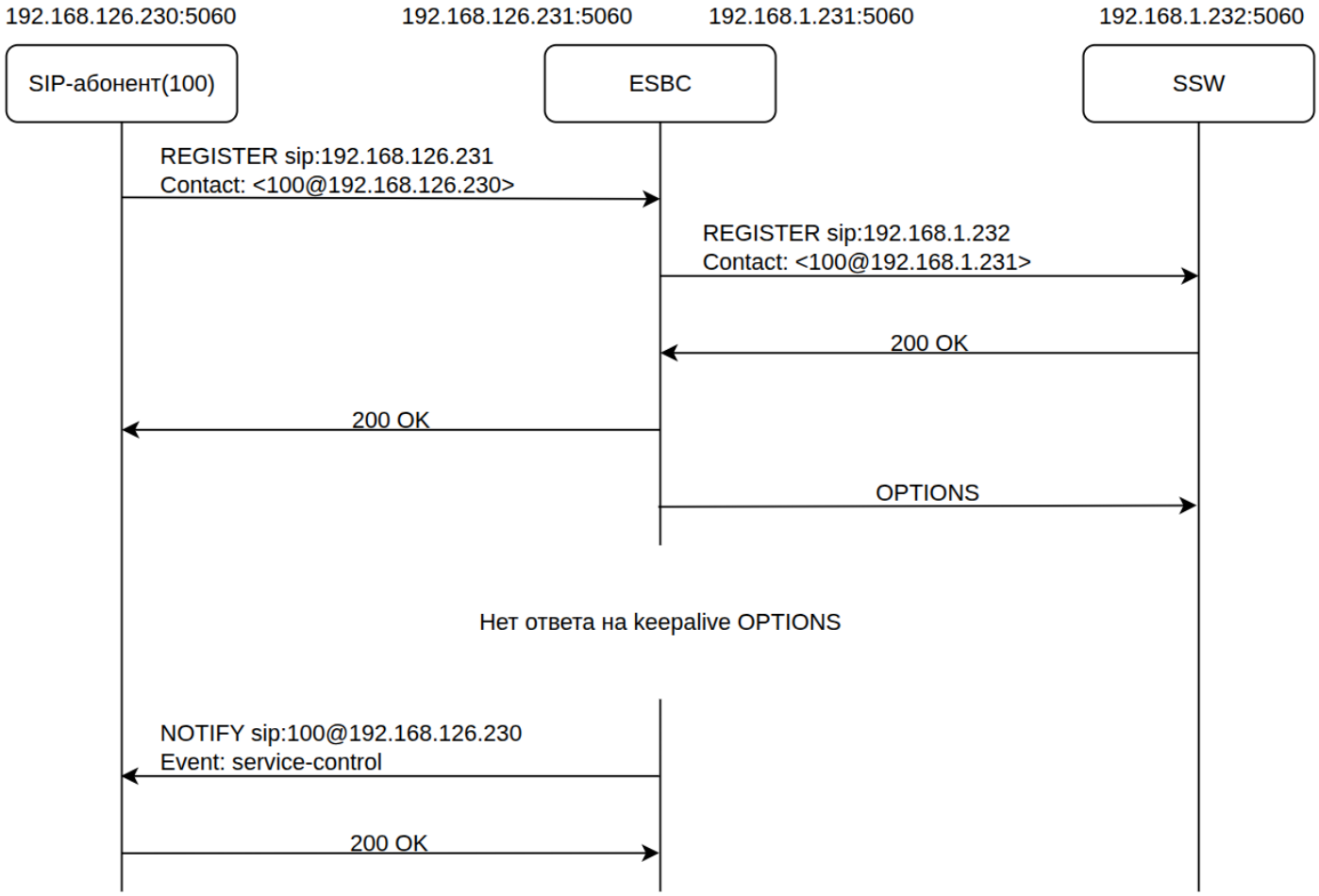
На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW. На транке настроен SIP-профиль с контролем доступности направления.

Пример настройки абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Включение отправки сообщения NOTIFY для устройств CISCO с заголовком "Event: service-control"
 в случае недоступности сервера регистрации:
vesbc(config-esbc-user-interface-sip)# sip notify-unavailable enable cisco-service-control

#Назначение параметра "action=reset" в тело сообщения NOTIFY для устройств CISCO в случае
недоступности сервера регистрации:
vesbc(config-esbc-user-interface-sip)# sip notify-unavailable cisco-service-control reset

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Теперь при недоступности сервера SSW зарегистрированному абоненту будет отправлен NOTIFY с заголовком «**Event: service-control**» и телом со значением «**action=reset**».

Схема прохождения запросов:



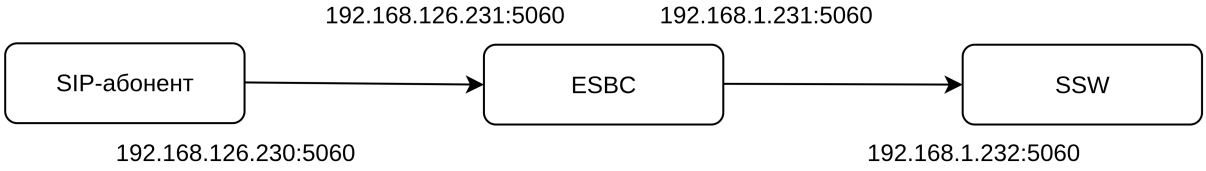
Пример сообщения NOTIFY:

```
Session Initiation Protocol (NOTIFY)
    Request-Line: NOTIFY sip:100@192.168.126.230:5060;transport=udp SIP/2.0
    Message Header
        Via: SIP/2.0/UDP 192.168.126.231:5060;rport;branch=z9hG4bKPjebbcb9c2-9484-4d61-8d30-5cf
a71fc3b06
        Max-Forwards: 70
        From: <sip:100@192.168.126.231>;tag=a2d23820-d7f8-44bc-9ab0-b5824cefbd31
        To: <sip:100@192.168.126.231>
        Call-ID: fb7dbd3675a6cb8111a9a0d611ba4cae
        [Generated Call-ID: fb7dbd3675a6cb8111a9a0d611ba4cae]
        CSeq: 9607 NOTIFY
        Event: service-control
        Content-Type: text/plain
        Content-Length:    69

    Message Body
        Line-based text data: text/plain (2 lines)
            action=reset\r\n
            RegisterCallId={a5cda0d8-98a1-1200-e8a8-6813e20921bf}\r\n
```

Передача сообщения NOTIFY для устройств Cisco при ручном удалении регистрации абонента:

При ручном удалении регистрации абонента Cisco можно отправить SIP-сообщение NOTIFY с заголовком **«Event: service-control»** и телом с сообщением **«action=restart»** или **«action=reset»**.

Схема:

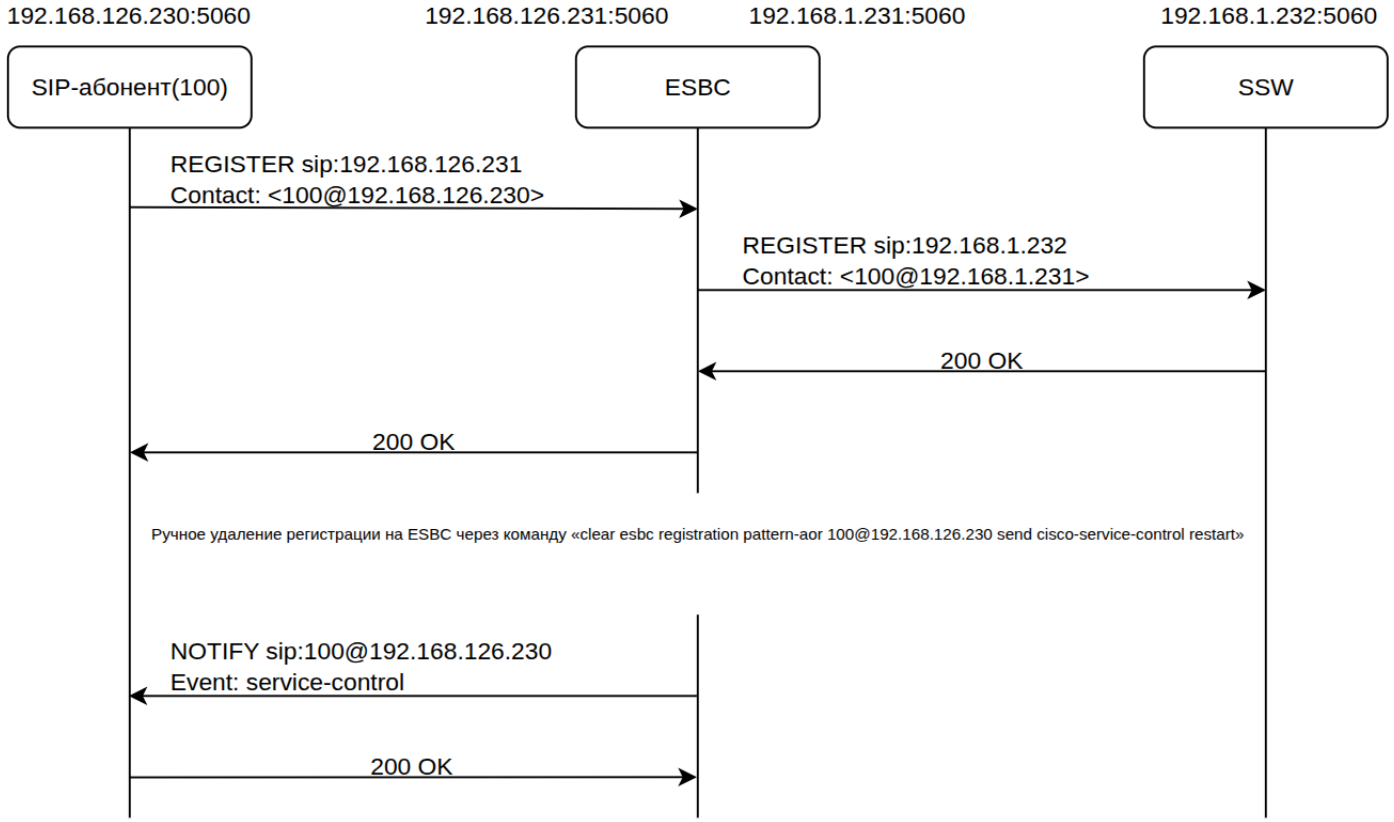


На ESBC настроена конфигурация с интерфейсами, смотрящими в сторону SIP-абонентов и SSW, с IP-адресами 192.168.126.231 и 192.168.1.231 соответственно. Также есть абонентский интерфейс и транк, смотрящий в сторону SSW.

Пример ручного удаления регистрации с отправкой NOTIFY через CLI:

```
#Вывод всех зарегистрированных абонентов:
vesbc# show esbc users sip
 Total AORs:       1
 Total Contacts:   1

User AOR                          User       Contact
                                  type       count
------------------------------    --------   --------
100@192.168.126.230               SIP        1

#Вывод детальной информации абонента с AOR 100@192.168.126.230:
vesbc# show esbc users sip 100@192.168.126.230 detailed
 User AOR:       100@192.168.126.230
 User type:      SIP
 Contact count:  1
IN User contact             IP address of     User         Trunk name        IP address of
OUT Trunk contact           Status          Expires
                            user              interface                       registrar
                                              name
-------------------------   ---------------   ----------   ---------------   ---------------
-------------------------   -------------   ------------
<sip:100@192.168.126.230:   192.168.126.230   ABONENT      TO_SSW            192.168.1.232
<sip:100@192.168.1.231:50   Uncontrolled    2950/3000
5060>
60;line=6d29aa7232hg789d5

0412dwad4ds2bc6>

#Удаление регистрации абонента с AOR 100@192.168.126.230 с отправкой NOTIFY с заголовком
"Event: service-control" и "action=restart" в теле сообщения:
vesbc# clear esbc registration pattern-aor 100@192.168.126.230 send cisco-service-control
restart
Following records will be deleted:
-------------------------------------   -------------------------------------
100@192.168.126.230
Do you really want to clear user registrations with pattern? (y/N): y
vesbc#
```

Схема прохождения запросов:



Пример сообщения NOTIFY:

```
Session Initiation Protocol (NOTIFY)
    Request-Line: NOTIFY sip:100@192.168.126.230:5060;transport=udp SIP/2.0
    Message Header
        Via: SIP/2.0/UDP 192.168.126.231:5060;rport;branch=z9hG4bKPj362e9c4a-f201-4c55-86c8-
af862ab6df9b
        Max-Forwards: 70
        From: <sip:100@192.168.126.231>;tag=559d013f-ece6-444f-a239-8c5ecab73392
        To: <sip:100@192.168.126.231>
        Call-ID: fb7dbd3675a6cb8111a9a0d611ba4cae
        [Generated Call-ID: fb7dbd3675a6cb8111a9a0d611ba4cae]
        CSeq: 9607 NOTIFY
        Event: service-control
        Content-Type: text/plain
        Content-Length:    71
    Message Body
        Line-based text data: text/plain (2 lines)
            action=restart\r\n
            RegisterCallId={a5cda0d8-98a1-1200-e8a8-6813e20921bf}\r\n
```

При включении двух типов работы NOTIFY, ESBC отправит сначала обычный **«Event»**. Если ответ будет отличен от 200 или его не будет вовсе, запрос завершится таймаутом, и ESBC отправит NOTIFY с **«Event: service-control»**.

9.3 Настройка SIP-транков

SIP-транк представляет собой интерфейс для подключения к вышестоящему SIP-устройству (IP АТС/ SIP-proxy/Удаленный SSW и др.) или группе вышестоящих устройств при включении динамического режима работы транка. При включении динамического режима работы в конфигурации необходимо задать адрес и порт удалённой стороны или диапазон адресов и портов. Эти параметры используются для идентификации источника запроса.

⚠ На транке запрещена обработка входящих запросов REGISTER.

Для создания SIP-транка необходимо настроить:

- Адрес удалённой стороны (или диапазон адресов для динамического режима);
- Порт удалённой стороны (или диапазон адресов для динамического режима);
- SIP-транспорт;
- Медиаресурсы.

Эти настройки являются обязательными. Описание конфигурирования и базовой схемы применения представлено в разделе Примеры настройки ESBC.

Помимо этого SIP-транк содержит набор следующих настроек:

- Таблица маршрутизации;
- Таблица модификаций (для входящих и исходящих сообщений);
- SIP-профиль;
- Медиапрофиль;
- Профиль безопасности;
- Режим работы за NAT;
- Режим локальной аутентификации запросов;
- Режим клиентской регистрации транка;
- Public IP;
- STUN-сервер;
- QoS;
- Контроль входящего и исходящего трафика;
- Динамический режим. Используется для подключения к группе вышестоящих SIP-устройств (IP АТС/ SIP-proxy/Удаленный SSW и др.);
- Опция "Доверенная сеть" для переадресаций;
- AAA профиль;
- SIP-домен. При настройке домен будет использоваться в host-part URI в заголовках To и From для исходящих сообщений. Во входящих сообщениях будет осуществляться проверка домена в заголовке From.

Подробное описание параметров всех настроек можно найти в разделе Настройки SIP-транка справочника команд CLI.

⚠ Создание транков с одинаковым SIP-транспортом и IP:Port разрешено только в случае, если отличается SIP-домен.

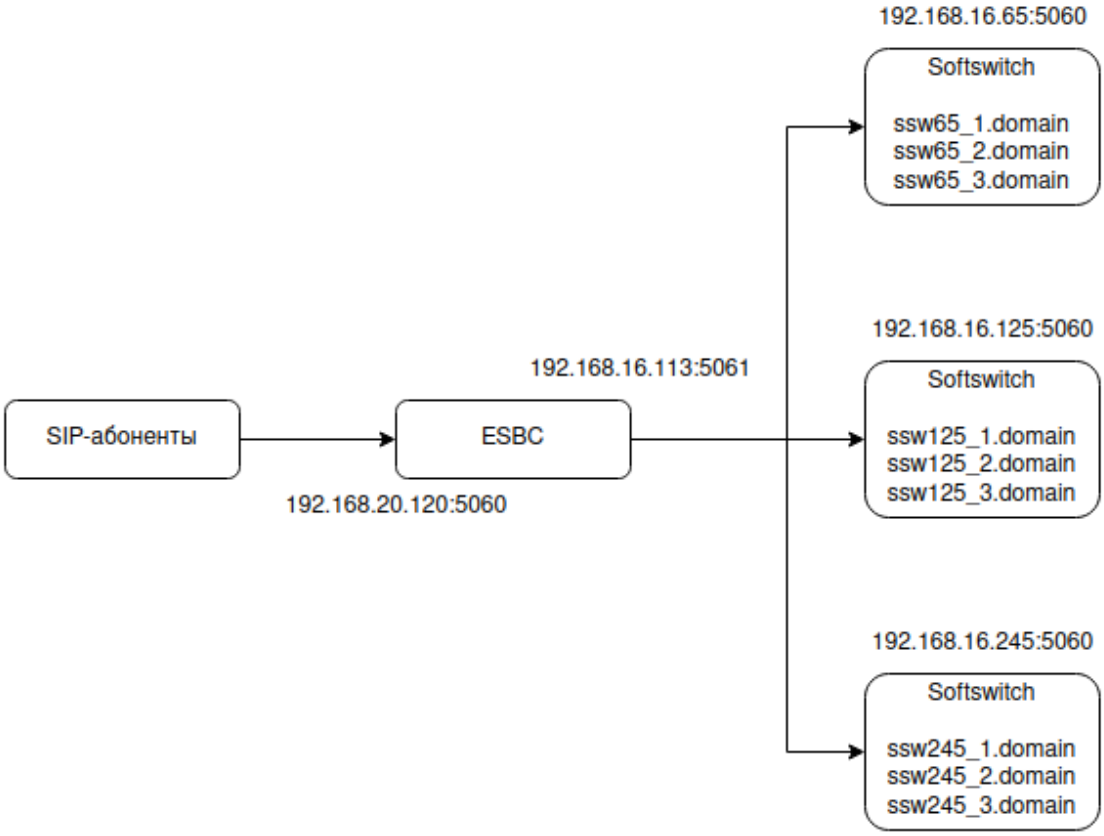
⚠ С целью повышения безопасности, входящие запросы OPTIONS, поступающие в транк будут игнорироваться. Для изменения поведения см. раздел Игнорирование OPTIONS.

9.3.1 Динамический режим транка

При использовании динамического режима фактический адрес назначения не задаётся в конфигурации, а определяется внешним сервисом при первом запросе, который должен быть смаршрутизирован в этот транк.

> ⚠ В текущей версии ПО в качестве внешнего сервиса могут выступать DNS и RADIUS.

Схема применения:



Описание:

SIP-абоненты (IP-телефон/VoIP-шлюз/Мобильный SIP-клиент и т. д.) отправляют SIP-запросы на IP-адрес 192.168.20.120 порт 5060.
ESBC должен смаршрутизировать запрос в зависимости от домена в hostname части RURI (или заголовка To), полученного от абонента. Вызовы/регистрации могут быть смаршрутизированы на один из трёх Softswitch (IP АТС/SIP-proxy и т. д), на каждом из них настроено несколько доменов.

> ⚠ Алгоритм поиска домена для определения адреса динамического транка:
> 1. Если в Request-URI указан домен, то он будет использоваться для определения адреса транка;
> 2. Если в Request-URI указан адрес, то проверяется значение в заголовке To:
> a. Если в To указан домен, то он будет использоваться для определения адреса транка;
> b. Если в To указан адрес, то произойдёт отбой вызова.

Для реализации данной схемы общий алгоритм настройки следующий:

1. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.16.113/24
vesbc(config-if-gi)# ip firewall disable
```

2. Настроить IP-адрес на внешнем интерфейсе в сторону абонентов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "ABONENTS"
vesbc(config-if-gi)# ip address 192.168.20.120/24
```

3. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5061
```

4. Создать SIP-транспорт в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
```

5. Создать медиаресурсы для согласования и передачи голоса на плече SSW --- ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000–65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC --- Абонентский шлюз/SIP-абоненты:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_ABONENTS
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать абонентский интерфейс в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_ABONENTS

#Если абоненты находятся за NAT, выполнить команду:
vesbc(config-esbc-user-interface-sip)# nat comedia-mode on
```

8. Настроить внешний сервис, создать SIP-транк в динамическом режиме и настроить маршрутизацию.

DNS

1. Настроить DNS:

```
vesbc(config)# domain lookup enable
vesbc(config)# domain nameserver 192.168.16.200
vesbc(config)#
```

Описание всех доступных настроек DNS приведено в разделе Настройка DNS Справочника команд CLI.

2. Создать динамический SIP-транк, в качестве адреса указать подсеть, в которой находятся SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip SSW_DYNAMIC
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.0/24
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# dynamic-mode dns
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

3. Создать таблицу маршрутизации и добавить туда правила, по которым вызовы, приходящие с абонентов, будут маршрутизироваться на SIP-транк с динамическим режимом:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk SSW_DYNAMIC
```

4. Привязать созданную таблицу маршрутизации к абонентскому интерфейсу:

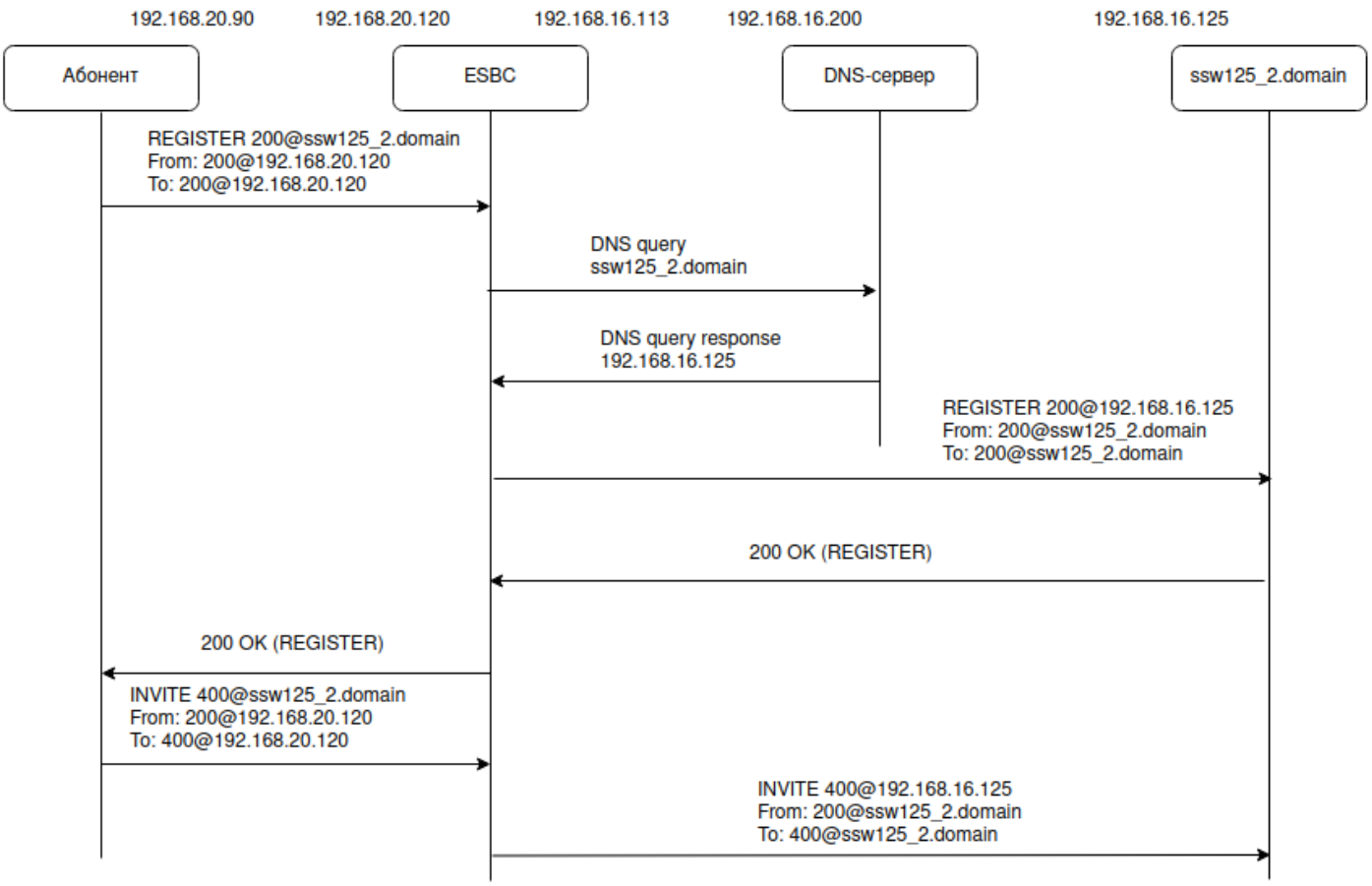
```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
```

5. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

Пример:

SIP-абонент отправляет сообщение REGISTER, в hostname RURI указывает **ssw125_2.domain.** ESBC отправляет запрос на DNS-сервер (192.168.16.200) для определения адреса назначения, внешний сервис в ответ присылает адрес SIP-сервера (192.168.16.125), на который нужно отправить запрос. ESBC отправляет регистрацию на указанный адрес, подставляя в заголовки To и From **ssw125_2.domain**, последующие запросы с этого абонента при указании того же домена будут отправляться в транк 192.168.16.125 без предварительного обращения к внешнему сервису.



⚠ Исходящий запрос на динамический транк будет отправлен на тот же порт, что указан в RURI входящего запроса. Если порт явно не указан, то запрос отправится на стандартный порт 5060.

Для определения входящего запроса из транка с динамическим режимом используется адрес/маска подсети из **remote address** и порт/диапазон портов из **remote port**.

RADIUS

1. Настроить RADIUS-сервер.

Пример конфигурации freeradius:

```
Файл clients.conf:
client ESBC {
#Адрес интерфейса ESBC, с которого будут отправляться запросы на RADIUS-сервер:
    ipaddr = 192.168.16.113
#Ключ для аутентификации клиента:
    secret = password
}

Файл users:
#Обязательный пароль при использовании динамического режима:
ssw125_2.domain Cleartext-Password := "domain_resolve"
#Адрес, который сервер отправит в ответе на Access-Request с User-Name: ssw125_2.domain:
    Framed-IP-Address = 192.168.16.125
```

2. Задать параметры RADIUS-сервера на ESBC:

```
vesbc(config)# radius-server host 192.168.16.250

#Пароль, который должен совпадать с secret на сервере:
vesbc(config-radius-server)# key ascii-text password
vesbc(config-radius-server)# usage voip

#Адрес интерфейса, с которого будут отправляться запросы:
vesbc(config-radius-server)# source-address 192.168.16.113
```

Описание всех доступных настроек RADIUS-сервера приведено в разделе Настройка AAA Справочника команд CLI.

3. Создать RADIUS-профиль, добавить в его конфигурацию RADIUS-сервер:

```
vesbc(config)# esbc
vesbc(config-esbc)# radius profile RADIUS_PROFILE
vesbc(config-radius-profile)# radius-server host 192.168.16.250
```

Описание всех доступных настроек RADIUS-профиля приведено в разделе Настройка ESBC Справочника команд CLI.

4. Создать динамический SIP-транк, в качестве адреса указать подсеть, в которой находятся SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip SSW_DYNAMIC
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.0/24
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# dynamic-mode radius RADIUS_PROFILE
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

5. Создать таблицу маршрутизации и добавить туда правила, по которым вызовы, приходящие с абонентов, будут маршрутизироваться на SIP-транк с динамическим режимом:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk SSW_DYNAMIC
```

6. Привязать созданную таблицу маршрутизации к абонентскому интерфейсу:

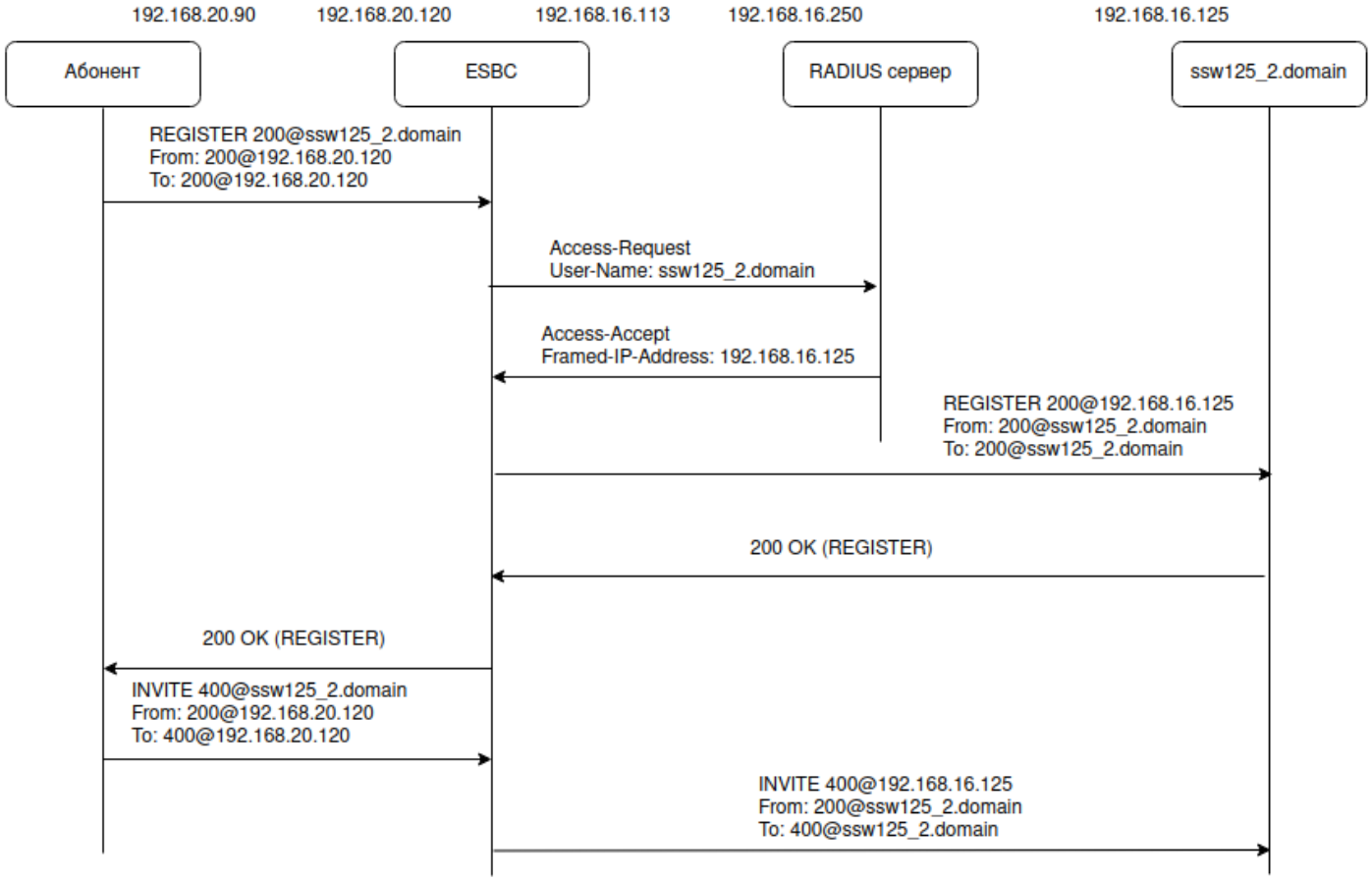
```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
```

7. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

Пример:

SIP-абонент отправляет сообщение REGISTER, в hostname RURI указывает **ssw125_2.domain.** ESBC для определения адреса назначения отправляет запрос (Access-Request) на RADIUS-сервер (192.168.16.250) с атрибутом User-Name, в котором содержится домен **ssw125_2.domain**.
RADIUS-сервер присылает Access-Accept с адресом SIP-сервера (192.168.16.125) в атрибуте Framed-IP-Address.
ESBC отправляет регистрацию на указанный адрес, подставляя в заголовки To и From **ssw125_2.domain**, последующие запросы с этого абонента при указании того же домена будут отправляться в транк 192.168.16.125 без предварительного обращения к внешнему сервису.



⚠ Исходящий запрос на динамический транк будет отправлен на тот же порт, что указан в RURI входящего запроса. Если порт явно не указан, то запрос отправится на стандартный порт 5060.

Для определения входящего запроса из транка с динамическим режимом используется адрес/маска подсети из **remote address** и порт/диапазон портов из **remote port**.

9.4 Настройка транковых групп

Транк-группа представляет собой набор транков различного типа (в текущей версии поддерживается только SIP-транк) и алгоритм балансировки нагрузки между ними. В текущей версии балансировка вызовов осуществляется алгоритмами round robin и weighted round robin.

Помимо этого группа содержит набор следующих настроек:

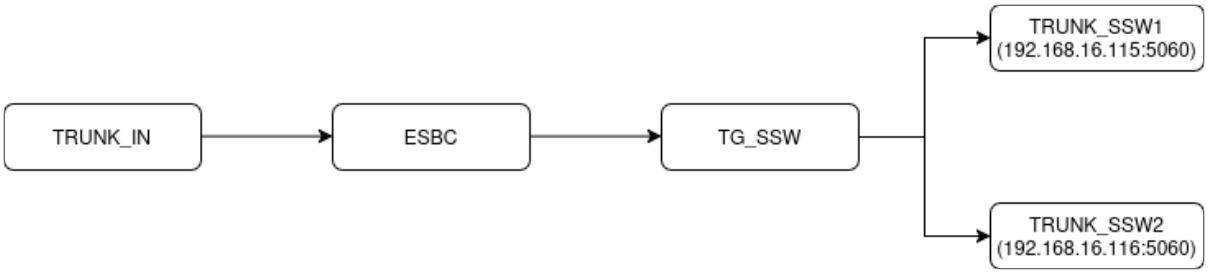
- Таблица маршрутизации;
- Медиапрофиль;
- Медиаресурсы;
- SIP-профиль;
- Профиль безопасности;
- Таблица модификаций (для входящих и исходящих сообщений);
- QoS;
- Public IP;
- STUN-сервер;
- Контроль входящего и исходящего трафика;
- Опция "Доверенная сеть" для переадресаций.

Логика работы:

Все перечисленные в предыдущем пункте настройки являются общими для всех транков, включенных в состав транковой группы. Это значит, что при отсутствии у транка, входящего в состав транковой группы, какой-либо из перечисленных настроек, будет использоваться настройка из транковой группы. Такой подход позволяет создавать множество транков с минимальным набором настроек, и, объединяя их в транковую группу, производить донастройку через нее. При необходимости можно изменить какие-либо параметры отдельно взятых транков из группы через индивидуальную настройку транков.

Пример работы общих настроек:

Схема:



На ESBC настроена транковая группа TG_SSW, в состав которой входят 2 транка TRUNK_SSW1 и TRUNK_SSW2, также настроен еще один транк TRUNK_IN, который не входит в состав транковой группы. Требуется настроить схему таким образом, чтобы вызовы, которые пришли с TRUNK_IN, маршрутизировались на TG_SSW, и наоборот, вызовы, которые пришли с TRUNK_SSW1 и TRUNK_SSW2, маршрутизировались на TRUNK_IN.

Решение:

1. Создать SIP-транспорт в сторону TRUNK_SSW1 и TRUNK_SSW2:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5065
```

2. Создать SIP-транспорт в сторону TRUNK_IN:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_IN
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# port 5067
```

3. Создать медиаресурсы для согласования и передачи голоса на плече TRUNK_IN --- ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_TRUNK_IN
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

4. Создать медиаресурсы для согласования и передачи голоса на плече ESBC --- TG_SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_TG_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113
```

5. Создать SIP-транк в сторону TRUNK_IN:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_IN
vesbc(config-esbc-trunk-sip)# remote address 192.168.20.99
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_TRUNK_IN
```

6. Создать SIP-транк в сторону TRUNK_SSW1 и TRUNK_SSW2:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW1
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.115
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# trunk sip TRUNK_SSW2
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.116
vesbc(config-esbc-trunk-sip)# remote port 5060
```

7. Создать транковую группу TG_SSW и добавить туда транки TRUNK_SSW1 и TRUNK_SSW2:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание и переход в настройки транковой группы TG_SSW:
vesbc(config-esbc)# trunk-group TG_SSW

#Добавление в состав транковой группы транков TRUNK_SSW1 и TRUNK_SSW2:
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK_SSW1
vesbc(config-esbc-trunk-group)# trunk 1 TRUNK_SSW2

#Добавление медиаресурсов:
vesbc(config-esbc-trunk-group)# media resource 0 MEDIA_TG_SSW
```

8. Создать таблицу маршрутизации и добавить туда правила, по которым вызовы, приходящие с транка TRUNK_IN, будут маршрутизироваться в транковую группу TG_SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TG_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk-group TG_SSW
```

9. Создать таблицу маршрутизации и добавить туда правила, по которым вызовы, приходящие с TG_SSW, будут маршрутизироваться в транк TRUNK_IN:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TRUNK_IN
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_IN
```

10. Привязать созданные таблицы маршрутизации к транку TRUNK_IN и транковой группе TG_SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# route-table TO_TG_SSW
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# trunk-group TG_SSW
vesbc(config-esbc-trunk-sip)# route-table TO_TRUNK_IN
```

11. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

На шаге 6 при создании транков, в конфигурацию транков не были добавлены медиаресурсы и таблица маршрутизации. Но эти настройки есть в транковой группе TG_SSW, куда включены оба транка. Поэтому при поступлении вызовов с этих транков они будут маршрутизироваться по таблице маршрутизации, которая привязана к TG_SSW, медиаресурсы для согласования и передачи RTP также будут браться из транковой группы TG_SSW.

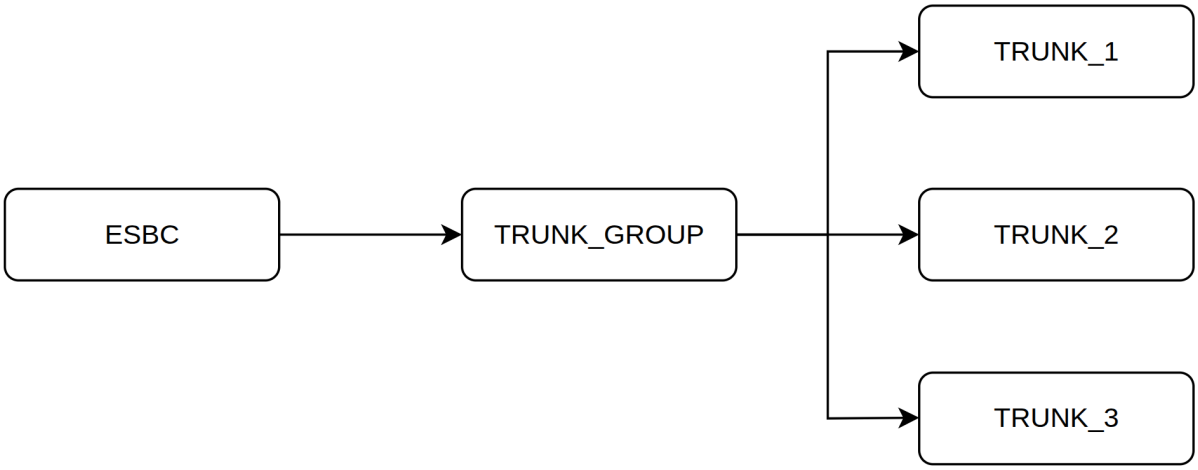
В случае если необходимо, чтобы один из транков, входящих в состав транковой группы, при поступлении на него входящих вызовов маршрутизировался по другой таблице маршрутизации или использовал другие медиаресурсы, нужно добавить соответствующие настройки в данный транк. Настройки транковой группы при этом не меняются, т. к. настройки транка в приоритете.

9.4.1 Логика работы транковой группы для распределения вызовов на транки, входящие в ее состав

1. Распределение вызовов без использования алгоритма балансировки:

Все исходящие вызовы, маршрутизируемые через транковую группу, используют первый транк в группе. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя, вызов будет направлен через следующий транк в группе.

Пример:



На ESBC настроена транковая группа TRUNK_GROUP, в состав которой входят 3 транка (TRUNK_1, TRUNK_2 и TRUNK_3). Приходит вызов и по правилу маршрутизации уходит на эту транковую группу. В результате ESBC совершает попытку вызова в первый транк в составе транковой группы (TRUNK_1), если транк недоступен, то происходит попытка позвонить во второй транк (TRUNK_2). Если попытка вызова также неуспешна, то будет попытка позвонить в последний транк (TRUNK_3). Если попытка также неуспешна, то вызов на первом плече отбивается. Если на каком-то из транков пришел ответ 200OK, то вызов устанавливается.

Все последующие вызовы также будут сначала отправлены в TRUNK_1, и только в случае неудачи будут попытки позвонить в TRUNK_2 и TRUNK_3.

2. Распределение вызовов без использования алгоритма балансировки, но с включенной опцией **pick-once**:

Все исходящие вызовы, маршрутизируемые через транковую группу, используют первый транк в группе. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя, вызов **НЕ** будет направлен через следующий транк в группе, вызов на первом плече сразу отбивается.

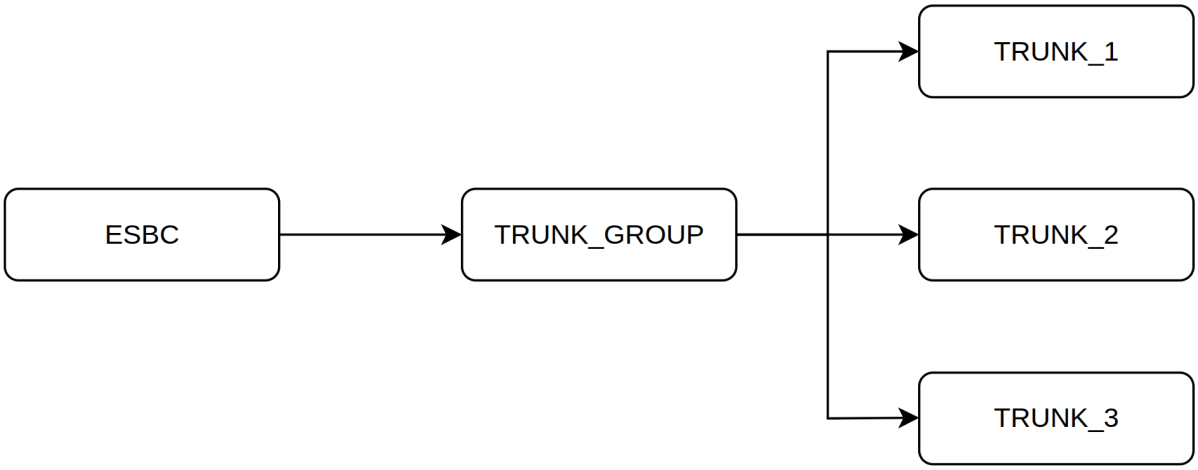
Опцию **pick-once** можно включить в настройках таблицы маршрутизации при выборе действия *direct-to-trunk-group:*

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TG_SSW
vesbc(config-esbc-route-table)# rule 0

#Включение опции pick-once при создании правила маршрутизации на транковую группу TG_SSW:
vesbc(config-esbc-route-table-rule)# action direct-to-trunk-group TG_SSW pick-once
```

3. Распределение вызовов с использованием алгоритма балансировки **round-robin** (опция **pick-once** выключена):

Каждый последующий исходящий вызов, маршрутизируемый через транковую группу, использует следующий транк в группе независимо от результата маршрутизации предыдущего вызова в данную транковую группу. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя, вызов будет направлен через следующий транк в группе.

Пример:



На ESBC настроена транковая группа TRUNK_GROUP, в состав которой входят 3 транка (TRUNK_1, TRUNK_2 и TRUNK_3). Приходит вызов и по правилу маршрутизации уходит на эту транковую группу. В результате ESBC совершает попытку вызова в первый транк в составе транковой группы (TRUNK_1), если вызов неуспешный (транк недоступен или ответ совпал с маской из списка причин отбоя), то происходит попытка позвонить во второй транк (TRUNK_2). Если попытка вызова также неуспешна, то будет попытка позвонить в последний транк (TRUNK_3). Если попытка также неуспешна, то вызов на первом плече отбивается. Если на каком-то из транков пришел ответ 200OK, то вызов устанавливается.

Второй вызов, который смаршрутизировался на данную транковую группу, сначала уйдет на TRUNK_2. Если вызов неуспешный, то ESBC совершит попытку позвонить в TRUNK_3 и потом в TRUNK_1. Если попытки неуспешны, то вызов на первом плече отбивается. По такому же принципу третий вызов сначала распределится в TRUNK_3, четвертый вызов — в TRUNK_1 и т. д.

Опция балансировки **round-robin** включается в настройках транковой группы:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание и переход в настройки транковой группы TRUNK_GROUP:
vesbc(config-esbc)# trunk-group TRUNK_GROUP

#Добавление в состав транковой группы транков TRUNK_1, TRUNK_2 и TRUNK_3:
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK_1
vesbc(config-esbc-trunk-group)# trunk 1 TRUNK_2
vesbc(config-esbc-trunk-group)# trunk 2 TRUNK_3

#Активация режима балансировки round-robin на траковой группе:
vesbc(config-esbc-trunk-group)# balancing round-robin
```

4. Распределение вызовов с использованием алгоритма балансировки **round-robin** (опция **pick-once** включена):

Каждый последующий исходящий вызов, маршрутизируемый через транковую группу, использует следующий транк в группе. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя вызов **НЕ** будет направлен через следующий транк в группе, вызов на первом плече сразу отбивается.

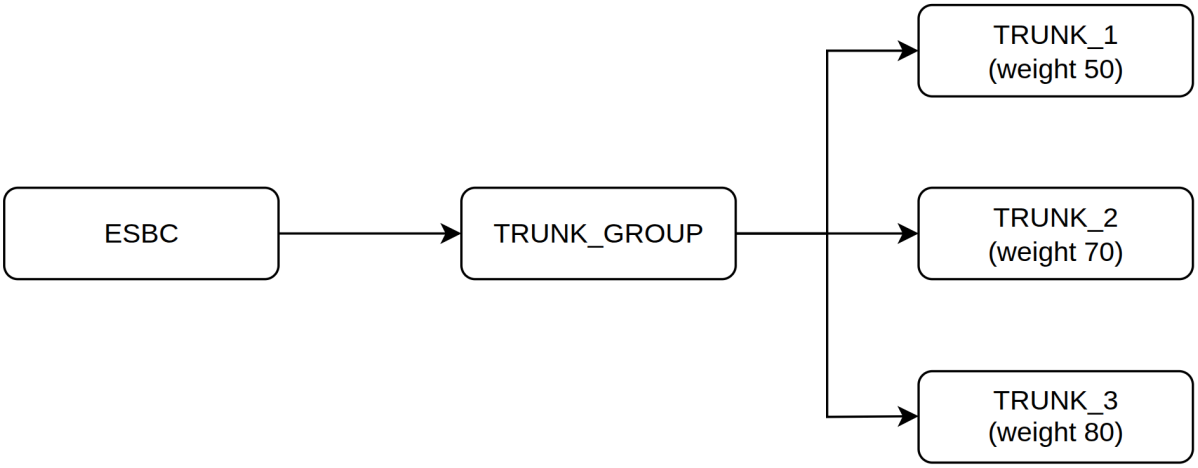
Пример:

В схеме из п. 3 первый вызов распределяется в TRUNK_1, если он отбивается, то первое плечо вызова сразу отбивается, попыток позвонить в TRUNK_2, TRUNK_3 нет. Второй вызов распределяется в TRUNK_2, третий — в TRUNK_3, четвертый — в TRUNK_1 и т. д.

5. Распределение вызовов с использованием алгоритма балансировки **weighted round-robin** (опция **pick-once** выключена):

Каждые последующие исходящие вызовы, маршрутизируемые через транковую группу, распределяются согласно весам, присвоенным транкам в этой группе. То есть чем больше вес транка, относительно весов других транков, тем больше запросов через него отправится. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя, вызов будет направлен через следующий транк в очереди.

Пример:



На ESBC настроена транковая группа TRUNK_GROUP, в состав которой входят 3 транка (TRUNK_1, TRUNK_2 и TRUNK_3) с весами 50, 70 и 80 соответственно. Приходит вызов и по правилу маршрутизации уходит на эту транковую группу. В результате ESBC будет распределять вызовы согласно выставленным весам. Например, из 100 вызовов, направленных через транковую группу в транк TRUNK_1 с весом 50 направится 25 вызовов, в транк TRUNK_2 направится 35 вызовов и в TRUNK_3 направится 40 вызовов.

Веса транков и опция балансировки **weighted round-robin** назначаются в настройках транковой группы:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание и переход в настройки транковой группы TRUNK_GROUP:
vesbc(config-esbc)# trunk-group TRUNK_GROUP

#Добавление в состав транковой группы транков TRUNK_1 с весом 50, TRUNK_2 с весом 70 и TRUNK_3
с весом 80:
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK_1 weight 50
vesbc(config-esbc-trunk-group)# trunk 1 TRUNK_2 weight 70
vesbc(config-esbc-trunk-group)# trunk 2 TRUNK_3 weight 80

#Активация режима балансировки weighted round-robin на траковой группе:
vesbc(config-esbc-trunk-group)# balancing weighted-round-robin
```

6. Распределение вызовов с использованием алгоритма балансировки **weighted round-robin** (опция **pick-once** включена):

Каждые последующие исходящие вызовы, маршрутизируемые через транковую группу, распределяются согласно настроенным весам на транках в этой группе. В случае недоступности транка или при совпадении ответа с маской из списка причин отбоя вызов **НЕ** будет направлен через следующий транк в очереди, вызов на первом плече сразу отбивается.

9.5 Настройка SIP-транспортов

SIP-транспорт представляет точку входа/выхода сигнализации, т. е. это IP-адрес и порт, с которого ESBC будет отправлять и на который будет принимать сигнальные сообщения.

> Возможно использование IP-адреса, полученного по DHCP.

Пример:

Требуется, чтобы ESBC для передачи/приема сигнальных сообщений на встречную сторону использовал IP-адрес 192.168.16.113, порт 5065.

Решение:

Перейти к настройкам модуля управления конфигурацией ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
```

Создать и настроить соответствующим образом SIP-транспорт:

```
#Создание/переход в настройки SIP-транспорта NEW_TRANSPORT:
vesbc(config-esbc)# sip transport NEW_TRANSPORT

#Назначить IP-адрес 192.168.16.113 для использования SIP-транспортом:
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113

#Назначить порт 5065 для использования SIP-транспортом:
vesbc(config-esbc-sip-transport)# port 5065

#Выбрать протокол транспортного уровня, используемый для приема/передачи сообщений SIP:
vesbc(config-esbc-sip-transport)# mode udp-prefer
```

После привязки созданного SIP-транспорта к какому-либо направлению (транку или абонентскому интерфейсу) он будет использоваться для передачи/получения сигнальных SIP-сообщений на выбранных направлениях.

Поддержано несколько режимов работы с протоколами транспортного уровня, конфигурируется командой *mode* из примера выше. Режимы работы следующие:

- *tcp-only* — использовать только TCP-протокол;
- *tcp-prefer* — прием по UDP и TCP. Отправка по TCP. В случае если не удалось установить соединение по TCP, отправка производится по UDP;
- *tls* — использовать tls;
- *udp-only* — использовать только UDP-протокол;
- *udp-prefer* — прием по UDP и TCP. Отправка пакетов более 1300 байт по TCP, менее 1300 байт — по UDP;
- *ws* — использовать WebSocket;
- *wss* — использовать WebSocket Secure.

При использовании типа транспорта tls или wss возможно использование пользовательских сертификатов. Подробнее о пользовательских сертификатах см. в разделе Настройка криптопрофилей.

> Пример настройки SIP-абонентов, использующих WebRTC есть в разделе Примеры настройки ESBC.

9.6 Настройка медиаресурсов

Медиаресурсы представляют собой диапазоны UDP-портов и IP-адресов, используемых ESBC для передачи/получения потоков RTP.

> Возможно использование IP-адреса, полученного по DHCP.

При указании диапазона портов для RTP, необходимо чтобы начальный порт был **четным** а конечный порт **нечетным**.

Минимальный начальный номер порта — 1024, минимальный конечный номер порта — 1027, максимальный конечный номер порта — 65535.

Т. о. минимальный диапазон портов не может быть менее четырех.

По умолчанию используется диапазон портов с 8000 по 65535.

Пример:

Требуется, чтобы ESBC для передачи медиатрафика использовал IP-адрес 192.168.16.113 и порты с 20000 до 29999.

Решение:

Перейти к настройкам модуля управления конфигурацией ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
```

Создать и настроить соответствующим образом медиаресурс:

```
#Создание/переход в настройки медиаресурса MEDIA_1:
vesbc(config-esbc)# media resource MEDIA_1

#Назначить IP-адрес 192.168.16.113 для использования в медиаресурсах:
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Настроить диапазон UDP-портов с 20000 до 29999 для использовании в медиаресурсах:
vesbc(config-esbc-media-resource)# port-range 20000-29999
```

После привязки созданного медиаресурса к какому-либо направлению (транку, транковой группе или абонентскому интерфейсу), он будет использоваться для передачи/получения потоков RTP на выбранных направлениях.

При использовании одинакового IP-адреса для разных медиаресурсов не допускается пересечение диапазонов портов между этими ресурсами.

9.7 Использование тегов для маршрутизации и модификации SIP-сообщений

Механизм **тегирования** предназначен для оптимизации и упрощения таблиц маршрутизации и увеличения гибкости модификации SIP-сообщений. Вместо создания множества отдельных правил для каждого телефонного префикса, маршрутизация строится на основе текстовых меток (тегов), которые динамически присваиваются вызову на этапе анализа SIP-сообщения.

Использование тегов позволяет:

- Сократить объем таблицы маршрутизации до нескольких универсальных правил;
- Вынести обслуживание сотен и тысяч префиксов (например, кодов мобильных операторов или направлений МГ/МН) в префикс-группы;
- Применять модификации SIP-сообщений с учетом тега, что значительно повышает гибкость модификаторов.

Для добавления и удаления тегов используются **Общие модификаторы (common)**.

> ⚠ Тег является виртуальным параметром, который используется только логикой взаимодействия объектов ESBC (таблицы маршрутизации, таблицы модификации) и не добавляется в сообщения, передаваемые через ESBC.

Примеры использовании тегов находятся в разделах Настройка таблиц маршрутизации (см. пункт ниже) и Настройка модификаторов.

9.8 Настройка таблиц маршрутизации

Схема осуществления маршрутизации SIP-сообщения:

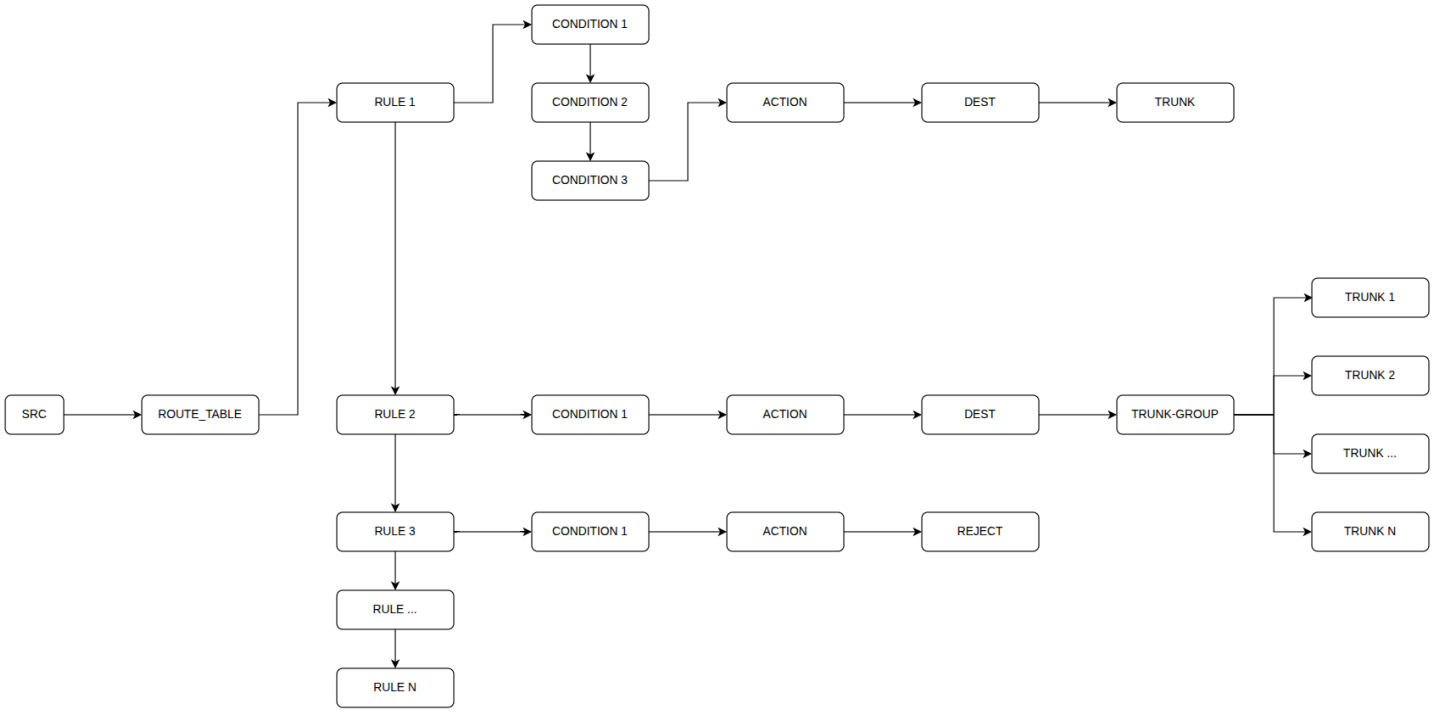


Таблица маршрутизации представляет собой набор правил и действий, по которым обрабатывается входящий вызов и указывается исходящий транк (или транк-группа) или переход на другую таблицу маршрутизации для формирования исходящего вызова.

Таблицы маршрутизации применяются к входящим вызовам и могут быть настроены для транков, транк-групп и абонентских интерфейсов.

Таблица состоит из правил (RULE), правило обязательно должно содержать действие (ACTION), и, опционально, условия (CONDITION), которые должны быть соблюдены для выполнения данного действия маршрутизации. Если условия отсутствуют, действия совершаются безусловно.
Действие — это операция, результатом которой будет являться конкретное направление или переход на другую таблицу маршрутизации.
В текущей версии в качестве направлений выступают транки и транк-группы.

Правила маршрутизации выбираются по порядку до тех пор, пока второе плечо не будет успешно согласовано, или не будет рассмотрено последнее правило. Если рассматривать на примере вызова, то роутинг будет выполняться до тех пор, пока второе плечо не примет вызов.

Также порядок выбора правила для исходящих вызовов может составляться исходя из алгоритмов балансировки round robin и weighted round robin.

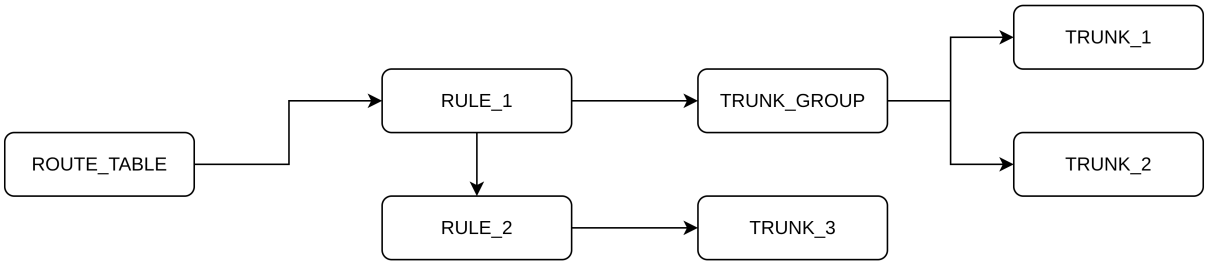
В случае маршрутизации на транк-группу действует тот же алгоритм. Т. е. проход осуществляется по всем транкам выбранной группы по порядку до тех пор, пока сессия не согласуется, или не будет выбран последний транк. Если после прохождения по всем транкам выбранной группы не удалось согласовать второе плечо, продолжается выбор оставшихся правил из таблицы маршрутизации.

В общем, этот алгоритм можно описать так: **проход по всем направлениям, всех правил маршрутизации, пока сессия не будет согласована, или не будет рассмотрено последнее правило.**
Исключением является правило **Reject** — отбой входящей сессии. Это правило завершает проход по таблице маршрутизации. Для него можно выбрать SIP-код отбоя, которым будет отбиваться входящее сообщение.

Выбор следующего направления будет происходить:

- при внутренних сбоях, до согласования сессии;
- при отбое с встречной стороны, кроме 3xx кодов SIP.

Пример перебора правил:



В таблице маршрутизации два правила, первое направляет вызов в TRUNK_GROUP, второе направляет вызов в TRUNK_3, условия нигде не настроены. Приходит вызов и начинает маршрутизироваться по данной таблице маршрутизации. В результате вызов уходит на TRUNK_GROUP и оттуда в TRUNK_1, в случае если вызов через TRUNK_1 не установился (например, транк недоступен), то маршрутизация продолжает выполняться, вызов отправляется в TRUNK_2. Если попытка вызова в TRUNK_2 также завершилась неудачно, ESBC переходит к RULE_2 и отправляет вызов в TRUNK_3. Если и здесь попытка установить вызов также оказалась неуспешной, то первое плечо отбивается, и вызов завершается, т. к. больше правил в таблице маршрутизации нет. Если попытка установить вызов успешна, то вызов устанавливается.

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table ROUTE_TABLE

#Добавление первого правила с действием отправить вызов в транковую группу TRUNK_GROUP:
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk-group TRUNK_GROUP
vesbc(config-esbc-route-table-rule)# exit

#Добавление второго правила с действием отправить вызов в транк TRUNK_3:
vesbc(config-esbc-route-table)# rule 2
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_3
```

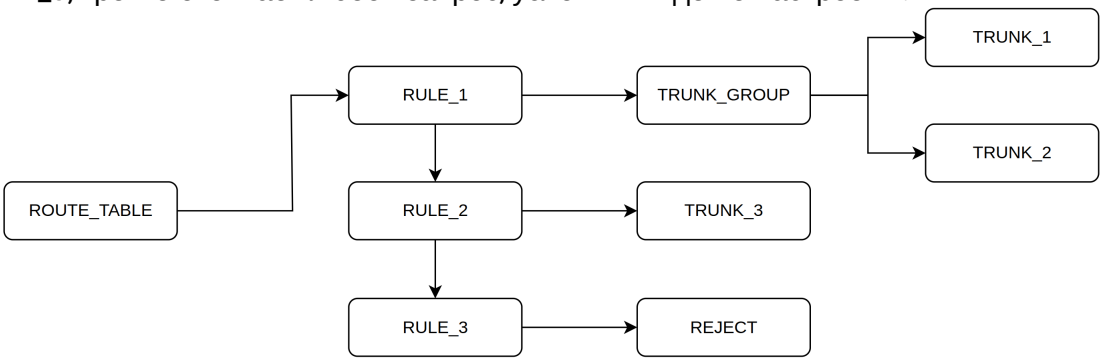
9.8.1 Балансировка правил таблицы маршрутизации

Пример балансировки правил по алгоритму round robin

При настроенной балансировке правил таблицы маршрутизации по алгоритму round robin каждый последующий вызов будет направляться на следующее правило в таблице маршрутизации.

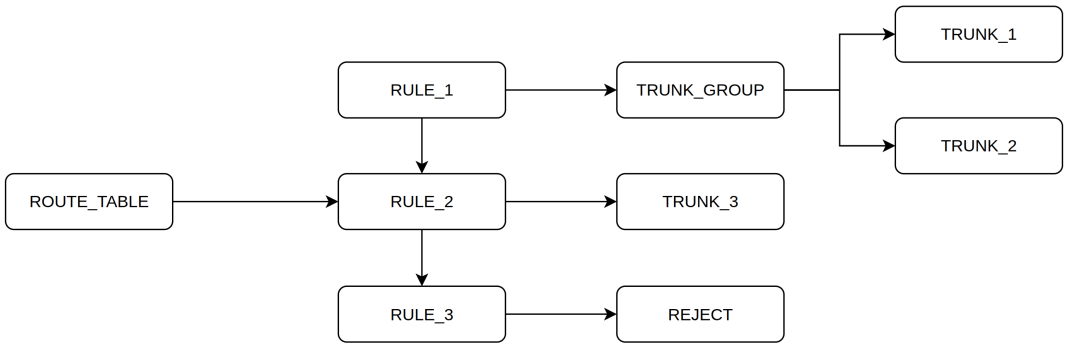
Схема:

В таблице маршрутизации три правила, первое направляет вызов в TRUNK_GROUP, второе направляет вызов в TRUNK_3, третье отбивает любой запрос, условия нигде не настроены.

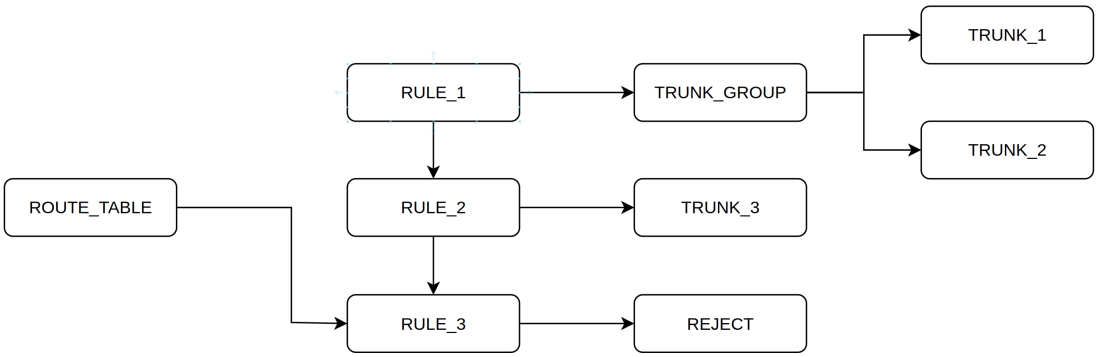


Приходит вызов и начинает маршрутизироваться по данной таблице маршрутизации. В результате вызов по правилу RULE_1 уходит на TRUNK_GROUP и оттуда в TRUNK_1, в случае если вызов через TRUNK_1 не установился (например, транк недоступен), то маршрутизация продолжает выполняться, вызов отправляется в TRUNK_2. Если попытка вызова в TRUNK_2 также завершилась неудачно, ESBC переходит к RULE_2 и отправляет вызов в TRUNK_3. Если и здесь попытка установить вызов также оказалась неуспешной, то ESBC переходит к RULE_3 и первое плечо отбивается, вызов завершается, т. к. установлено правило Reject.

Следующий вызов будет направлен по правилу RULE_2 в транк TRUNK_3. Если попытка вызова завершилась неудачно, то ESBC переходит к RULE_3 и первое плечо отбивается, вызов завершается, т. к. установлено правило Reject.



Третий вызов будет направлен по правилу RULE_3 и будет отбит, т.к. установлено правило Reject.



Т.к. в таблице маршрутизации больше нет правил, то следующий вызов направится по RULE_1 в транковую группу TRUNK_GROUP и алгоритм повторится. Таким образом, каждый следующий вызов будет маршрутизироваться по следующему правилу таблицы маршрутизации по кругу.

Опция балансировки **round-robin** включается в настройках таблицы маршрутизации:

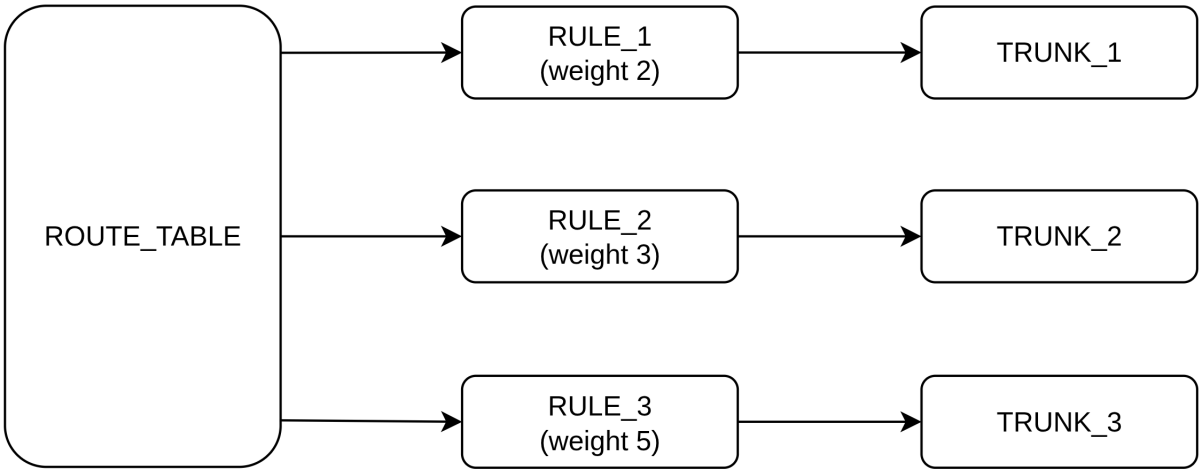
```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table ROUTE_TABLE

#Добавление первого правила с действием отправить вызов в транковую группу TRUNK_GROUP:
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk-group TRUNK_GROUP
vesbc(config-esbc-route-table-rule)# exit

#Добавление второго правила с действием отправить вызов в транк TRUNK_3:
vesbc(config-esbc-route-table)# rule 2
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_3

#Добавление второго правила с действием отбить входящий вызов:
vesbc(config-esbc-route-table)# rule 3
vesbc(config-esbc-route-table-rule)# action reject

#Активация режима балансировки правил таблицы маршрутизации по алгоритму round robin:
vesbc(config-esbc-route-table)# balancing round-robin
vesbc(config-esbc-route-table)#
```

Пример балансировки правил по алгоритму weighted round robin

Каждые последующие исходящие вызовы, маршрутизируемые по таблице маршрутизации с включённой балансировкой **weighted round robin**, распределяются согласно весам, присвоенным правилам в этой таблице. То есть чем больше вес правила, относительно весов других правил, тем

больше запросов будет маршрутизироваться по нему. В случае недоступности направления вызов будет смаршрутизирован по следующему правилу в очереди.

Схема:



На ESBC настроена таблица маршрутизации ROUTE_TABLE в состав которой входят правила (RULE_1, RULE_2 и RULE_3) с весами 2, 3 и 5 соответственно. ESBC будет применять правила маршрутизации к запросам, согласно выставленным весам. Например из 100 запросов маршрутизируемых по этой таблице правило RULE_1 с весом 2 применится к 20 вызовам, правило RULE_2 с весом 3 применится к 30 вызовам и правило RULE_3 с весом 5 применится к 50 вызовам.

Веса правил и опция балансировки **weighted round robin** назначаются в настройках таблицы маршрутизации:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание и переход в настройки таблицы маршрутизации ROUTE_TABLE:
vesbc(config-esbc)# route-table ROUTE_TABLE

#Создание правила с маршрутизацией в транк TRUNK_1 и весом 2:
vesbc(config-esbc-route-table)# rule 1 weight 2
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# exit

#Создание правила с маршрутизацией в транк TRUNK_2 и весом 3:
vesbc(config-esbc-route-table)# rule 2 weight 3
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_2
vesbc(config-esbc-route-table-rule)# exit

#Создание правила с маршрутизацией в транк TRUNK_3 и весом 5:
vesbc(config-esbc-route-table)# rule 3 weight 5
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_2
vesbc(config-esbc-route-table-rule)# exit

#Активация режима балансировки правил таблицы маршрутизации по алгоритму weighted round robin:
vesbc(config-esbc-route-table)# balancing weighted-round-robin
```

9.8.2 Код отбоя для правила Reject

Для правила Reject можно выбрать код SIP-ответа, которым будут отбиваться входящие сообщения. По умолчанию установлен код 403.

⚠ В текущей версии ПО поддержано изменение кода только для сигнализации SIP.

Пример:

Устройства Cisco могут отправлять регистрации для контроля доступности направления. Чтобы направление считалось доступным на эти запросы должен приходить ответ 200 OK. Для того чтобы отвечать на такие регистрации, можно использовать правило отбоя с кодом 200 и условие в таблице маршрутизации для Cisco keepalive REGISTER.

```
vesbc# configure
vesbc(config)# esbc

#Создание таблицы маршрутизации ROUTE_TABLE и добавление правила для отбоя запросов с номеров,
которые начинаются с буквы:
vesbc(config-esbc)# route-table ROUTE_TABLE
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action reject sip 200
vesbc(config-esbc-route-table-rule)# condition 0 sip-message ";cisco-keep-alive"
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
```

Запросы, в которых есть строка ";cisco-keep-alive", будут отбиваться ответом 200 OK. Таким образом, устройство Cisco будет считать, что направление доступно и не переключится на резервное направление.

⚠ При изменении кода отбоя меняется только Status-Line сообщения, остальные заголовки и тело сообщения не изменяются. Поэтому при выборе кода 200 или 401 не будет создан диалог или отправлен запрос аутентификации.

9.8.3 Смена таблиц маршрутизации

В конфигурации таблицы маршрутизации реализована команда *switch-route-table <route-table-name>* для продолжения маршрутизации SIP-запроса через другую таблицу маршрутизации.

Использование каскадного принципа таблиц маршрутизации позволяет гибко настраивать маршрутизацию между направлениями.

Примеры использования каскадных таблиц маршрутизации

Допускается использовать каскадный переход из нескольких таблиц маршрутизации, например:

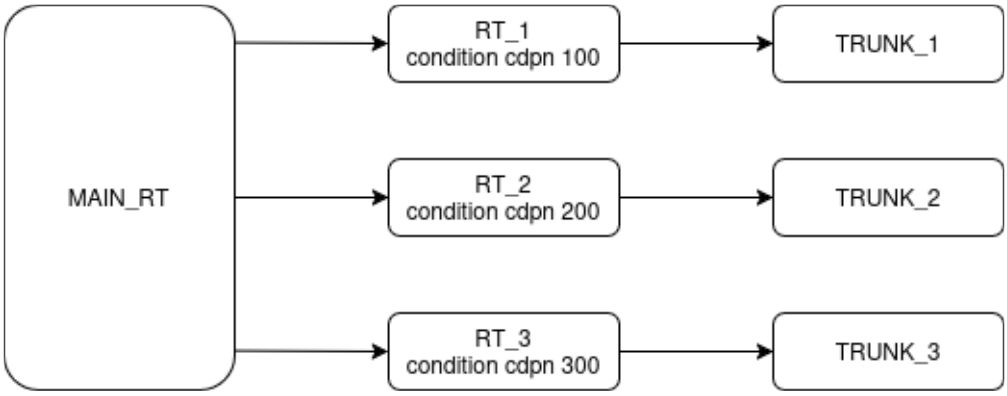
```
vesbc# configure
vesbc(config)# esbc

#Создание таблицы маршрутизации ROUTE_TABLE_1 и добавление правила с действием сменить таблицу
маршрутизации на ROUTE_TABLE_2:
vesbc(config-esbc)# route-table ROUTE_TABLE_1
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action switch-route-table ROUTE_TABLE_2
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit

#Создание таблицы маршрутизации ROUTE_TABLE_2 и добавление правила с действием сменить таблицу
маршрутизации на ROUTE_TABLE_3:
vesbc(config-esbc)# route-table ROUTE_TABLE_2
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action switch-route-table ROUTE_TABLE_3
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit

#Создание таблицы маршрутизации ROUTE_TABLE_3 и добавление правила с действием сменить таблицу
маршрутизации на ROUTE_TABLE_4:
vesbc(config-esbc)# route-table ROUTE_TABLE_3
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action switch-route-table ROUTE_TABLE_4
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit

#Создание таблицы маршрутизации ROUTE_TABLE_4 и добавление правила с действием отправить вызов
в транк TRUNK_1:
vesbc(config-esbc)# route-table ROUTE_TABLE_4
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
```

Запрещено указывать в правиле action switch-route-table название той таблицы, в которой указывается данное правило.
Пример:

```
route-table MAIN
  rule 5
     action switch-route-table MAIN
  exit
exit
```

Маршрутизация на разные направления через общую таблицу маршрутизации

Для всех транков (например, TRUNK_1, TRUNK_2, TRUNK_3) указать одну таблицу маршрутизации (например, MAIN). Все правила и условия для маршрутизации в конкретный транк указаны в отдельной таблице маршрутизации для этого транка (таблицы маршрутизации RT_1, RT_2, RT_3). При добавлении транков в конфигурации необходимо создать соответствующую таблицу маршрутизации и добавить новое правило в таблицу MAIN.



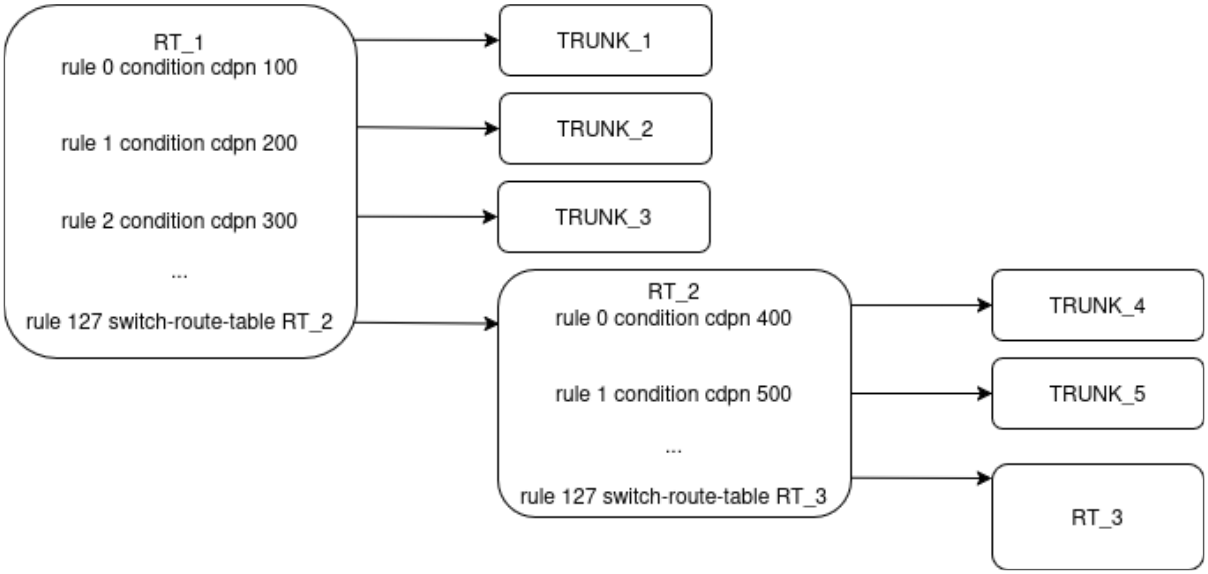
Пример конфигурации:

```
route-table MAIN
   rule 0
     action switch-route-table RT_1
   exit
   rule 1
     action switch-route-table RT_2
   exit
   rule 2
     action switch-route-table RT_3
   exit
 exit
 route-table RT_1
   rule 0
     action direct-to-trunk TRUNK_1
     condition 0 cdpn pattern "100"
   exit
 exit
 route-table RT_2
   rule 0
     action direct-to-trunk TRUNK_2
     condition 0 cdpn pattern "200"
   exit
 exit
 route-table RT_3
   rule 0
     action direct-to-trunk TRUNK_3
     condition 0 cdpn pattern "300"
   exit
 exit
   trunk sip TRUNK_1
   route-table MAIN
   ...
 exit
   trunk sip TRUNK_2
   route-table MAIN
   ...
 exit
   trunk sip TRUNK_3
   route-table MAIN
   ...
 exit
```

В таком случае маршрутизация любого вызова из любого направления будет осуществляться по всем правилам всех таблиц маршрутизации до первого совпадения CdPN. Если совпадение не будет найдено, то вызов завершится кодом SIP 404 "Not Found".

Маршрутизация на разные направления через последовательный каскад таблиц маршрутизации

Правила для маршрутизации добавляются в одну таблицу маршрутизации, при достижении ограничения количества правил (128) в качестве последнего правила добавляется переход на следующую таблицу маршрутизации.



Пример конфигурации:

```
route-table RT_1
    rule 0
      action direct-to-trunk TRUNK_1
      condition 0 cdpn pattern "100"
    exit
    rule 1
      action direct-to-trunk TRUNK_2
      condition 0 cdpn pattern "200"
    exit
    rule 2
      action direct-to-trunk TRUNK_3
      condition 0 cdpn pattern "300"
    exit
    ...
     rule 127
      action switch-route-table RT_2
    exit
  exit
  route-table RT_2
    rule 0
      action direct-to-trunk TRUNK_4
      condition 0 cdpn pattern "400"
    exit
    rule 1
      action direct-to-trunk TRUNK_5
      condition 0 cdpn pattern "500"
    exit
    ...
    rule 127
      action switch-route-table RT_3
    exit
  exit
  route-table RT_3
    ...
  exit
 ...
    trunk sip TRUNK_1
    route-table RT_1
    ...
  exit
    trunk sip TRUNK_2
    route-table RT_1
    ...
  exit
    trunk sip TRUNK_3
    route-table RT_1
    ...
  exit
    trunk sip TRUNK_4
    route-table RT_1
    ...
  exit
    trunk sip TRUNK_5
    route-table RT_1
    ...
  exit
```

Обработка вызова также как и в предыдущем примере будет осуществляться последовательно по всем правилам всех таблиц маршрутизации. Если совпадение не будет найдено, то вызов завершится кодом SIP 404 "Not Found".

Логика обработки SIP-запроса при использовании каскада таблиц маршрутизации

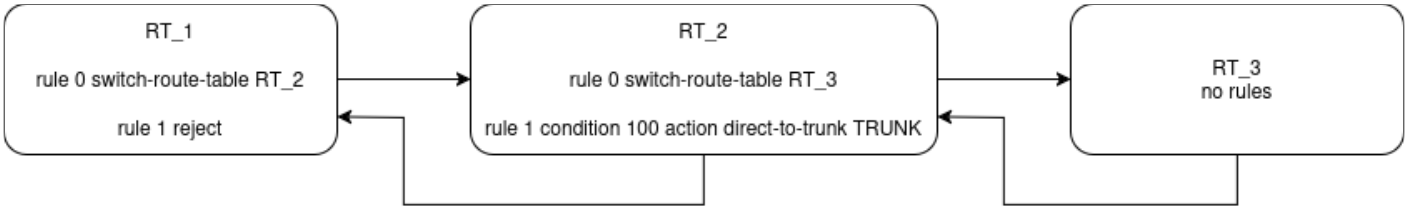
Осуществляется последовательный проход по всем направлениям, каждого правила каждой таблицы маршрутизации до тех пор, пока сессия не будет согласована, или не будет рассмотрено последнее правило в каждой таблице маршрутизации.

При проверке правил ESBC сохраняет информацию о всех пройденных объектах (транки, транк-группы, таблицы маршрутизации) с целью исключения петли маршрутизации.

Если используется маршрутизация через каскад таблиц маршрутизации, и нет подходящего правила для передачи SIP-запроса на плечо Б, то осуществляется возврат на предыдущую таблицу маршрутизации к следующему правилу.

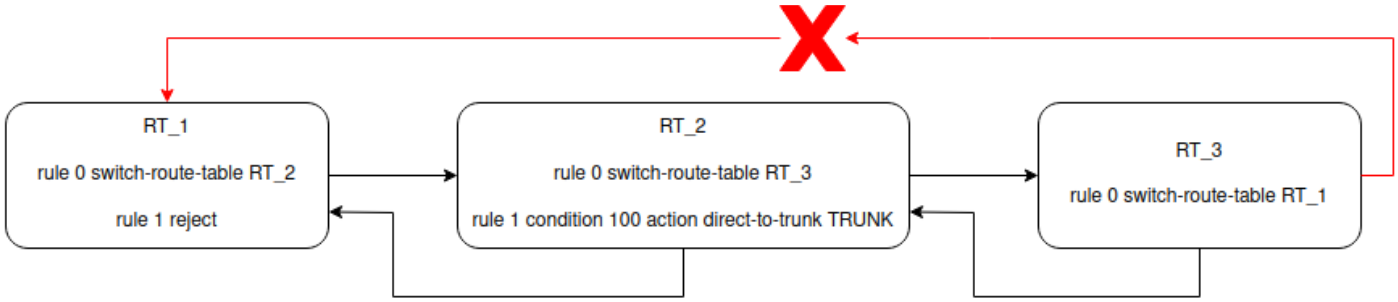
Пример 1:

Настроена маршрутизация через три таблицы. Осуществляется вызов на номер 200. При осуществлении маршрутизации, в соответствии с правилом rule 0 таблицы RT_1 происходит переход к таблице RT_2. В таблице RT_2 также первым правилом указан переход на таблицу RT_3. В таблице RT_3 нет правил маршрутизации, поэтому осуществляется переход к предыдущей таблице (RT_2) к следующему по порядку правилу (rule1). Т. к. это правило не подходит для направления вызова в транк TRUNK (CdPN 200 !== CdPN 100), то осуществляется переход к предыдущей таблице (RT_1) к следующему по порядку правилу (rule1). Последнее правило в этой таблице — reject, вызов будет завершен кодом SIP 403 "Forbidden".



Пример 2:

Сценарий аналогичный примеру 1, за исключением того, что в таблице RT_3 настроено правило перехода к таблице RT_1. В данном случае логика обработки запроса будет аналогична примеру выше, т. к. при маршрутизации RT_1 уже была пройдена и повторный возврат к правилу rule 0 этой таблицы привел бы к зацикливанию вызова.



9.8.4 Настройка "закрепления" регистрации

В конфигурации таблицы маршрутизации реализована команда **refresh-registration** для закрепления регистрации на одном направлении. Используется для обновления активной регистрации абонента на том транке, на котором он был зарегистрирован ранее. Если пользователь ранее не регистрировался, правило пропускается.

Пример:

Нужно настроить конфигурацию так, чтобы абоненты регистрировались на основном транке, а при его недоступности регистрации распределялись по резервным транкам.

Схема:

В конфигурации ESBC настроено 4 транка: MAIN_TRUNK, TRUNK_1, TRUNK_2 и TRUNK_3. Транки TRUNK_1, TRUNK_2 и TRUNK_3 находятся в транковой группе, на которой настроена балансировка round-robin, поэтому все запросы регистрации, приходящие на эту группу, будут распределяться по транкам согласно балансировке. В таком случае, чтобы закрепить регистрацию абонента на одном транке, нужно добавить правило **refresh-registration** в таблицу маршрутизации. При прохождении этого правила первой регистрации абонента оно пропускается, и запрос направится в транк согласно следующим правилам маршрутизации. Но если у абонента уже есть регистрация и приходит её обновление, то оно направится в тот транк, на котором он уже зарегистрирован.

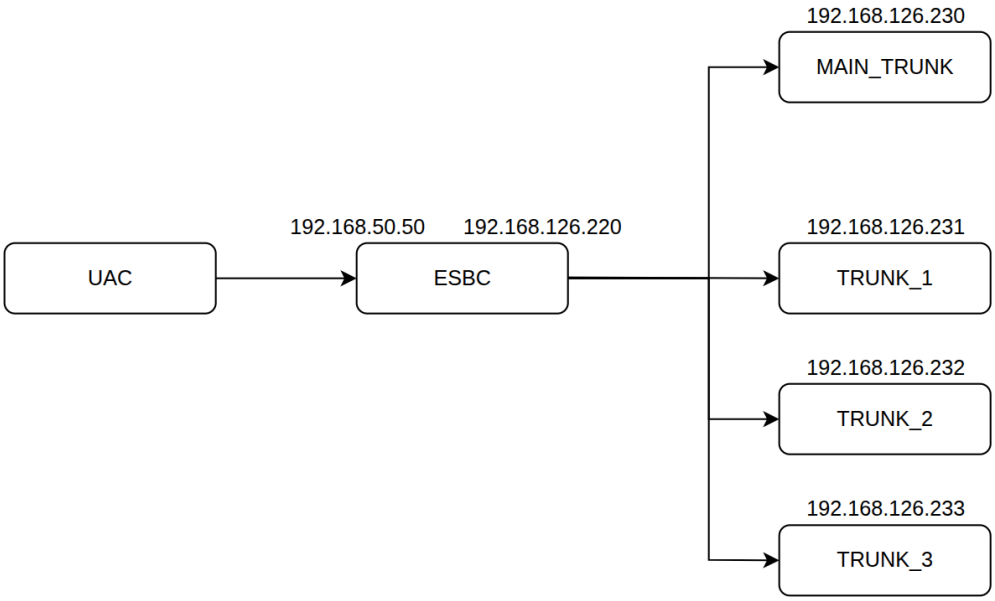
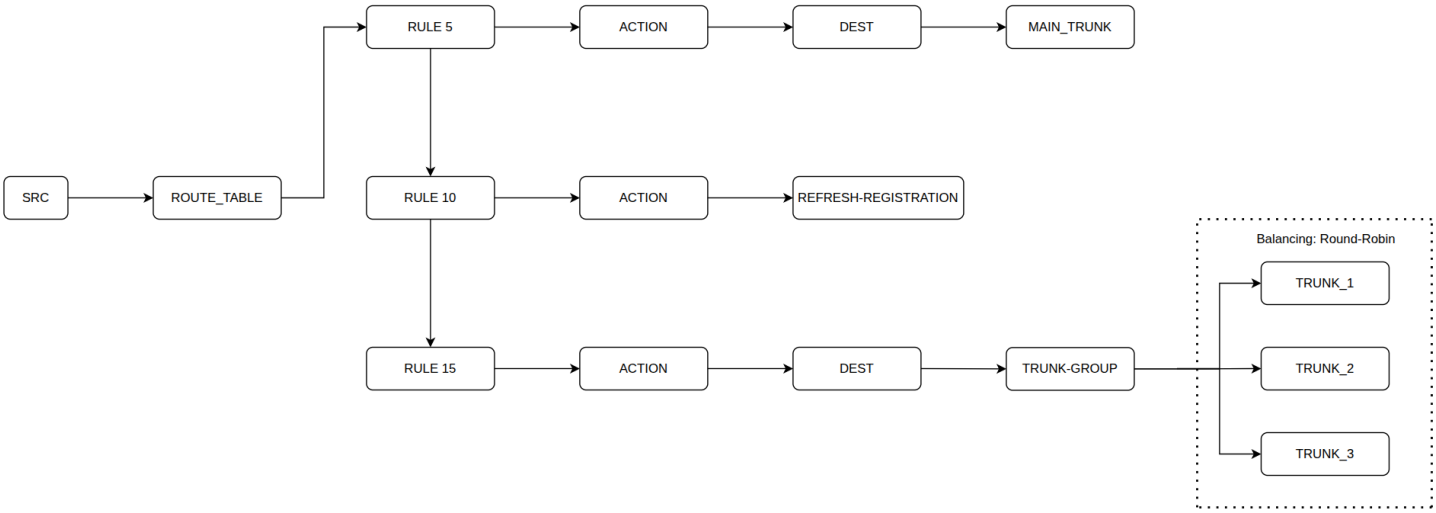


Схема таблицы маршрутизации:

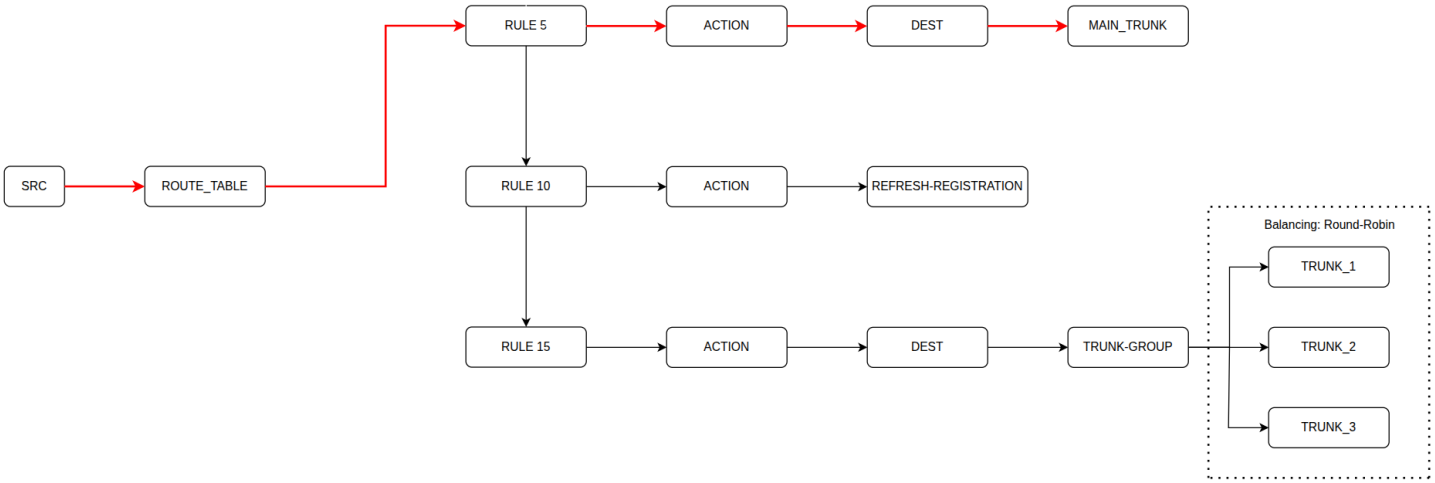


Пример конфигурации приведен ниже.

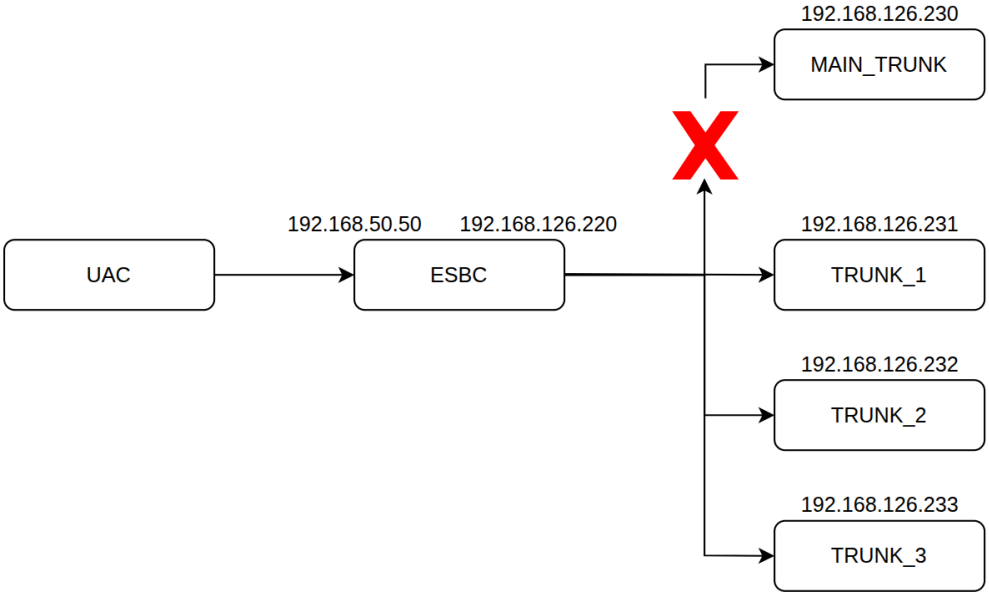
```
vesbc# config
vesbc(config)# esbc

#Создание медиаресурса для транков:
vesbc(config-esbc)# media resource MEDIA
vesbc(config-esbc-media-resource)# ip address 192.168.126.220
vesbc(config-esbc-media-resource)# port-range 5000-10001
vesbc(config-esbc-media-resource)# exit

#Создание транспорта для транка MAIN_TRUNK:
vesbc(config-esbc)# sip transport TRANSPORT_MAIN_TRUNK
vesbc(config-esbc-sip-transport)# ip address 192.168.126.220
vesbc(config-esbc-sip-transport)# port 5060
vesbc(config-esbc-sip-transport)# exit

#Создание транспорта для транка TRUNK_1:
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-sip-transport)# ip address 192.168.126.220
vesbc(config-esbc-sip-transport)# port 5070
vesbc(config-esbc-sip-transport)# exit

#Создание транспорта для транка TRUNK_2:
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_2
vesbc(config-esbc-sip-transport)# ip address 192.168.126.220
vesbc(config-esbc-sip-transport)# port 5080
vesbc(config-esbc-sip-transport)# exit

#Создание транспорта для транка TRUNK_3:
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_3
vesbc(config-esbc-sip-transport)# ip address 192.168.126.220
vesbc(config-esbc-sip-transport)# port 5090
vesbc(config-esbc-sip-transport)# exit

#Создание транка MAIN_TRUNK:
vesbc(config-esbc)# trunk sip MAIN_TRUNK
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_MAIN_TRUNK
vesbc(config-esbc-trunk-sip)# remote address 192.168.126.230
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# exit

#Создание транка TRUNK_1:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-trunk-sip)# remote address 192.168.126.231
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# exit

#Создание транка TRUNK_2:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_2
vesbc(config-esbc-trunk-sip)# remote address 192.168.126.232
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# exit
```

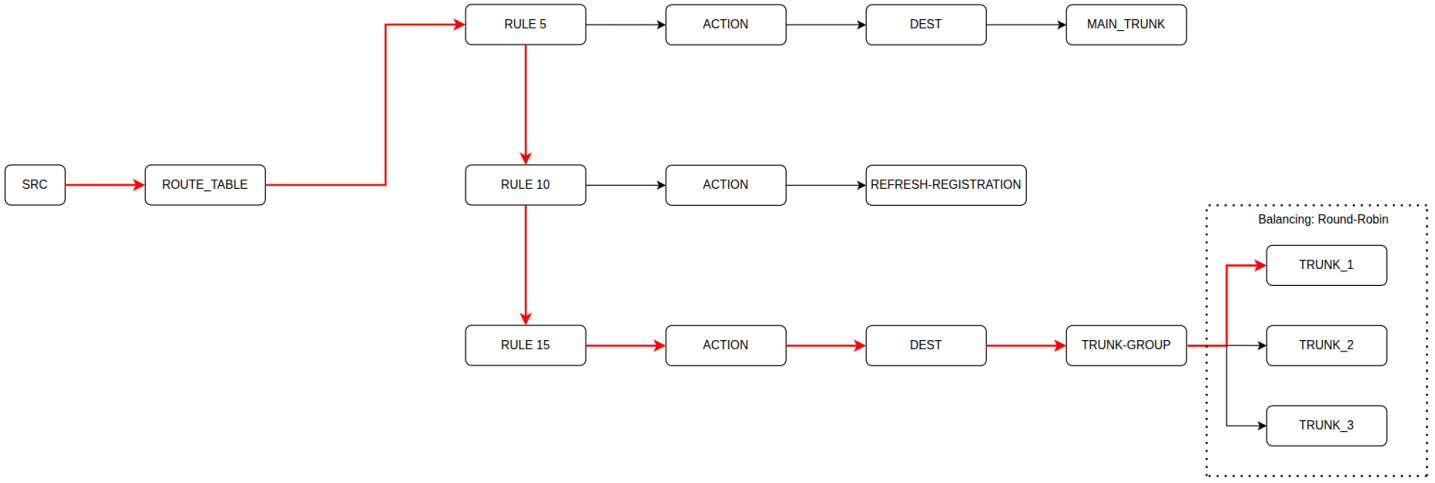
```
#Создание транка TRUNK_3:
vesbc(config-esbc)# trunk sip TRUNK_3
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_3
vesbc(config-esbc-trunk-sip)# remote address 192.168.126.233
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# exit

#Создание транковой группы TRUNK_GROUP с транками TRUNK_1, TRUNK_2 и TRUNK_3 с балансировкой
 round-robin:
vesbc(config-esbc)# trunk-group TRUNK_GROUP
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK_1
vesbc(config-esbc-trunk-group)# trunk 1 TRUNK_2
vesbc(config-esbc-trunk-group)# trunk 2 TRUNK_3
vesbc(config-esbc-trunk-group)# balancing round-robin
vesbc(config-esbc-trunk-group)# exit

#Создание таблицы маршрутизации ROUTE_TABLE и добавление в неё правил:
vesbc(config-esbc)# route-table ROUTE_TABLE
vesbc(config-esbc-route-table)# rule 5
vesbc(config-esbc-route-table-rule)# action direct-to-trunk MAIN_TRUNK
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# rule 10
vesbc(config-esbc-route-table-rule)# action refresh-registration
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# rule 15
vesbc(config-esbc-route-table-rule)# action direct-to-trunk-group TRUNK_GROUP
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit

#Создание медиаресурса MEDIA_USER для абонентского интерфейса:
vesbc(config-esbc)# media resource MEDIA_USER
vesbc(config-esbc-media-resource)# ip address 192.168.50.50
vesbc(config-esbc-media-resource)# port-range 20000-30001
vesbc(config-esbc-media-resource)# exit

#Создание транспорта для абонентского интерфейса:
vesbc(config-esbc)# sip transport TRANSPORT_USER
vesbc(config-esbc-sip-transport)# ip address 192.168.50.50
vesbc(config-esbc-sip-transport)# port 5060
vesbc(config-esbc-sip-transport)# exit

#Создание транспорта абонентского интерфейса USER_INTERFACE с таблицей маршрутизации
 ROUTE_TABLE:
vesbc(config-esbc)# user-interface sip USER_INTRFACE
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_USER_USER
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_USER
vesbc(config-esbc-user-interface-sip)# route-table ROUTE_TABLE
vesbc(config-esbc-user-interface-sip)# exit

#Применить и подтвердить изменения:
vesbc(config-esbc)# do commit
vesbc(config-esbc)# do confirm
```

При такой конфигурации регистрации абонентов будут отправляться в основной транк MAIN_TRUNK.

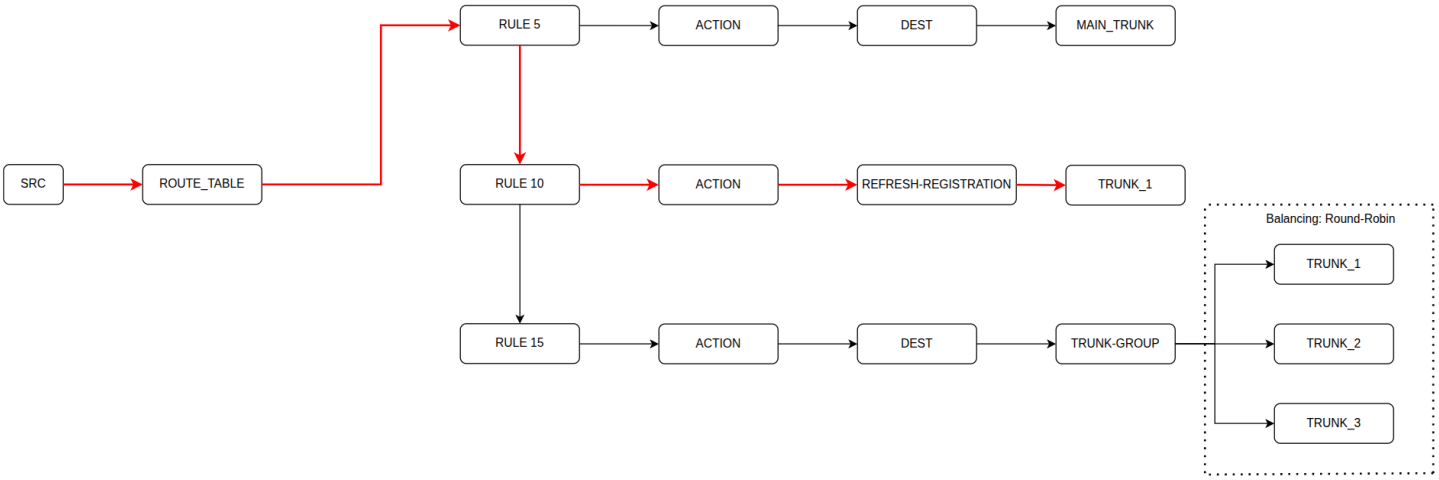


При недоступности транка MAIN_TRUNK следующим правилом в таблице маршрутизации стоит **refresh-registration**, и так как на системе нет активной регистрации этого абонента, то оно пропускается и запрос направляется на следующее направление, то есть в транковую группу TRUNK_GROUP. В транковой группе настроена балансировка round-robin, поэтому запрос смаршрутизируется на один из транков, входящих в эту группу. При этом в системе появится запись о том, что данный абонент зарегистрирован на конкретном транке.

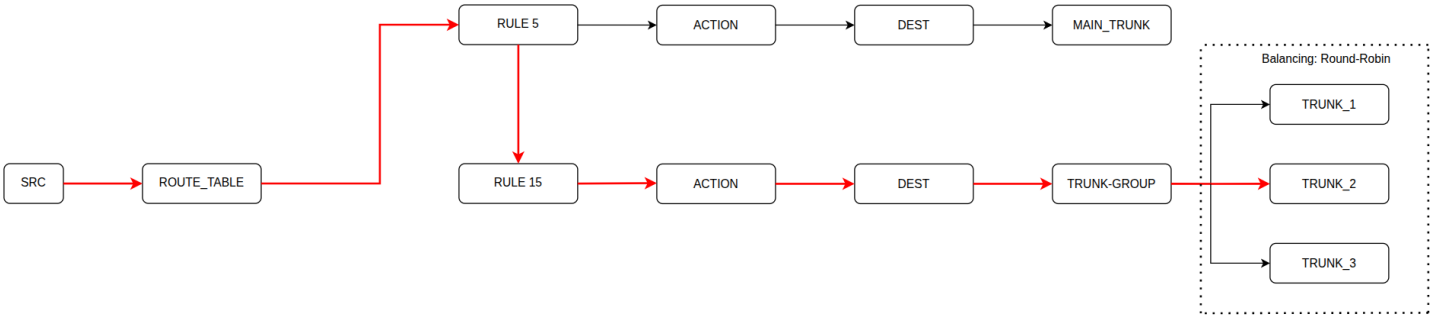




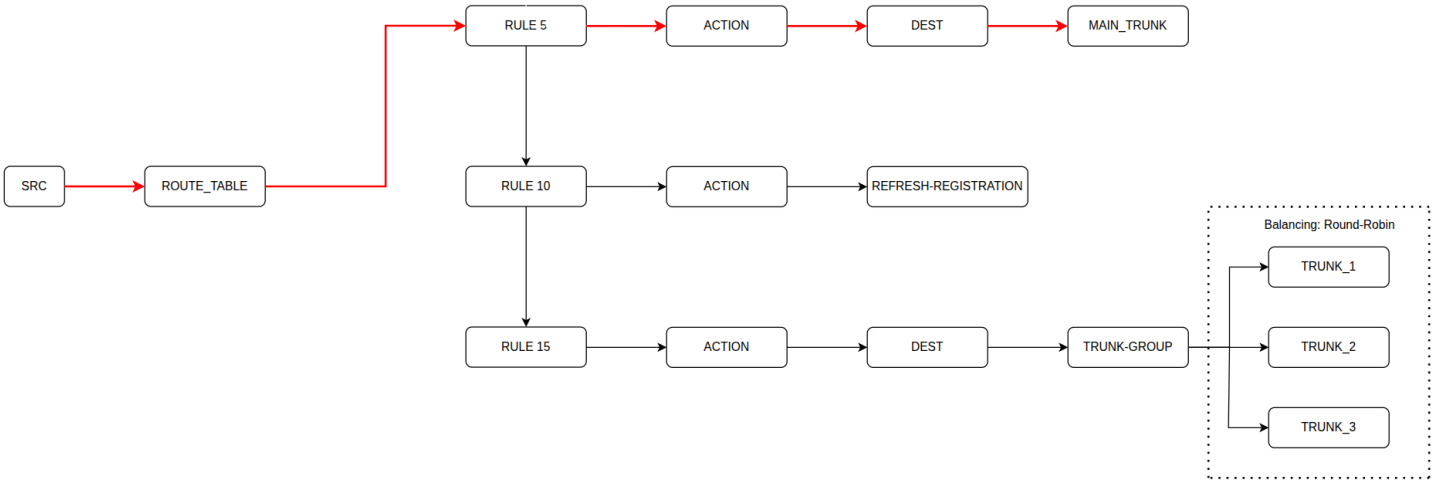
Теперь при обновлении регистрации абонента, если основной транк не будет доступен, запрос регистрации смаршрутизируется на правило **refresh-registration.** И так как на ESBC есть запись о том, что абонент зарегистрирован на конкретном направлении, то регистрация направится сразу в этот транк.



Если бы в таблице маршрутизации не стояло правило **refresh-registration**, то при обновлении регистрации абонент был бы зарегистрирован на следующем транке из транковой группы.



После восстановления доступа к основному транку MAIN_TRUNK запросы регистрации будут отправляться на него согласно таблице маршрутизации.



⚠ Правило маршрутизации **refresh-registration** работает только для запросов REGISTER.

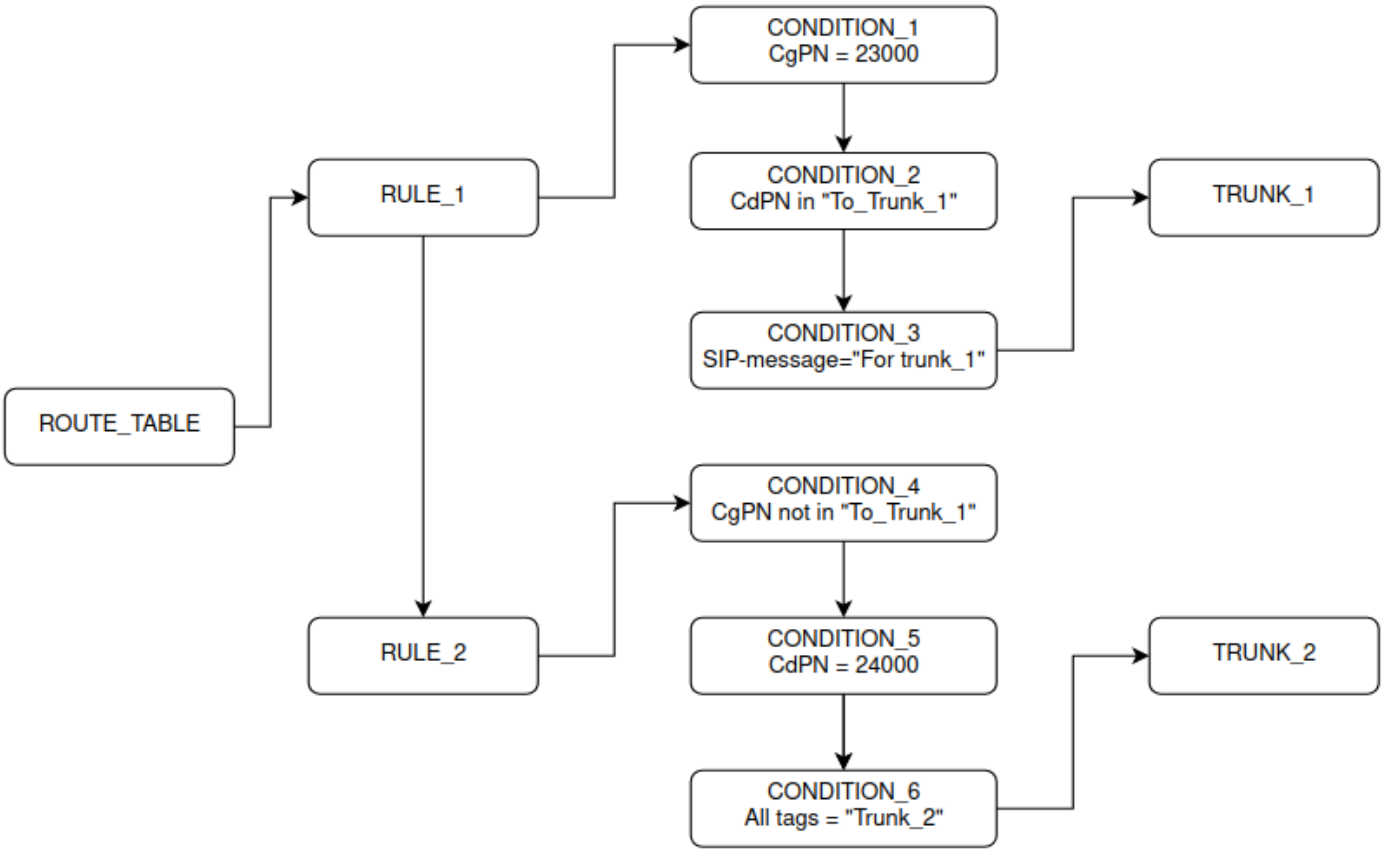
9.8.5 Использование условий в таблице маршрутизации

Условия маршрутизации:

- безусловная маршрутизация — маршрутизация всех SIP-сообщений без анализа содержимого;
- CgPN, CdPN — анализируются user-part из заголовков From и To в сообщении SIP и сравниваются либо с шаблоном, либо со списком префиксов/номеров;
- SIP-message — маршрутизация по наличию любого совпадения в любой части SIP-сообщения;
- Тег — маршрутизация по наличию совпадения в любом из тегов/во всех тегах сессии.

Каждое правило может содержать до 8 условий.

Пример работы условий:



В таблице маршрутизации два правила, у обоих есть условия. Например, приходит вызов, у которого номер А=23000, номер Б=24000 и кастомный заголовок "Trunk: For trunk_1". ESBC заходит в RULE_1 и анализирует условие CONDITION_1, условие истинно, далее происходит анализ условия из CONDITION_2, ESBC проверяет список префиксов из префикс-группы "To_Trunk_1", находит префикс "24.*", который совпадает с CdPN – условие истинно, далее происходит анализ условия из CONDITION_3, условие также истинно. Значит правило RULE_1 подходит для маршрутизации, и вызов отправляется в TRUNK_1.

Рассмотрим вызов с номерами, которые подходят под условия из RULE_2.

Приходит вызов, у которого номер А=25000, номер Б=24000 и кастомный заголовок "Trunk: For trunk_2". ESBC заходит в RULE_1 и анализирует условие CONDITION_1, условие ложно. Значит правило не подходит (правило подходит, только если все условия внутри правила истинны). Далее ESBC переходит к RULE_2, анализирует условие CONDITION_4, ESBC проверяет список префиксов из префикс группы "To_Trunk_1", не найдено префиксов, совпадающих с CgPN – условие истинно, далее происходит анализ условия из CONDITION_5, условие истинно, далее происходит анализ условия из CONDITION_6, ESBC проверяет теги сессии, которые могут быть добавлены с помощью common-модификаторов, найден тег

"Trunk_2" – условие также истинно. Значит правило RULE_2 подходит для маршрутизации, и вызов отправляется в TRUNK_2.

Если приходит вызов, который не подходит ни под одно правило, то такой вызов отбивается.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Префикс-группа "To_Trunk_1" с типом "Список префиксов"
vesbc(config-esbc)# prefix-group prefix-list To_Trunk_1
vesbc(esbc-prefix-group)# prefix 0 "3\\d{4}"
vesbc(esbc-prefix-group)# prefix 1 "24.*"
vesbc(esbc-prefix-group)# prefix 2 "555"
vesbc(esbc-prefix-group)# exit

# Таблица common-модификаторов, привязанная к направлению, с которого поступил второй вызов
vesbc(config-esbc)# mod-table common ADD_TAG
vesbc(esbc-mod-table-common)# mod 0
vesbc(esbc-mod-table-common-modification)# action 0 add-tag Trunk_2
vesbc(esbc-mod-table-common-modification)# exit
vesbc(esbc-mod-table-common)# exit

vesbc(config-esbc)# route-table ROUTE_TABLE

#Добавление первого правила с условиями CONDITION_1, CONDITION_2, CONDITION_3 и действием
отправить вызов в TRUNK_1:
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# condition 0 cgpn pattern ^23000$
vesbc(config-esbc-route-table-rule)# condition 1 cdpn in-prefix-group To_Trunk_1
vesbc(config-esbc-route-table-rule)# condition 2 sip-message "For trunk_1"
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# exit

#Добавление второго правила с условиями CONDITION_4, CONDITION_5, CONDITION_6 и действием
отправить вызов в TRUNK_2:
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# condition 0 cgpn not-in-prefix-group To_Trunk_1
vesbc(config-esbc-route-table-rule)# condition 1 cdpn pattern ^24000$
vesbc(config-esbc-route-table-rule)# condition 2 tag all Trunk_2
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_2
```

Логика применения условий

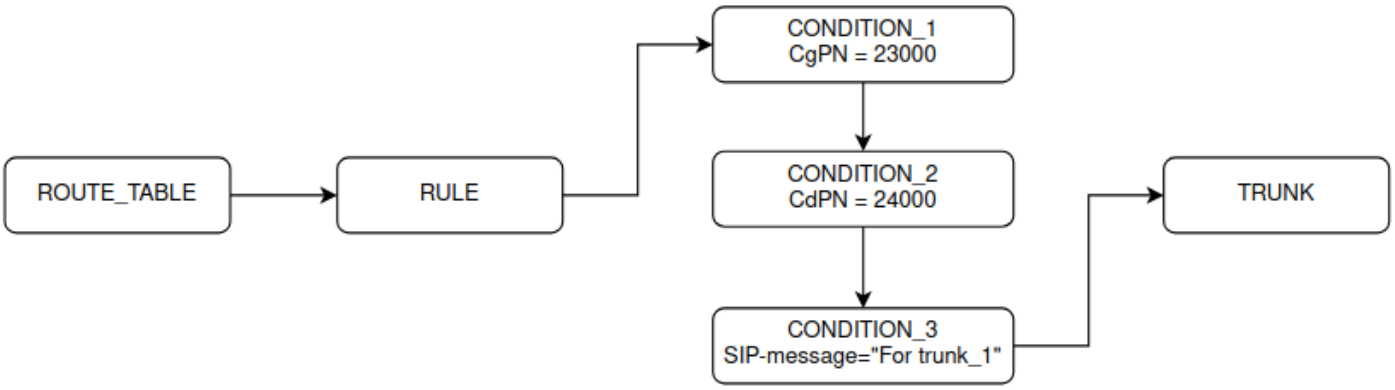
По умолчанию все условия применяются с логикой "И", то есть действие правила выполнится только при успешной проверке всех условий, если хотя бы одно условие не выполняется, то ESBC переходит к обработке следующего правила.

Есть возможность с помощью выражения задавать логику применения условий, доступные логические операторы:

- & — логическое И;
- | — логическое ИЛИ;
- ! — логическое отрицание.

Для обращения к условию необходимо использовать его индекс. В выражении можно использовать скобки для группировки условий.

Пример изменения логики применения условий:




```
vesbc#
vesbc# configure
vesbc(config)# esbc

vesbc(config-esbc)# route-table ROUTE_TABLE

#Добавление правила с условиями CONDITION_1, CONDITION_2, CONDITION_3 и действием отправить
вызов в TRUNK_1:
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# condition 0 cgpn pattern ^23000$
vesbc(config-esbc-route-table-rule)# condition 1 cdpn pattern ^24000$
vesbc(config-esbc-route-table-rule)# condition 2 sip-message "For trunk_1"
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# do commit
vesbc(config-esbc-route-table-rule)# do confirm
```

В таблице маршрутизации есть правило с тремя условиями. Например, приходит вызов, у которого номер А=25000, номер Б=26000 и кастомный заголовок "Trunk: For trunk_1". ESBC заходит в RULE_1 и анализирует условие CONDITION_1, условие ложно. Со стандартной логикой применения условий это значит, что правило не подходит и вызов не смаршрутизируется в указанный транк.

Рассмотрим пример с изменением логики применения условий:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

vesbc(config-esbc)# route-table ROUTE_TABLE

#Добавление правила с условиями CONDITION_1, CONDITION_2, CONDITION_3, логическим выражением и
действием отправить вызов в TRUNK_1:
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# condition 0 cgpn pattern ^23000$
vesbc(config-esbc-route-table-rule)# condition 1 cdpn pattern ^24000$
vesbc(config-esbc-route-table-rule)# condition 2 sip-message "For trunk_1"
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# condition expression "(0&1)|2"
vesbc(config-esbc-route-table-rule)# do commit
vesbc(config-esbc-route-table-rule)# do confirm
```

Рассмотрим пример с таким же вызовом. Номер А=25000, номер Б=26000 и кастомный заголовок "Trunk: For trunk_1". ESBC заходит в RULE_1 и анализирует условие CONDITION_1, условие ложно, далее ESBC переходит к CONDITION_2, условие ложно, далее ESBC переходит к CONDITION_3, текст шаблона найден в сообщении – условие истинно. Согласно выражению **"(0&1)|2"** , успешной проверки условия с индексом 2 достаточно для выполнения действия, несмотря на то, что условия с индексами 0 и 1 не прошли проверку. Значит правило подходит для маршрутизации, и вызов отправляется в TRUNK_1.

Синтаксис для написания условий

Для написания условий используются регулярные выражения PCRE.

> Регулярное выражение для условия по номеру или тегу применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать номеру "20405". В данном случае шаблон "^20.*" будет корректным, а шаблон "20" — нет.
> Регулярное выражение для условия по значению SIP-сообщения применяется с логикой частичного совпадения (match).

При вводе выражения в одинарных и двойных кавычках различается логика экранирования символа обратного слеша - «\».

- В одинарных кавычках:

При вводе регулярного выражения в одинарных кавычках символы обратного слеша и двойные кавычки экранируются обратным слешем автоматически.

Пример ввода шаблона:

```
vesbc(config-esbc-route-table-rule)# condition 0 cdpn pattern '\d'
```

Пример отображения шаблона в выводе конфигурации:

```
  route-table ROUTE_TABLE
    rule 0
      condition 0 cdpn pattern "\\d"
    exit
  exit
```

В данном случае будет использоваться поиск совпадения по метасимволу «\d», то есть по совпадению с любой цифрой.

- В двойных кавычках:

При вводе регулярного выражения в двойных кавычках символ обратного слеша используется для экранирования следующего символа, поэтому для использования обратного слеша как символ его нужно дополнительно экранировать обратным слешем.

1. Пример ввода шаблона с одним обратным слешем:

```
  vesbc(config-esbc-route-table-rule)# condition 0 cdpn pattern "\d"
```

Пример отображения шаблона в выводе конфигурации:

```
route-table ROUTE_TABLE
  rule 0
    condition 0 cdpn pattern "d"
  exit
exit
```

В данном случае будет использоваться поиск совпадения по букве «d».

2. Пример ввода шаблона с двумя обратными слешами:

```
vesbc(config-esbc-route-table-rule)# condition 0 cdpn pattern "\\d"
```

Пример отображения шаблона в выводе конфигурации:

```
route-table ROUTE_TABLE
  rule 0
    condition 0 cdpn pattern "\\d"
  exit
exit
```

В данном случае будет использоваться поиск с использованием метасимвола «\d», то есть по совпадению с любой цифрой.

9.9 Настройка модификаторов

ESBC поддерживает два типа модификаторов — **common** и **sip**.

Модификаторы **common** позволяют модифицировать параметры сессии без привязки к протоколу сигнализации. Параметры, доступные для модификации:

- CgPN;
- CdPN;
- Тег.

Модификаторы **sip** позволяют модифицировать любые заголовки сообщений SIP.

В одном модификаторе может быть до 8 модификаций.

Таблицы модификаций применяются в транках, транковых группах и абонентских интерфейсах. Их можно подключить как **out** — тогда правила будут применяться при отправке сообщения или как **in** — тогда правила применяются при получении сообщения. Таблица модификаций, используемая для транковой группы, будет использоваться только в том случае, если в транке, входящем в эту транковую группу, не настроена своя таблица.

В таблицах модификации для отбора значений (header pattern, header value, response-pattern, value-pattern, и др.) используются регулярные выражения PCRE.

Перед использованием модификаторов рекомендуется ознакомиться с описанием синтаксиса регулярных выражений PCRE.

Регулярное выражение для:
- method pattern
- response-pattern
- value-pattern(mod-table sip → mod(no-transit, transit))

применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать значению "20405". В данном случае шаблон "^20.*" будет корректным, а шаблон "20" — нет.
Регулярное выражение для value-pattern (mod-table common → action replace и mod-table sip → mod (copy, replace)) применяется с логикой частичного совпадения (match).

Допускается использование следующей конструкции при составлении регулярных выражений PCRE для помещения значений в локальные переменные (от 0 до 9) с помощью цифр, экранированных обратной чертой ("\\1-9"). "\\0" — весь текст:

```
value-pattern "(some)-(value)"
#Значения some и value заносятся в локальные переменные pcre 1 и 2 соответственно:
 "\\2-\\1"
#Значения переменных меняются местами
```

Результат замены: value-some

Данные переменные используются в рамках одной модификации. Для использования переменных в разных модификациях одной таблицы модификаций используется модификатор типа **copy**.

⚠ При применении на транке/абонентском интерфейсе модификаторов обоих типов одновременно, используется следующий порядок их обработки в зависимости от направления модификации:
- IN — сначала применяется модификатор sip, затем — модификатор common;
- OUT — сначала применяется модификатор common, затем — sip.

9.9.1 Общие модификаторы

Модификаторы **common** позволяют модифицировать параметры сессии без привязки к протоколу сигнализации. Параметры, доступные для модификации:

- CgPN;
- CdPN;
- Tег.

В текущей версии ПО поддерживается только протокол SIP. Учитывая это, при использовании модификаторов в транках и абонентских интерфейсах, модификаторами CdPN можно изменять user part Request-URI и SIP URI заголовка To. Модификатором CgPN можно изменить user-part SIP URI заголовков Contact, From.
Модификаторы **common** позволяют добавлять и удалять теги. Тег – это один из параметров сессии, который представляет собой символьную строку. Тег можно использовать при модификации и маршрутизации.

⚠ Модификаторы common, настроенные в качестве IN, могут влиять на результат маршрутизации при использовании в route-table условий (condition), т. к. правила route-table обрабатываются после применения модификации. Модификаторы, настроенные в качестве OUT, не влияют на результат маршрутизации.

Модификатор замены номера (replace)

Схема вызова:

На ESBC настроена следующая конфигурация:

```
  route-table TO_UAS
    rule 0
      action direct-to-trunk UAS
    exit
  exit
  mod-table common COMMON_MOD
    mod 0
      action 0 replace cgpn
        value-pattern "2(.+)"
        #Осуществляется выбор номеров, начинающихся с 2. Остальная часть номера сохраняется в
локальную переменную 1
        replacement "8\\1"
        #Выполняется замена 2 на 8, и добавляется значение из переменной 1
      exit
      action 5 replace cdpn
        value-pattern "23002"
        #Осуществляется выбор номера 23002
        replacement "22222"
        #Выполняется замена номера 23002 на 22222
      exit
    exit
  exit
  trunk sip UAC
    remote address 192.168.80.26
    remote port 5070
    sip transport UAC
    route-table TO_UAS
    mod-table common in COMMON_MOD
    media resource 0 MEDIA
  exit
  trunk sip UAS
    remote addr 192.168.80.26
    remote port 5080
    sip transport UAS
    media resource 0 MEDIA
  exit
```

На транк UAC приходит INVITE:

```
INVITE sip:24001@192.168.80.129:5080;line=76196f92c8f42f97c3b78125dd1b842c SIP/2.0
Via: SIP/2.0/UDP 192.168.80.26:5070;rport;branch=z9hG4bK-294378-1-1
From: <sip:24001@192.168.80.26:5070>;tag=1
To: <sip:23002@192.168.80.129:5070>
Call-ID: 1-294378@192.168.80.26
Cseq: 1 INVITE
Contact: <sip:24001@192.168.80.26:5070>
Max-Forwards: 70
Allow: INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE
Content-Type: application/sdp
Content-Length:   174

[SDP]...
```

В результате применения модификатора **COMMON_MOD** в транке UAC, из транка UAS будет отправлен INVITE:

```
INVITE sip:22222@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPjWDx0A5VQhCqmg7Sf-wS7Huya0dESxrro
Max-Forwards: 70
From: <sip:84001@192.168.80.129>;tag=epoMSc5qF1.Pfc5pcypn80ONBKHCaO-x
To: <sip:22222@192.168.80.26>
Contact: <sip:84001@192.168.80.129:5080>
Call-ID: 326c0035a257a9f76185383b49df705f
CSeq: 9446 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: 100rel, replaces
Content-Type: application/sdp
Content-Length:   177

[SDP]...
```

В результате действия action 0 replace cgpn выполнена модификация CgPN 24001 на 84001, в результате action 5 replace cdpn — модификация CdPN 23002 на 22222.

> При использовании модификатора CgPN, помимо заголовка From, изменяется user part SIP URI заголовка Contact. При использовании модификатора CdPN, помимо заголовка To, изменяется user part SIP в Request-URI.
> При модификации запросов REGISTER, CgPN влияет на значение user part SIP URI в заголовке From, CdPN на значение user part SIP URI в заголовках To и Contact.
> Значение replacement можно не задавать, в этом случае будет происходить удаление частей заголовков, подходящих по условию.

Модификатор ротации номера (number-rotation)

Схема вызова:

На ESBC настроена следующая конфигурация:

```
route-table TO_UAS
  rule 0
    action direct-to-trunk UAS
  exit
exit
mod-table common COMMON_MOD
  mod 0
    action 0 number-rotation cgpn
      number-list NUMBER_LIST_CGPN
      #Выбор списка, номера из которого будут использованы при ротации
    exit
     action 5 number-rotation cdpn
      number-list NUMBER_LIST_CDPN
      #Выбор списка, номера из которого будут использованы при ротации
      rotation-type random-from-pool
      #Номера из списка будут выбираться случайным образом
    exit
  exit
exit
trunk sip UAC
  remote addr 192.168.80.26
  remote port 5070
  sip transport UAC
  route-table TO_UAS
  mod-table common in COMMON_MOD
  media resource 0 MEDIA
exit
trunk sip UAS
  remote addr 192.168.80.26
  remote port 5080
  sip transport UAS
  media resource 0 MEDIA
exit
prefix-group number-list NUMBER_LIST_CDPN
  number 0 "2331"
  number 5 "2332"
  number 10 "2333"
exit
prefix-group number-list NUMBER_LIST_CGPN
  number 5 "8111"
  number 10 "8112"
  number 15 "8113"
exit
```

На транк UAC приходит INVITE:

```
INVITE sip:24001@192.168.80.129:5080;line=76196f92c8f42f97c3b78125dd1b842c SIP/2.0
Via: SIP/2.0/UDP 192.168.80.26:5070;rport;branch=z9hG4bK-294378-1-1
From: <sip:24001@192.168.80.26:5070>;tag=1
To: <sip:23002@192.168.80.129:5070>
Call-ID: 1-294378@192.168.80.26
Cseq: 1 INVITE
Contact: <sip:24001@192.168.80.26:5070>
Max-Forwards: 70
Allow: INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE
Content-Type: application/sdp
Content-Length:   174

[SDP]...
```

В результате применения модификатора **COMMON_MOD** в транке UAC, из транка UAS будет отправлен INVITE:

```
INVITE sip:2333@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPjWDx0A5VQhCqmg7Sf-wS7Huya0dESxrro
Max-Forwards: 70
From: <sip:8111@192.168.80.129>;tag=epoMSc5qF1.Pfc5pcypn80ONBKHCaO-x
To: <sip:2333@192.168.80.26>
Contact: <sip:8111@192.168.80.129:5080>
Call-ID: 326c0035a257a9f76185383b49df705f
CSeq: 9446 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: 100rel, replaces
Content-Type: application/sdp
Content-Length:   177

[SDP]...
```

В результате модификации CgPN 24001 был заменён на номер 8111 – номер с индексом 5 (наименьший) в списке "NUMBER_LIST_CGPN". CdPN 23002 был заменён на номер 2333 – случайный номер из списка "NUMBER_LIST_CDPN".

При следующем вызове CgPN будет заменён на 8112 – следующий в списке "NUMBER_LIST_CGPN". CdPN будет заменён на случайный номер из списка "NUMBER_LIST_CDPN".

> Для ротации номера можно использовать только префикс-группу с типом "Список номеров" (number-list). Префикс-группы с типом "Список префиксов" (prefix-list) могут использоваться только в условиях.

> При использовании модификатора CgPN, помимо заголовка From, изменяется user part SIP URI заголовка Contact. При использовании модификатора CdPN, помимо заголовка To, изменяется user part SIP в Request-URI.
> При модификации запросов REGISTER, CgPN влияет на значение user part SIP URI в заголовке From, CdPN на значение user part SIP URI в заголовках To и Contact.

Модификатор добавления тега (add-tag)

Схема вызова:

На ESBC настроена следующая конфигурация:

```
route-table TO_UAS
  rule 0
    action direct-to-trunk UAS
  exit
exit
mod-table sip SIP_MOD
  mod 5 add
    method type Invite
    #Тип запроса, в котором необходима модификация
    header name "Source"
    #Название нового заголовка
    header value "UAC"
    #Значение нового заголовка
    condition 0 tag any "From_UAC"
    #Условие для модификации
  exit
exit
mod-table common COMMON_MOD
  mod 0
    action 0 add-tag "From_UAC"
  exit
exit
trunk sip UAC
  remote addr 192.168.80.26
  remote port 5070
  sip transport UAC
  route-table TO_UAS
  mod-table common in COMMON_MOD
  media resource 0 MEDIA
exit
trunk sip UAS
  remote addr 192.168.80.26
  remote port 5080
  sip transport UAS
  mod-table sip out SIP_MOD
  media resource 0 MEDIA
exit
```

На транк UAC приходит INVITE, ESBC добавляет к этой сессии тег "From_UAC":

```
INVITE sip:24001@192.168.80.129:5080;line=76196f92c8f42f97c3b78125dd1b842c SIP/2.0
Via: SIP/2.0/UDP 192.168.80.26:5070;rport;branch=z9hG4bK-294378-1-1
From: <sip:24001@192.168.80.26:5070>;tag=1
To: <sip:23002@192.168.80.129:5070>
Call-ID: 1-294378@192.168.80.26
Cseq: 1 INVITE
Contact: <sip:24001@192.168.80.26:5070>
Max-Forwards: 70
Allow: INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE
Content-Type: application/sdp
Content-Length:   174

[SDP]...
```

Так как тег добавлен (условие выполнено), то выполняется модификация **SIP_MOD** и в транк UAS будет отправлен INVITE с новым заголовком:

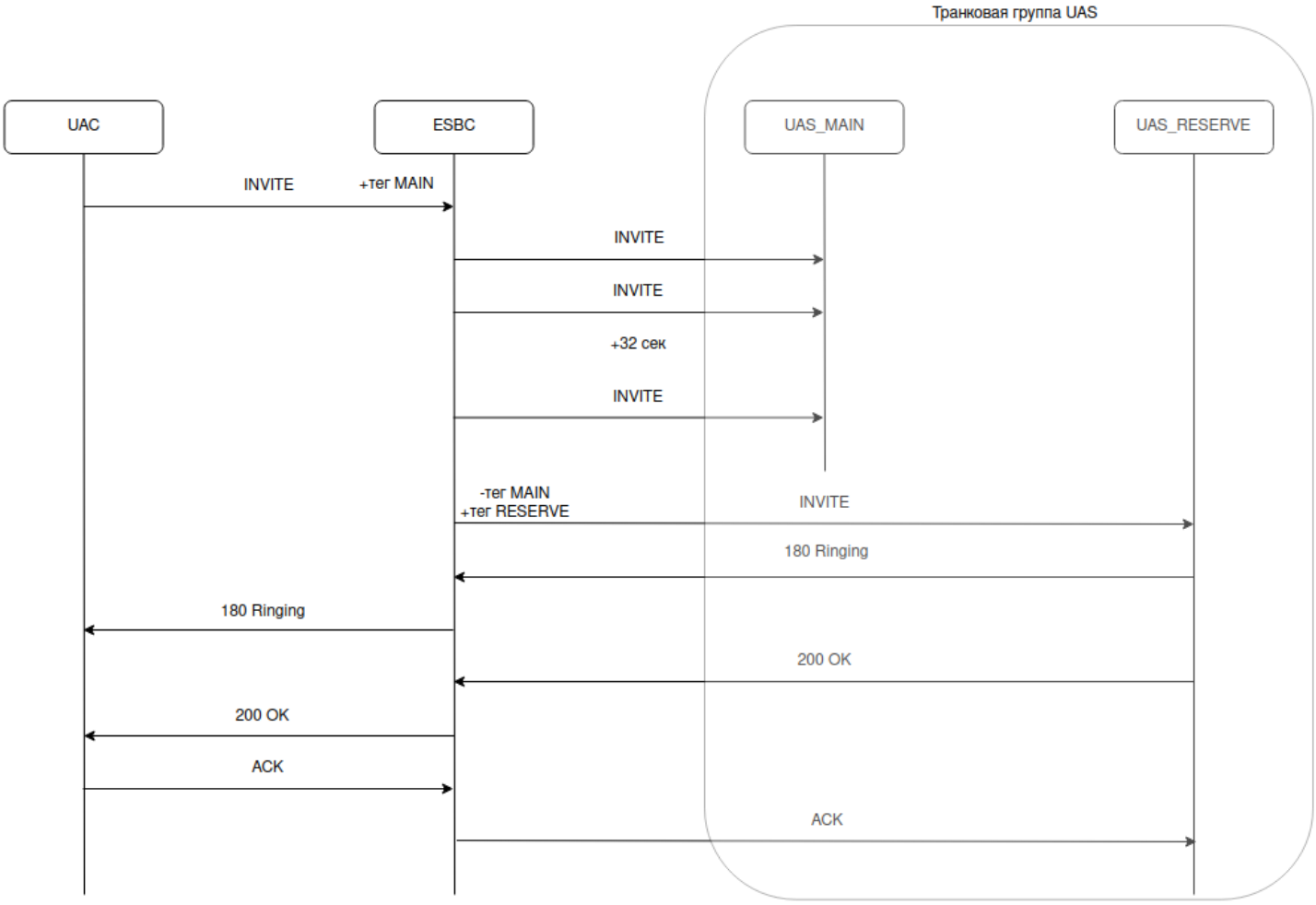
```
INVITE sip:23002@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPjWDx0A5VQhCqmg7Sf-wS7Huya0dESxrro
Max-Forwards: 70
From: <sip:24001@192.168.80.129>;tag=epoMSc5qF1.Pfc5pcypn80ONBKHCaO-x
To: <sip:23002@192.168.80.26>
Contact: <sip:24001@192.168.80.129:5080>
Call-ID: 326c0035a257a9f76185383b49df705f
CSeq: 9446 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: 100rel, replaces
Source: UAC
Content-Type: application/sdp
Content-Length:   177

[SDP]...
```

Если вызов придёт с направления, к которому не привязана таблица модификации **COMMON_MOD**, то тег "From_UAC" к сессии добавлен не будет и в исходящем INVITE не будет заголовка "Source: UAC".

> ⚠ Теги можно использовать в условиях при конфигурации модификаторов или правил маршрутизации.

Модификатор удаления тега (remove-tag)

Схема вызова:



Есть транковая группа UAS, в которую входят 2 транка – UAS_MAIN и UAS_RESERVE. В транковой группе настроена исходящая таблица SIP-модификаторов, которые добавляют заголовок Destination, значение заголовка зависит от того, куда направлен вызов. В транке UAC настроена таблица общих модификаторов, которая добавляет тег MAIN, у транка UAS_RESERVE настроена таблица общих модификаторов, которая удаляет тег MAIN и добавляет тег RESERVE. Конфигурация ESBC выглядит следующим образом:

```
esbc
  media resource MEDIA
    ip address 192.168.126.195
  exit
  sip transport SIP_TRANSPORT
    ip address 192.168.126.195
  exit
  route-table TO_UAS
    rule 0
      action direct-to-trunk-group UAS
    exit
  exit
  trunk-group UAS
    mod-table sip out SIP_MOD
    trunk 5 UAS_MAIN
    trunk 10 UAS_RESERVE
  exit
  mod-table sip SIP_MOD
    mod 5 add
      method type Invite
      header name "Destination"
      header value "Main"
      condition 0 tag all "MAIN"
    exit
    mod 10 add
      method type Invite
      header name "Destination"
      header value "Reserve"
      condition 0 tag all "RESERVE"
    exit
  exit
  mod-table common COMMON_UAS_MAIN
    mod 0
      action 0 add-tag "MAIN"
    exit
  exit
  mod-table common COMMON_UAS_RESERVE
    mod 0
      action 0 add-tag "RESERVE"
      action 1 remove-tag "MAIN"
    exit
  exit
  trunk sip UAC
    sip transport SIP_TRANSPORT
    route-table TO_UAS
    mod-table common in COMMON_UAS_MAIN
    media resource 0 MEDIA
    remote address 192.168.126.190
    remote port 5061
  exit
  trunk sip UAS_MAIN
    sip transport SIP_TRANSPORT
    media resource 0 MEDIA
    remote address 192.168.126.190
    remote port 5064
  exit
  trunk sip UAS_RESERVE
    sip transport SIP_TRANSPORT
```

```
    mod-table common out COMMON_UAS_RESERVE
    media resource 0 MEDIA
    remote address 192.168.126.190
    remote port 5063
  exit
```

На транк UAC приходит INVITE, ESBC добавляет к этой сессии тег "MAIN":

```
INVITE sip:23002@192.168.126.195:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.126.190:5061;rport;branch=z9hG4bK-453702-1-1
From: "24001" <sip:24001@192.168.126.195>;tag=1
To: "23002" <sip:23002@192.168.126.195:5060>
Call-ID: 1-453702@192.168.126.190
Cseq: 1 INVITE
Contact: <sip:24001@192.168.126.190:5061>
Max-Forwards: 70
Allow: INVITE, ACK, BYE, CANCEL
Content-Type: application/sdp
Content-Length:   142

[SDP]...
```

В транк UAS_MAIN отправляется INVITE с заголовком Destination: Main:

```
INVITE sip:23002@192.168.126.190:5064 SIP/2.0
Via: SIP/2.0/UDP 192.168.126.195:5060;rport;branch=z9hG4bKPj16765696-c913-4ffc-9f10-
f92beca0c0d0
Max-Forwards: 70
From: "24001" <sip:24001@192.168.126.195>;tag=b6899be4-af2c-490e-bcf0-a8802cff7276
To: "23002" <sip:23002@192.168.126.190>
Contact: <sip:24001@192.168.126.195:5060;transport=udp>
Call-ID: 50e31c6d5ad746a0eb4a9db3e8543adb
CSeq: 31819 INVITE
Allow: INVITE, ACK, BYE, CANCEL
Supported: 100rel, replaces, ice
Destination: Main
Content-Type: application/sdp
Content-Length:   143

[SDP]...
```

Из транка UAS_MAIN не поступило ответа, вызов отправляется в следующий транк в транковой группе – UAS_RESERVE, из сессии удаляется тег MAIN, добавляется тег RESERVE.

Так как в сессии остался только тег RESERVE, то согласно таблице модификаций SIP_MOD в запрос INVITE добавляется заголовок Destination: Reserve.

```
INVITE sip:23002@192.168.126.190:5063 SIP/2.0
Via: SIP/2.0/UDP
192.168.126.195:5060;rport;branch=z9hG4bKPj40877976-9faa-4950-99e2-507a8176160a
Max-Forwards: 70
From: "24001" <sip:24001@192.168.126.195>;tag=3b1a257a-9167-4d11-b200-caa26358f2e5
To: "23002" <sip:23002@192.168.126.190>
Contact: <sip:24001@192.168.126.195:5060;transport=udp>
Call-ID: 50e31c6d5ad746a0eb4a9db3e8543adb
CSeq: 17681 INVITE
Allow: INVITE, ACK, BYE, CANCEL
Supported: 100rel, replaces, ice
Destination: Reserve
Content-Type: application/sdp
Content-Length:   143

[SDP]...
```

Использование условий в общих модификаторах

В ESBC поддержано использование условий в общих модификаторах:

- CgPN, CdPN — анализируются user-part из заголовков From и To в сообщении SIP и сравниваются либо с шаблоном, либо со списком префиксов/номеров;
- SIP-message — модификация происходит при наличии любого совпадения в любой части SIP-сообщения;
- Тег — модификация происходит при наличии совпадения в любом из тегов/во всех тегах сессии.

Каждая модификация может содержать до 8 условий.

Если в настройках модификатора задано условие, то перед применением модификации параметры сессии проверяются на соответствие условию. Если условий несколько, то для применения модификации необходимо выполнение всех условий.

Пример работы условий приведен в разделе Настройка таблиц маршрутизации. Настройка условий для таблиц маршрутизации и таблиц общих модификаторов идентична.

Логика применения условий

По умолчанию все условия применяются с логикой "И", то есть модификация выполнится только при успешной проверке всех условий, если хотя бы одно условие не выполняется, то ESBC переходит к обработке следующего модификатора.

Есть возможность с помощью выражения задавать логику применения условий, доступные логические операторы:

- & — логическое И;
- | — логическое ИЛИ;
- ! — логическое отрицание.

Для обращения к условию необходимо использовать его индекс. В выражении можно использовать скобки для группировки условий.

Пример настройки логики применения условий приведен в разделе Настройка таблиц маршрутизации.

Синтаксис для написания условий

Для написания условий используются регулярные выражения PCRE.

> Регулярное выражение для условия по номеру или тегу применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать номеру "20405". В данном случае шаблон "^20.*" будет корректным, а шаблон "20" — нет.
> Регулярное выражение для условия по значению SIP-сообщения применяется с логикой частичного совпадения (match).

Описание всех команд для настройки общих модификаторов приведено в разделе Настройки общих модификаторов.

При вводе выражения в одинарных и двойных кавычках различается логика экранирования символа обратного слеша - «\».

- В одинарных кавычках:

При вводе регулярного выражения в одинарных кавычках символы обратного слеша и двойные кавычки экранируются обратным слешем автоматически.

Пример ввода шаблона:

```
vesbc(config-esbc-mod-table-common-modification)# condition 0 cgpn pattern '\d'
```

Пример отображения шаблона в выводе конфигурации:

```
mod-table common MOD
  mod 0
    condition 0 cgpn pattern "\\d"
  exit
exit
```

В данном случае будет использоваться поиск совпадения по метасимволу «\d», то есть по совпадению с любой цифрой.

- В двойных кавычках:

При вводе регулярного выражения в двойных кавычках символ обратного слеша используется для экранирования следующего символа, поэтому для использования обратного слеша как символ его нужно дополнительно экранировать обратным слешем.

1. Пример ввода шаблона с одним обратным слешем:

```
vesbc(config-esbc-mod-table-common-modification)# condition 0 cgpn pattern "\d"
```

Пример отображения шаблона в выводе конфигурации:

```
mod-table common MOD
  mod 0
    condition 0 cgpn pattern "d"
  exit
exit
```

В данном случае будет использоваться поиск совпадения по букве «d».

2. Пример ввода шаблона с двумя обратными слешами:

```
vesbc(config-esbc-mod-table-common-modification)# condition 0 cgpn pattern "\\d"
```

Пример отображения шаблона в выводе конфигурации:

```
mod-table common MOD
  mod 0
    condition 0 cgpn pattern "\\d"
  exit
exit
```

В данном случае будет использоваться поиск совпадения с использованием метасимвола «\d», то есть по совпадению с любой цифрой.

9.9.2 Модификаторы SIP

Данный тип модификации позволяет изменять любые заголовки сообщений SIP.

> ⚠ Процесс модификации заголовков отличается в зависимости от режима использования модификатора IN или OUT.

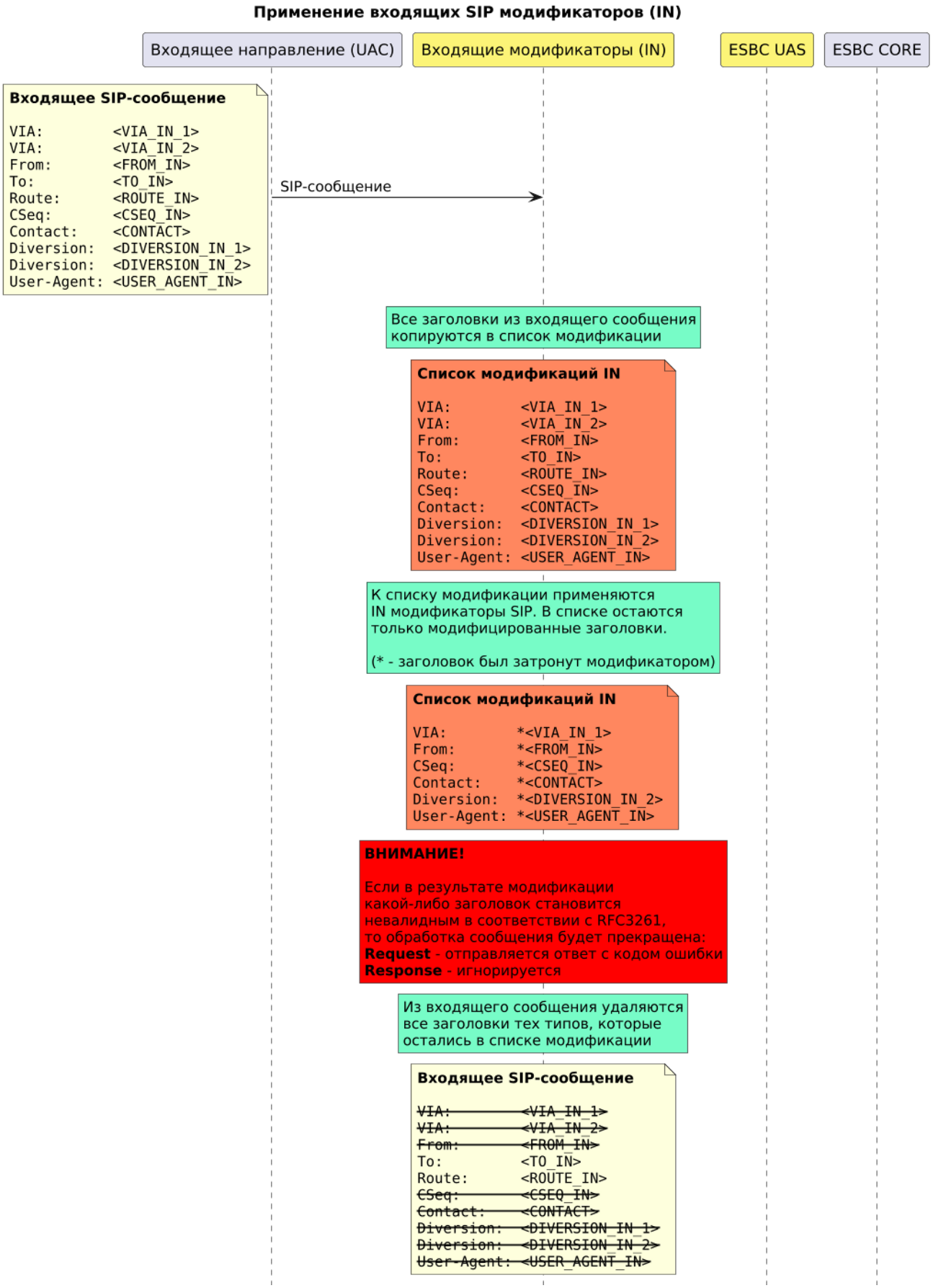
Существуют ограничения на модификацию основных заголовков SIP, к которым относятся: Call-ID, From, To, Via, CSeq, Contact, Max-Forwards, Route, Record-Route, Content-Type, Content-Length, Require, Supported.

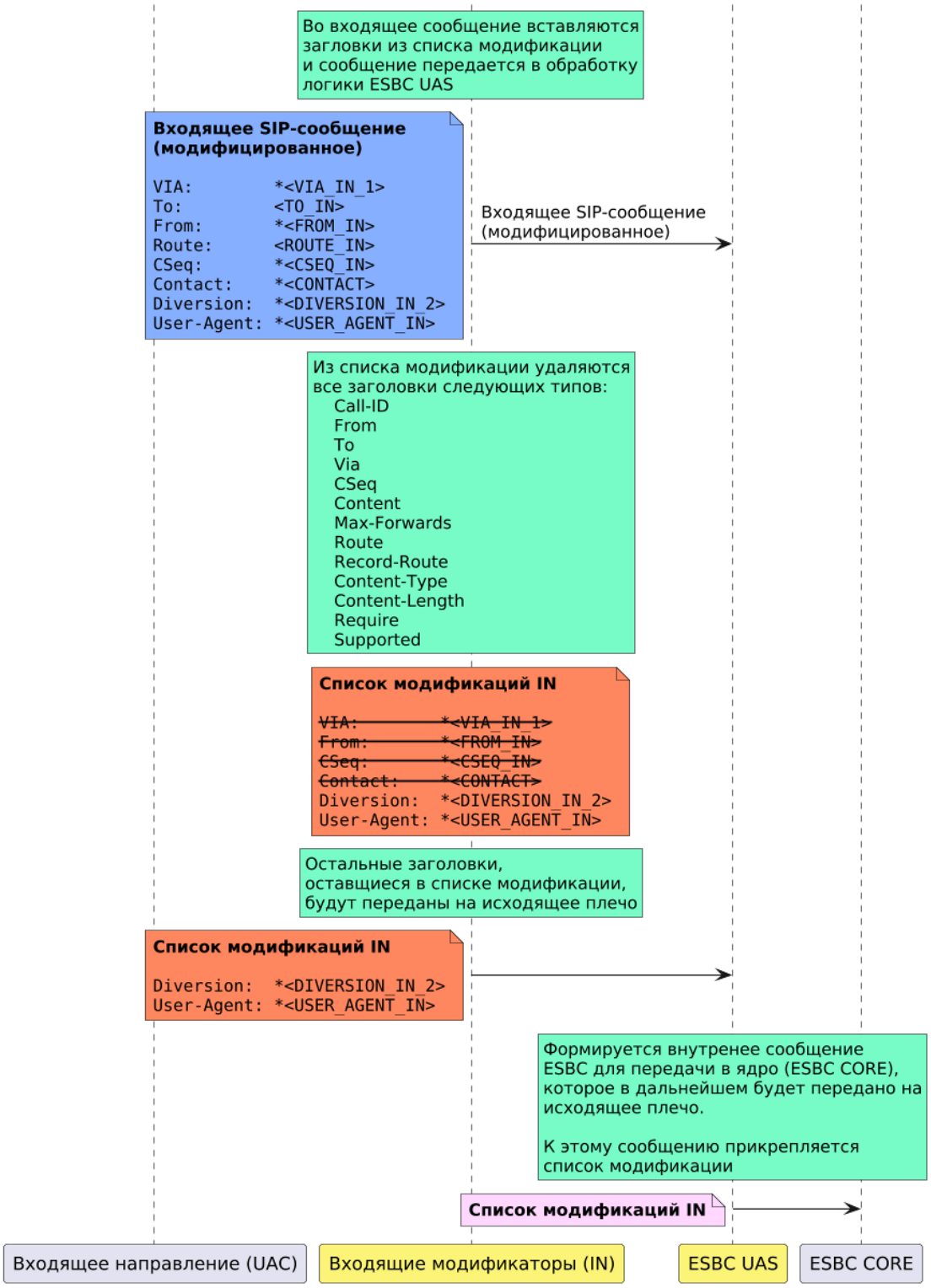
После применения к сообщению модификатора IN и использования модификаций основных заголовков, дальнейшая обработка диалога sip будет осуществляться в соответствии с модифицированным сообщением, т. к. в ядро системы попадает модифицированное сообщение. В результате в ответных сообщениях будут использоваться данные, которые могут отличаться от исходного сообщения. Модификация IN также влияет на дальнейшую маршрутизацию сообщения.

Применение к сообщению модификатора OUT и использования модификаций основных заголовков, изменяет только значения заголовков непосредственно перед отправкой, но не влияет на последующие сообщения в диалоге, т. к. исходное сообщение формируется ядром системы до применения модификаторов OUT.

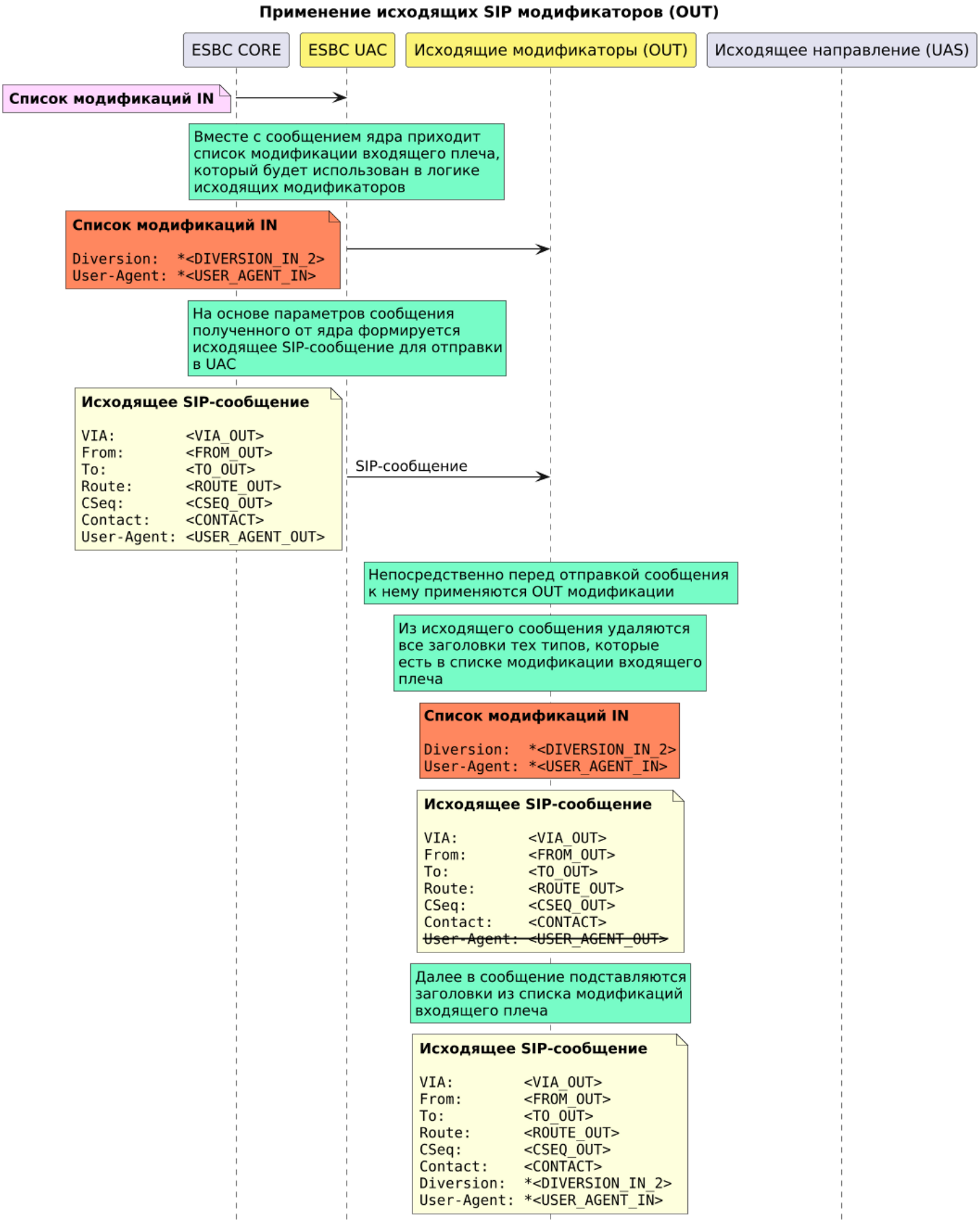
Применение модификаторов к основным заголовкам SIP может привести к нарушению обработки сообщений.

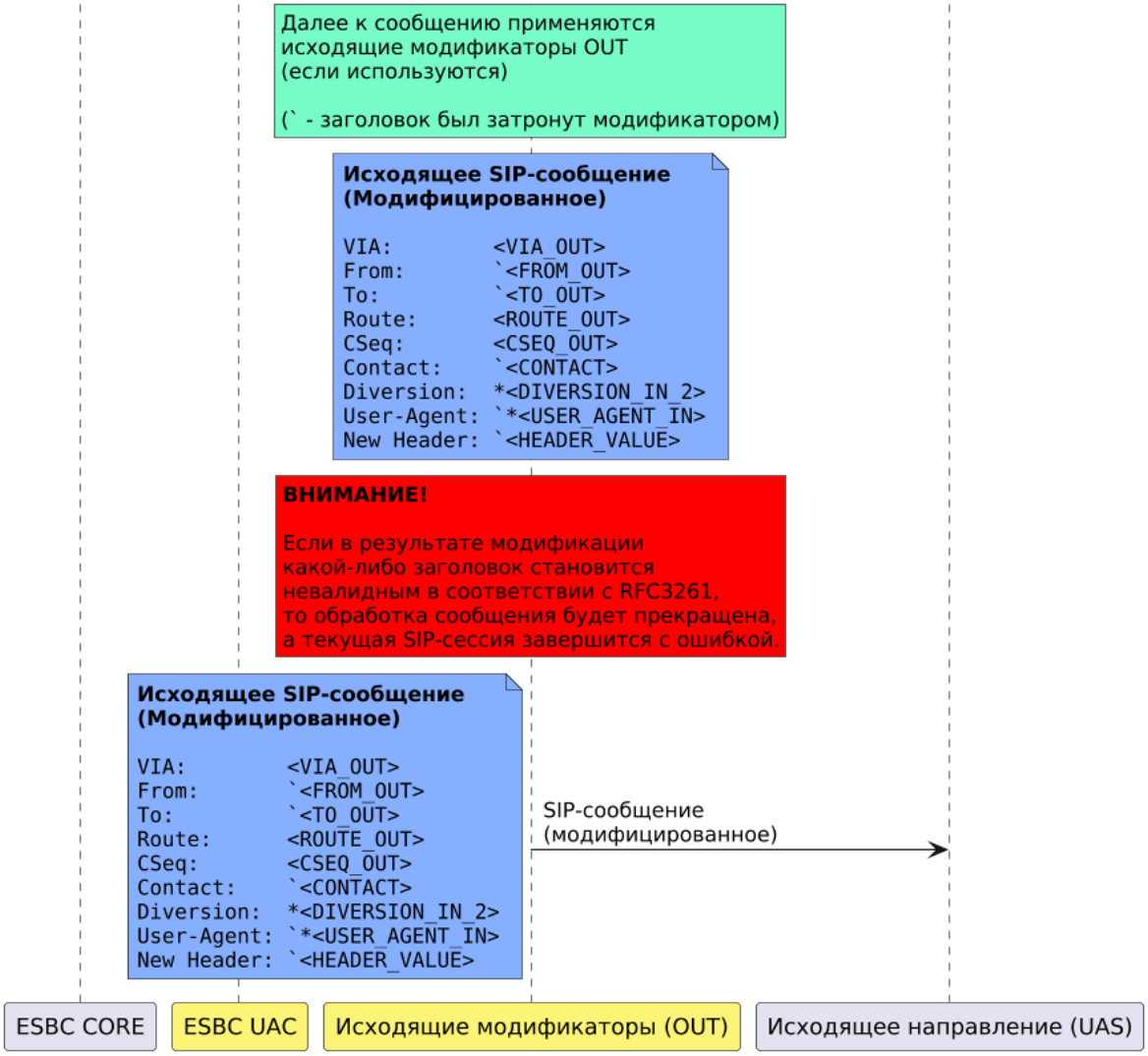
Логика обработки сообщения SIP при использовании IN-модификации:




Во входящее сообщение вставляются загловки из списка модификации и сообщение передается в обработку логики ESBC UAS
Входящее SIP-сообщение (модифицированное)
VIA: *<VIA_IN_1>
To: <TO_IN>
From: *<FROM_IN>
Route: <ROUTE_IN>
CSeq: *<CSEQ_IN>
Contact: *<CONTACT>
Diversion: *<DIVERSION_IN_2>
User-Agent: *<USER_AGENT_IN>
Входящее SIP-сообщение (модифицированное)
Из списка модификации удаляются все заголовки следующих типов:
Call-ID
From
To
Via
CSeq
Content
Max-Forwards
Route
Record-Route
Content-Type
Content-Length
Require
Supported
Список модификаций IN
VIA: *<VIA_IN_1>
From: *<FROM_IN>
CSeq: *<CSEQ_IN>
Contact: *<CONTACT>
Diversion: *<DIVERSION_IN_2>
User-Agent: *<USER_AGENT_IN>
Остальные заголовки, оставщиеся в списке модификации, будут переданы на исходящее плечо
Список модификаций IN
Diversion: *<DIVERSION_IN_2>
User-Agent: *<USER_AGENT_IN>
Формируется внутренее сообщение ESBC для передачи в ядро (ESBC CORE), которое в дальнейшем будет передано на исходящее плечо.
К этому сообщению прикрепляется список модификации
Список модификаций IN
Входящее направление (UAC)
Входящие модификаторы (IN)
ESBC UAS
ESBC CORE

Логика обработки сообщения SIP при использовании OUT-модификации:





Модификаторы SIP позволяют гибко осуществлять выбор требуемого метода (Request) или ответного сообщения (Response) по конкретному коду. Для этого используются команды:

- *method pattern <PATTERN>* — необходимый метод или несколько методов задается при помощи регулярного выражения PCRE.
- *method type <TYPE>* — необходимый метод выбирается из списка шести базовых методов стандарта RFC3261 (REGISTER, INVITE, ACK, CANCEL, BYE и OPTIONS).
- *response-pattern <PATTERN>* — необходимый код ответного сообщения задается при помощи регулярного выражения PCRE.

Команда *method type* аналогична команде *method pattern* и может использоваться в случае, когда модификацию требуется осуществлять только в одном из шести вышеуказанных методов. При использовании этой команды нет необходимости использовать pattern для написания регулярного выражения, достаточно выбрать метод из списка. Команды *method type* и *method pattern* являются взаимоисключающими.

При использовании *pattern* имейте в виду, что по умолчанию синтаксис PCRE является регистрозависимым. Поэтому, например, шаблон "method pattern INVITE" не эквивалентен шаблону "method pattern invite" и отбор метода INVITE **не будет** осуществляться при использовании выражения "method pattern invite".

Помимо работы с заголовками, SIP-модификаторы позволяют взаимодействовать с **Request-URI** и **Status-Line**. Для этого необходимо задать соответствующий шаблон при указании header name-pattern.
Пример конфигурации:

```
mod-table sip MOD_RURI
  mod 1 replace
    method type Invite
    header name-pattern Request-URI
    header value-pattern ".*"
    replacement sip:123456789@test.loc:5060
  exit
exit
```

При настройке модификатора обязательно следует указывать командами выше, для каких методов и кодов ответа он будет применяться. Иначе модификация не будет применена ни к одному сообщению.

Примеры отбора сообщений SIP для модификации:

Требуется применять модификатор, который добавляет заголовок Test_header со значением test_value только в сообщение INVITE:

```
mod-table sip SIP_MOD
  mod 1 add
#Т.к. по условию требуется добавлять заголовок только в сообщения INVITE, можно воспользоваться
командой method type:
    method type Invite
    header name Test_header
    header value test_value
  exit
```

Требуется применять модификатор, который добавляет заголовок Test_header со значением test_value только в сообщения INVITE, BYE и в ответы 200 OK:

```
mod-table sip SIP_MOD
  mod 1 add
#Т.к. по условию требуется добавлять заголовок в INVITE и BYE, надо воспользоваться командой
 "method pattern":
    method pattern INVITE|BYE
#Для добавления заголовка в ответы 200 OK следует добавить команду "response-pattern"
    response-pattern 200
    header name Test_header
    header value test_value
  exit
```

Требуется применять модификатор, который добавляет заголовок Test_header со значением test_value во все запросы и ответы:

```
mod-table sip SIP_MOD
  mod 1 add
#Т.к. по условию требуется добавлять заголовок во все методы, используется отбор любых
значений:
    method pattern .+
#Т.к. по условию требуется добавлять заголовок во все ответы, используется отбор любых
значений:
    response-pattern .+
    header name Test_header
    header value test_value
  exit
```

Требуется применять модификатор, который добавляет заголовок Test_header со значением test_value только в предварительные ответы 100–199:

```
mod-table sip SIP_MOD
  mod 1 add
#Т.к. по условию требуется добавлять заголовок во все ответы от 100 до 199, используется,
например, регулярное выражение "1\\d{2}"
    response-pattern "1\\d{2}"
    header name Test_header
    header value test_value
  exit
```

Поддерживаемые модификации

Поддерживаются следующие типы модификации:

- **add** — добавление заголовка.
- **no-transit** — удаление заголовка. Данная модификация применяется только при использовании в качестве **out** (таблицы **in** всегда удаляют все заголовки, полученные в сообщении из сети).
- **replace** — замена заголовка.
- **transit** — передача заголовка. Данная модификация применяется только при использовании в качестве **in** (таблицы **out** всегда передают все заголовки, полученные с другого плеча).
- **copy** — позволяет скопировать значение или часть значения заголовка в переменную для использования этого значения в модификаторах **add** или **transit** в рамках одной таблицы модификаций (на одном плече вызова).

Порядок применения модификаций в таблице

Модификации в рамках одной таблицы применяются последовательно ко всем заголовкам в порядке, добавленном в конфигурации, т. е. первая модификация применяется ко всем заголовкам, затем вторая модификация применится ко всем заголовкам и т. д.

В результате если какой-либо заголовок был добавлен модификацией add, а затем этот же заголовок был указан в правиле no-transit, то в исходящем сообщении этот заголовок не будет передан.

Пример:

Таблица модификации SIP_MOD используется в качестве OUT:

```
mod-table sip SIP_MOD
  mod 1 add
    method pattern ".+"
    response-pattern ".+"
    header name Test_header
    header value Test_value
  exit
  mod 2 no-transit
    header name-pattern "Test_header"
    method pattern ".+"
    response-pattern ".+"
    value-pattern "Test_value"
  exit
```

Заголовок Test_header не будет передан.

Описание всех команд для настройки общих модификаторов приведено в разделе Настройки SIP-модификаторов.

Модификатор добавления заголовка (add)

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. Требуется, чтобы в запросе INVITE, который отправляется в TRUNK_OUT, был добавлен заголовок Test_header со значением example string.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN:
vesbc(config-esbc)# mod-table sip MODTABLE_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила на добавление заголовка:
vesbc(esbc-mod-table-sip)# mod 0 add
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет добавлен заголовок (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, который необходимо вставить (в данном случае Test_header):
vesbc(esbc-mod-table-sip-modification)# header name Test_header

#Указать содержимое заголовка, которое необходимо вставить (в данном случае example string):
vesbc(esbc-mod-table-sip-modification)# header value "example string"

vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:24000@192.168.114.130:5461 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.129:5461;branch=z9hG4bK-372660-1-5
From: "Simple UAC send bye" <sip:24001@192.168.114.130;cpc=priority>;tag=1372660
To: "24000" <sip:24000@192.168.114.130>
Call-ID: 1-372660@192.168.114.129
CSeq: 1 INVITE
Contact: <sip:24001@192.168.114.129:5461>
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 149

v=0
o=tester 1256 654321 IN IP4 192.168.114.129
s=A conversation
c=IN IP4 192.168.114.129
t=0 0
m=audio 8338 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

На TRUNK_OUT отправляется уже модифицированный INVITE с добавленным заголовком:

```
INVITE sip:24000@192.168.114.129:5460 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.130:5460;rport;branch=z9hG4bKPj-fvzSQlwN2zoMaGUR5JCLMkjmkBV3Vz1
Max-Forwards: 70
From: "Simple UAC send bye" <sip:24001@192.168.114.130>;tag=l2jkRSMeumV03IdhjPntOt7l0XBKy-Ln
To: "24000" <sip:24000@192.168.114.129>
Contact: <sip:24001@192.168.114.130:5460;transport=udp>
Call-ID: P-W.2oee.2vJw0JoaFbNkRDvnxY40FoP
CSeq: 30738 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: timer, 100rel, replaces
Session-Expires: 1800
Min-SE: 90

#Добавленный через таблицу модификаторов заголовок:
Test_header: example string
Content-Type: application/sdp
Content-Length: 157

v=0
o=tester 3927594021 3927594021 IN IP4 192.168.114.130
s=A conversation
c=IN IP4 192.168.114.130
t=0 0
m=audio 8062 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

Модификатор передачи заголовка (transit)

Схема:

В конфигурации настроено два транка и настроена маршрутизация из транка TRUNK_IN в TRUNK_OUT. Требуется передать заголовок "User-Agent" из входящего INVITE, только если в нем указано значение "TestUA".

> ⚠ По умолчанию все необязательные и пользовательские заголовки удаляются на входящем плече и не передаются на исходящее плечо.

Решение:

Создаем таблицу модификации MODTABLE_IN:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN:
vesbc(config-esbc)# mod-table sip MODTABLE_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила для транзита заголовков:
vesbc(esbc-mod-table-sip)# mod 0 transit
vesbc(esbc-mod-table-sip-modification)#

#Выбор метода, в котором будет осуществляться поиск заголовка (по условиям задачи – INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать шаблон для выбора названия заголовка, который необходимо передавать (по условиям
задачи – User-Agent):
vesbc(esbc-mod-table-sip-modification)# header name-pattern User-Agent

#Указать содержимое заголовка, при совпадении с которым заголовок будет передан (по условиям
задачи – TestUA):
vesbc(esbc-mod-table-sip-modification)# header value-pattern TestUA
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit
```

Используем созданную таблицу в качестве IN для транка TRUNK_IN:

```
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

В результате, когда в транке TRUNK_IN будет получен INVITE, содержащий заголовок "User-Agent" со значением "TestUA", в исходящем INVITE также будет присутствовать этот заголовок:

```
#INVITE, полученный в TRUNK_IN:
INVITE sip:23002@192.168.23.199:5070 SIP/2.0
Via: SIP/2.0/UDP 192.168.23.200:5070;rport;branch=z9hG4bK-17639-1-1
From: sipp <sip:24001@192.168.23.200:5070>;tag=1
To: sut <sip:23002@192.168.23.199:5070>
Call-ID: 1-17639@192.168.23.200
Cseq: 1 INVITE
Contact: <sip:24001@192.168.23.200:5070>
Max-Forwards: 70
Subject: Performance Test
Allow: INVITE, ACK, BYE, CANCEL, PRACK, REGISTER, INFO, REFER, NOTIFY, OPTIONS, SUBSCRIBE,
MESSAGE, UPDATE, PUBLISH
Content-Type: application/sdp
User-Agent: TestUA
Content-Length:   240

[SDP]

#INVITE, отправленный с TRUNK_OUT:
INVITE sip:23002@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPj88572ec4-0af0-4323-abc6-fe1db1ea6e37
Max-Forwards: 70
From: "sipp" <sip:24001@192.168.80.129>;tag=7da4c833-38da-4523-89d7-adc88b581397
To: "sut" <sip:23002@192.168.80.26>
Contact: <sip:24001@192.168.80.129:5080;transport=udp>
Call-ID: d21bdedb499dbfe992a939a27255c536
CSeq: 6199 INVITE
Allow: INVITE, ACK, BYE, CANCEL, PRACK, REGISTER, INFO, REFER, NOTIFY, OPTIONS, SUBSCRIBE,
MESSAGE, UPDATE
Supported: 100rel, replaces, ice
User-Agent: TestUA
Content-Type: application/sdp
Content-Length:   241

[SDP]
```

Данный модификатор сработает только в случае, когда значение заголовка User-Agent будет "TestUA". Если значение отличается, то заголовок User-Agent не будет передаваться на второе плечо. Для передачи заголовка с любым значением не следует использовать команду *header value-pattern* в модификаторе, требуется указать в ней все возможные варианты, например, *header value-pattern .**

Модификатор удаления заголовка (no-transit)

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. В TRUNK_OUT отправляется запрос INVITE, в теле которого есть заголовок Test_header. Требуется, чтобы в запросе INVITE, который отправляется в TRUNK_OUT, вырезался заголовок Test_header, если в его содержимом есть "example string".

> ⚠ По умолчанию все необязательные и пользовательские заголовки удаляются на входящем плече и не передаются на исходящее плечо. В данном примере демонстрируется модификация только для исходящего плеча (TRUNK_OUT), поэтому подразумевается что на входящем плече (TRUNK_IN) настроен модификатор transit для заголовка Test_header.

> ⚠ При применении no-transit к Request-URI и Status-Line отменятся их модификации, но сами они не удалятся.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_OUT:
vesbc(config-esbc)# mod-table sip MODTABLE_OUT
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила на удаление заголовка:
vesbc(esbc-mod-table-sip)# mod 0 no-transit
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет удален заголовок (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, который необходимо удалить (в данном случае Test_header):
vesbc(esbc-mod-table-sip-modification)# header name-pattern Test_header

#Указать содержимое заголовка, при совпадении с которым заголовок будет удален (в данном случае
 example string):
vesbc(esbc-mod-table-sip-modification)# header value-pattern "example string"

vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table)# exit

#Привязать таблицу модификаторов к исходящему транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip out MODTABLE_OUT

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

До внесения изменений в конфигурацию в TRUNK_OUT отправлялся следующий INVITE:

```
INVITE sip:24000@192.168.114.129:5460 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.130:5460;rport;branch=z9hG4bKPjjju.7u4O03Aty93vQq0Q1huigSIqGVIr
Max-Forwards: 70
From: "Simple UAC send bye" <sip:24001@192.168.114.130>;tag=CW.53L5FPJAUBsiRspMYqtjTtOTzZxHg
To: "24000" <sip:24000@192.168.114.129>
Contact: <sip:24001@192.168.114.130:5460;transport=udp>
Call-ID: V4OORojNahUbinXtA648s9eI2kjE5cCI
CSeq: 18905 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: timer, 100rel, replaces
Session-Expires: 1800
Min-SE: 90
#Заголовок, который должен быть удален:
Test_header: example string
Content-Type: application/sdp
Content-Length:   157

v=0
o=tester 39275952 39275952 IN IP4 192.168.114.130
s=A conversation
c=IN IP4 192.168.114.130
t=0 0
m=audio 8066 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

После внесения изменений в конфигурацию в TRUNK_OUT отправляется следующий INVITE (заголовок Test_header отсутствует):

```
INVITE sip:24000@192.168.114.129:5460 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.130:5460;rport;branch=z9hG4bKPjz8Y5BfoTrBQlqecLCuTIyYn-6rX5dH
Max-Forwards: 70
From: "Simple UAC send bye" <sip:24001@192.168.114.130>;tag=qTwcY3ZHvA6SHvuRsoo7w40r9yXzjEEp
To: "24000" <sip:24000@192.168.114.129>
Contact: <sip:24001@192.168.114.130:5460;transport=udp>
Call-ID: yHvNLSIvp0DQYSRFPRpfgVUv9U0uKEHT
CSeq: 10147 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: timer, 100rel, replaces
Session-Expires: 1800
Min-SE: 90
Content-Type: application/sdp
Content-Length:   157

v=0
o=tester 3927597375 3927597375 IN IP4 192.168.114.130
s=A conversation
c=IN IP4 192.168.114.130
t=0 0
m=audio 8070 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

В случае если в заголовке Test_header будет содержимое, отличное от "example string", заголовок будет отправлен в TRUNK_OUT:

```
INVITE sip:24000@192.168.114.129:5460 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.130:5460;rport;branch=z9hG4bKPj8e1WEAvAy16Bk8Vrj-VZiFK-bNOjnjY9
Max-Forwards: 70
From: "Simple UAC send bye" <sip:24001@192.168.114.130>;tag=R83mrTm4KQsFL1Bk87hTOB8e182yCSJ.
To: "24000" <sip:24000@192.168.114.129>
Contact: <sip:24001@192.168.114.130:5460;transport=udp>
Call-ID: eQueXFpyDZESB.hXK.uCGn7XL7TBUdmQ
CSeq: 8831 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: timer, 100rel, replaces
Session-Expires: 1800
Min-SE: 90

#Заголовок Test_header с содержимым, отличным от "example string", не удаляется:
Test_header: new string
Content-Type: application/sdp
Content-Length:   157

v=0
o=tester 3927597832 3927597832 IN IP4 192.168.114.130
s=A conversation
c=IN IP4 192.168.114.130
t=0 0
m=audio 8074 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

Модификатор транзита и замены заголовка (replace)

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. Из TRUNK_IN приходит INVITE с заголовком Test_header: 123. Требуется, чтобы в TRUNK_OUT отправился INVITE с заголовком Test_header: 123456.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN:
vesbc(config-esbc)# mod-table sip MODTABLE_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила на замену заголовка:
vesbc(esbc-mod-table-sip)# mod 1 replace

#Выбор запроса, в котором будут заменяться заголовки:
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, содержимое которого необходимо заменить:
vesbc(esbc-mod-table-sip-modification)# header name-pattern Test_header

#Указать место в содержимом заголовка, которое необходимо заменить (конец строки исходного
содержимого заголовка):
vesbc(esbc-mod-table-sip-modification)# header value-pattern $

#Добавить правило для подмены содержимого заголовка (к концу строки исходного содержимого
заголовка добавляется 456):
vesbc(esbc-mod-table-sip-modification)# replacement 456

vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:24000@192.168.114.130:5461 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.129:5461;branch=z9hG4bK-375510-1-5
From: "Simple UAC send bye"  <sip:24001@192.168.114.130;cpc=priority>;tag=1375510
To: "24000" <sip:24000@192.168.114.130>
Call-ID: 1-375510@192.168.114.129
CSeq: 1 INVITE
Contact: <sip:24001@192.168.114.129:5461>
Max-Forwards: 70

#Заголовок, который необходимо протранзитить и заменить:
Test_header: 123
Content-Type: application/sdp
Content-Length:   149

v=0
o=tester 123456 654321 IN IP4 192.168.114.129
s=A conversation
c=IN IP4 192.168.114.129
t=0 0
m=audio 7624 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

На TRUNK_OUT отправляется уже модифицированный INVITE с измененным заголовком:

```
INVITE sip:24000@192.168.114.129:5460 SIP/2.0
Via: SIP/2.0/UDP 192.168.114.130:5460;rport;branch=z9hG4bKPjIbcILUaVB0cQTFaGLLb7ccpnbTQIRvV3
Max-Forwards: 70
From: "Simple UAC send bye" <sip:24001@192.168.114.130>;tag=toP8wIO79wo47ChSYy69MFOyd4vhGRNF
To: "24000" <sip:24000@192.168.114.129>
Contact: <sip:24001@192.168.114.130:5460;transport=udp>
Call-ID: dLsiFI4-aD2faceSTLZu.-kuHfN.pJtG
CSeq: 22556 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: timer, 100rel, replaces
Session-Expires: 1800
Min-SE: 90

#Измененный заголовок:
Test_header: 123456
Content-Type: application/sdp
Content-Length:   157

v=0
o=tester 3927607871 3927607871 IN IP4 192.168.114.130
s=A conversation
c=IN IP4 192.168.114.130
t=0 0
m=audio 8090 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

Пример использования локальных переменных pcre в модификации replace (схема та же):

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN:
vesbc(config-esbc)# mod-table sip MODTABLE_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила на замену заголовка:
vesbc(esbc-mod-table-sip)# mod 1 replace

#Выбор запроса, в котором будут заменяться заголовки:
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, содержимое которого необходимо заменить:
vesbc(esbc-mod-table-sip-modification)# header name-pattern Date

#Указать место в содержимом заголовка, которое необходимо заменить (шаблон – дата в формате
 "год-месяц-число"):
vesbc(esbc-mod-table-sip-modification)# header value-pattern "(\\d{4})-(\\d{2})-(\\d{2})"

#Добавить правило для подмены содержимого заголовка (меняем формат даты на "месяц/число/год"):
vesbc(esbc-mod-table-sip-modification)# replacement "\\2/\\3/\\1"

vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:135@10.25.72.151:5060 SIP/2.0
Via: SIP/2.0/UDP 10.25.72.35:5063;rport;branch=z9hG4bK-1104631-1-0
From: <sip:134@10.25.72.151:5060;user=phone>;tag=1
To: <sip:135@10.25.72.151:5060;user=phone>
Call-ID: 1-1104631@10.25.72.35
CSeq: 1 INVITE
Max-Forwards: 70
Supported: replaces, timer
Contact: <sip:134@10.25.72.35:5063>

#Заголовок, который необходимо протранзитить и изменить:
Date: 2024-09-10
Content-Type: application/sdp
Content-Length:   153
```

На TRUNK_OUT отправляется уже модифицированный INVITE с измененным заголовком:

```
Via: SIP/2.0/UDP 10.25.72.151:5060;rport;branch=z9hG4bKPjc5kLf-R0rh5Stla2eTvpoVAxOc0Jr.kX
Max-Forwards: 70
From: <sip:134@10.25.72.151>;tag=lMWgbj2x66hzNDHhP8ef8tWvB2HT2DwH
To: <sip:135@192.168.23.140>
Contact: <sip:134@10.25.72.151:5060;transport=udp>
Call-ID: c09c3761560702267daaee76eb769a9c
CSeq: 5021 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: 100rel, replaces

#Измененный заголовок:
Date: 09/10/2024
Content-Type: application/sdp
Content-Length:   163
```

Пример 2.

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. При отправке сообщения INVITE, полученного из TRUNK_IN в TRUNK_OUT, в host-part заголовков To и From будут использоваться IP-адрес, настроенный в качестве remote address в транке TRUNK_OUT, и IP-адрес sip-транспорта для TRUNK_OUT соответственно (при условии, что в транке TRUNK_OUT не настроен домен).

Требуется при отправке INVITE заменять эти адреса на testdomain.loc.

Решение:

Конфигурация ESBC до использования модификаторов:

```
esbc
  media resource MEDIA_IN
    ip address 192.168.23.199
  exit
  media resource MEDIA_OUT
    ip address 192.168.80.129
  exit
  sip transport IN
    ip address 192.168.23.199
    port 5070
  exit
  sip transport OUT
    ip address 192.168.80.129
    port 5080
  exit
  route-table TO_TRUNK_OUT
    rule 0
      action direct-to-trunk TRUNK_OUT
    exit
  exit
  trunk sip TRUNK_IN
    sip transport IN
    route-table TO_TRUNK_OUT
    media resource 0 MEDIA_IN
    remote address 192.168.23.200
    remote port 5070
  exit
  trunk sip TRUNK_OUT
    sip transport OUT
    media resource 0 MEDIA_OUT
    remote address 192.168.80.26
    remote port 5080
  exit
exit
```

Т. к. в транке TRUNK_OUT не настроен домен, то в host-part заголовков To и From сообщения INVITE будут указаны IP-адреса в соответствии с конфигурацией:

```
INVITE sip:23002@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPj11eb899a-a1c3-4659-b78d-4bba6bdc17ce
Max-Forwards: 70
From: "sipp" <sip:24001@192.168.80.129>;tag=c090d50d-4b15-4db1-94ac-3ea77fe3dd7d
To: "sut" <sip:23002@192.168.80.26>
Contact: <sip:24001@192.168.80.129:5080;transport=udp>
Call-ID: db38ba3ff093153f38b412372a1bed35
CSeq: 20022 INVITE
Allow: INVITE, ACK, BYE, CANCEL, PRACK, REGISTER, INFO, REFER, NOTIFY, OPTIONS, SUBSCRIBE,
MESSAGE, UPDATE
Supported: 100rel, replaces, ice
Content-Type: application/sdp
Content-Length:   241

[SDP]
```

Настраиваем модификатор **MOD_TABLE** для замены IP-адресов на testdomain.loc:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# mod-table sip MOD_TABLE
#Создаем модификатор замены:
vesbc(esbc-mod-table-sip)# mod 1 replace

#С помощью шаблона выбираем заголовки, в которых необходимо выполнить замену:
vesbc(esbc-mod-table-sip-modification)# header name-pattern "(From|To)"

#Указываем метод, в котором необходимо выполнить замену:
vesbc(esbc-mod-table-sip-modification)# method pattern "INVITE"

#Выбираем часть заголовка, которая начинается с символа @, содержит любое количество любых
символов и заканчивается символом >. Под это выражение попадает host-part заголовков:
vesbc(esbc-mod-table-sip-modification)# header value-pattern "@.*>"

#Указываем, что требуется заменить то, что мы получили в предыдущем правиле на @testdomain.loc>
:
vesbc(esbc-mod-table-sip-modification)# replacement "@testdomain.loc>"
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit
vesbc(config-esbc)#
```

Используем это правило в TRUNK_OUT в качестве OUT:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip out MOD_TABLE
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

Теперь в результате модификации в host-part заголовков To и From сообщения INVITE будет указан домен testdomain.loc:

```
INVITE sip:23002@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPje431c80c-619a-43cc-a631-9ad3be4e6679
Max-Forwards: 70
From: "sipp" <sip:24001@testdomain.loc>;tag=0a5f2f31-e27e-4f7c-a3f8-70ca1d5a9f22
To: "sut" <sip:23002@testdomain.loc>
Contact: <sip:24001@192.168.80.129:5080;transport=udp>
Call-ID: 81a874656978d43e11d57e3662996fde
CSeq: 26399 INVITE
Allow: INVITE, ACK, BYE, CANCEL, PRACK, REGISTER, INFO, REFER, NOTIFY, OPTIONS, SUBSCRIBE,
MESSAGE, UPDATE
Supported: 100rel, replaces, ice
Content-Type: application/sdp
Content-Length:   241

[SDP]
```

Модификатор копирования (copy)

Работа с переменными модификатора

Значения переменных, полученных в модификаторе **copy**, можно использовать в модификаторах **replace** (поле) и **add** (поле header value) в рамках одной SIP-сессии.

Значения переменных синхронизируются между плечами вызова, например, если переменная используется в модификаторах **copy** и **add (replace)** на обоих плечах, то при изменении переменной на одном из плеч, новое значение переменной будет передано на другое плечо.

Подстроки ${name_perem} будут заменены на значение соответствующей переменной. Длина переменной — до 128 символов.

> ⚠ В переменную будет скопирована та часть отбора, которая указана в скобках. Например, чтобы в переменную была занесена вся строка, шаблон должен выглядеть так:
>
> ```
> vesbc(esbc-mod-table-sip-modification)# header value-pattern "(.*)"
> ```

Пример 1.

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. В TRUNK_OUT отправляется запрос INVITE, в теле которого есть заголовок Diversion (предварительно следует настроить таблицу модификации на IN транка TRUNK_IN для транзита заголовка Diversion на второе плечо). Требуется, чтобы в запросе INVITE, который отправляется в TRUNK_OUT, вырезался заголовок Diversion, а его значение из user part было добавлено в display name заголовка From.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_OUT:
vesbc(config-esbc)# mod-table sip MODTABLE_OUT
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила copy для копирования значения user part в
переменную user_part:
vesbc(esbc-mod-table-sip)# mod 0 copy
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет использоваться модификатор copy (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, из которого необходимо копировать значение (в данном случае
 Diversion):
vesbc(esbc-mod-table-sip-modification)# header name-pattern Diversion

#Указать содержимое заголовка, при совпадении с которым будет выполнено копирование в
переменную. В переменную будет скопирована та часть отбора, которая указана в скобках:
vesbc(esbc-mod-table-sip-modification)# header value-pattern "<sip:(.+)@"

#Указать переменную, в которую будет скопировано значение, указанное в скобках, в примере - (.
+):
vesbc(esbc-mod-table-sip-modification)# variable-str "user_part"
vesbc(esbc-mod-table-sip-modification)# exit

#Добавление в таблицу модификаторов правила replace для замены заголовка From:
vesbc(esbc-mod-table-sip)# mod 1 replace

#Указать название заголовка, в котором будет осуществляться замена:
vesbc(esbc-mod-table-sip-modification)# header name-pattern "From"

#Выбор запроса, в котором будет использоваться модификатор replace (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать часть содержимого заголовка, которую необходимо заменить:
vesbc(esbc-mod-table-sip-modification)# header value-pattern ".+ <sip:"

#Указать переменную user_part, которая содержит значение, полученное в модификации copy:
vesbc(esbc-mod-table-sip-modification)# replacement "${user_part} <sip:$"
vesbc(esbc-mod-table-sip-modification)# exit

#Добавление в таблицу модификаторов правила no-transit для удаления заголовка Diversion:
vesbc(esbc-mod-table-sip)# mod 2 no-transit
vesbc(esbc-mod-table-sip-modification)# header name-pattern "Diversion"
vesbc(esbc-mod-table-sip-modification)# method type Invite
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к исходящему транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip out MODTABLE_OUT

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
```

```
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:24001@192.168.80.129:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.26:5070;rport;branch=z9hG4bK-473191-1-1
From: test <sip:24001@192.168.80.26:5070>;tag=1
To: sut <sip:23002@192.168.80.129:5070>
Call-ID: 1-473191@192.168.80.26
Cseq: 1 INVITE
Contact: <sip:24001@192.168.80.26:5070>
Max-Forwards: 70
Diversion: <sip:11111@test.loc>;reason=time-of-day
Subject: Performance Test
Allow: INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE
Content-Type: application/sdp
Content-Length:   118

[SDP]...
```

На TRUNK_OUT отправляется уже модифицированный INVITE с измененным заголовком From и без заголовка Diversion:

```
INVITE sip:23002@192.168.80.26:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.129:5080;rport;branch=z9hG4bKPjbURYAQZxa2m1zsT6x.s6RQ28ONE4EifS
Max-Forwards: 70
From: "11111" <sip:24001@192.168.80.129>;tag=Jfl7n8XBMrh6vjCcB036Ogz6QX4BTDCo
To: "sut" <sip:23002@192.168.80.26>
Contact: <sip:24001@192.168.80.129:5080>
Call-ID: bbf5db1c228015eecddfe0d7079ce876
CSeq: 8798 INVITE
Allow: PRACK, SUBSCRIBE, NOTIFY, REFER, INVITE, ACK, BYE, CANCEL, UPDATE
Supported: 100rel, replaces
Content-Type: application/sdp
Content-Length:   119

[SDP]...
```

Пример 2. Хранение переменных в течение сессии

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. Из TRUNK_IN приходит INVITE с заголовком Route. Требуется, чтобы значение заголовка Route из INVITE было добавлено в заголовок INVITE_Route запроса BYE.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN:
vesbc(config-esbc)# mod-table sip MODTABLE_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила copy для копирования значения заголовка Route в
переменную route:
vesbc(esbc-mod-table-sip)# mod 0 copy
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет использоваться модификатор copy (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, из которого необходимо копировать значение (в данном случае
 Route):
vesbc(esbc-mod-table-sip-modification)# header name-pattern Route

#Указать содержимое заголовка, при совпадении с которым будет выполнено копирование в
переменную. В переменную будет скопирована та часть отбора, которая указана в скобках:
vesbc(esbc-mod-table-sip-modification)# header value-pattern "(.+)"

#Указать переменную, в которую будет скопировано значение, указанное в скобках, в примере - (.
+):
vesbc(esbc-mod-table-sip-modification)# variable-str "route"
vesbc(esbc-mod-table-sip-modification)# exit

#Добавление в таблицу модификаторов правила add для добавления заголовка INVITE_Route в BYE:
vesbc(esbc-mod-table-sip)# mod 1 add

#Выбор запроса, в котором будет использоваться модификатор add (в данном случае BYE):
vesbc(esbc-mod-table-sip-modification)# method type Bye

#Название добавляемого заголовка:
vesbc(esbc-mod-table-sip-modification)# header name INVITE_Route

#Указать значение заголовка (переменная route):
vesbc(esbc-mod-table-sip-modification)# header value "${route}"
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:23002@192.168.113.195:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.113.190:5061;rport;branch=z9hG4bK-1142789-1-1
From: "24001" <sip:24001@192.168.113.195>;tag=1
To: "23002" <sip:23002@192.168.113.195:5060>
Call-ID: 1-1142789@192.168.113.190
Cseq: 1 INVITE
Contact: <sip:24001@192.168.113.190:5061>
Max-Forwards: 70
Subject: Performance Test
Allow: INVITE, ACK, BYE, CANCEL
Content-Type: application/sdp
Test_header: string old_value
Test2_header: another string
Test2_header: t"et
Test2_header: t"et
Test2_header: 1234
Test3_header: temporary data
Route: <sip:192.168.113.195>
Supported: timer
Content-Length:   142

[SDP]...
```

Модифицированный запрос BYE, отправленный с TRUNK_IN на TRUNK_OUT, содержит заголовок INVITE_Route со значением заголовка Route из запроса INVITE:

```
BYE sip:23002@192.168.113.190:5063;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.113.195:5060;rport;branch=z9hG4bKPja999d007-7d87-46f7-98ca-
dcc40ee3188e
Max-Forwards: 70
From: "24001" <sip:24001@192.168.113.195>;tag=a25bc557-78fd-4a9c-871d-87099fbfc9ce
To: "23002" <sip:23002@192.168.113.190>;tag=14
Call-ID: 38a3fddd817f6b738707b21ee5fe1c7d
CSeq: 10588 BYE
Allow: INVITE, ACK, BYE, CANCEL
INVITE_Route: <sip:192.168.113.195>
Content-Length:  0

[SDP]...
```

Пример 3. Синхронизация переменных между плечами вызова

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. Из TRUNK_IN приходит INVITE с заголовком Test_header. Требуется, используя только переменную var1, скопировать значение заголовка Test_header из INVITE на первом плече, вставить его в заголовок Test_header_INVITE в ответе 180 на втором плече, скопировать значение заголовка Test_header из ответа 200 на втором плече и вставить его в заголовок Test_header_200 в запрос BYE на первом плече.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_IN_TRUNK_IN для модификаций на первом плече:
vesbc(config-esbc)# mod-table sip MODTABLE_IN_TRUNK_IN
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила copy для копирования значения заголовка Test_header
в переменную var1:
vesbc(esbc-mod-table-sip)# mod 0 copy
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет использоваться модификатор copy (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, из которого необходимо копировать значение (в данном случае
 Test_header):
vesbc(esbc-mod-table-sip-modification)# header name-pattern Test_header

#Указать содержимое заголовка, при совпадении с которым будет выполнено копирование в
переменную. В переменную будет скопирована та часть отбора, которая указана в скобках:
vesbc(esbc-mod-table-sip-modification)# header value-pattern "(.+)"

#Указать переменную, в которую будет скопировано значение, указанное в скобках, в примере - (.
+):
vesbc(esbc-mod-table-sip-modification)# variable-str "var1"
vesbc(esbc-mod-table-sip-modification)# exit

#Добавление в таблицу модификаторов правила add для добавления заголовка Test_header_180 в BYE:
vesbc(esbc-mod-table-sip)# mod 1 add

#Выбор запроса, в котором будет использоваться модификатор add (в данном случае BYE):
vesbc(esbc-mod-table-sip-modification)# method type Bye

#Название добавляемого заголовка:
vesbc(esbc-mod-table-sip-modification)# header name Test_header_200

#Указать значение заголовка (переменная var1):
vesbc(esbc-mod-table-sip-modification)# header value "${var1}"
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN_TRUNK_IN
vesbc(config-esbc-trunk-sip)# exit

#Создание таблицы модификаторов MODTABLE_IN_TRUNK_OUT для модификаций на втором плече:
vesbc(config-esbc)# mod-table sip MODTABLE_IN_TRUNK_OUT
vesbc(esbc-mod-table-sip)#

#Добавление в таблицу модификаторов правила copy для копирования значения заголовка Test_header
в переменную var1:
vesbc(esbc-mod-table-sip)# mod 0 copy
vesbc(esbc-mod-table-sip-modification)#

#Выбор ответа, в котором будет использоваться модификатор copy (в данном случае 200):
```

```
vesbc(esbc-mod-table-sip-modification)# response-pattern 200

#Указать название заголовка, из которого необходимо копировать значение (в данном случае
 Test_header):
vesbc(esbc-mod-table-sip-modification)# header name-pattern Test_header

#Указать содержимое заголовка, при совпадении с которым будет выполнено копирование в
переменную. В переменную будет скопирована та часть отбора, которая указана в скобках:
vesbc(esbc-mod-table-sip-modification)# header value-pattern "(.+)"

#Указать переменную, в которую будет скопировано значение, указанное в скобках, в примере - (.
+):
vesbc(esbc-mod-table-sip-modification)# variable-str "var1"
vesbc(esbc-mod-table-sip-modification)# exit

#Добавление в таблицу модификаторов правила add для добавления заголовка Test_header_INVITE в
 180:
vesbc(esbc-mod-table-sip)# mod 1 add

#Выбор ответа, в котором будет использоваться модификатор add (в данном случае 180):
vesbc(esbc-mod-table-sip-modification)# response-pattern "180"

#Название добавляемого заголовка:
vesbc(esbc-mod-table-sip-modification)# header name Test_header_INVITE

#Указать значение заголовка (переменная var1):
vesbc(esbc-mod-table-sip-modification)# header value "${var1}"
vesbc(esbc-mod-table-sip-modification)# exit
vesbc(esbc-mod-table-sip)# exit

#Привязать таблицу модификаторов к входящему транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip in MODTABLE_IN_TRUNK_OUT

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE, значение заголовка Test_header копируется в переменную var1:

```
INVITE sip:23002@192.168.113.195:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.113.190:5061;rport;branch=z9hG4bK-1151121-1-1
From: "24001" <sip:24001@192.168.113.195>;tag=1
To: "23002" <sip:23002@192.168.113.195:5060>
Call-ID: 1-1151121@192.168.113.190
Cseq: 1 INVITE
Contact: <sip:24001@192.168.113.190:5061>
Max-Forwards: 70
Subject: Performance Test
Allow: INVITE, ACK, BYE, CANCEL
Content-Type: application/sdp
Test_header: INVITE
Supported: timer
Content-Length:   142

[SDP]...
```

Модифицированный ответ 180 Ringing, отправленный с TRUNK_OUT на TRUNK_IN, содержит заголовок Test_header_INVITE со значением из переменной var1:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
192.168.113.190:5061;rport=5061;received=192.168.113.190;branch=z9hG4bK-1152099-1-1
Call-ID: 1-1152099@192.168.113.190
From: "24001" <sip:24001@192.168.113.195>;tag=1
To: "23002" <sip:23002@192.168.113.195>;tag=e782be58-9527-409a-8c9f-de4f32d8fdb2
CSeq: 1 INVITE
Contact: <sip:23002@192.168.113.195:5060;transport=udp>
Allow: INVITE, ACK, BYE, CANCEL, UPDATE
Warning: warning
Test_header_INVITE: INVITE
Content-Length:  0

[SDP]...
```

Далее с TRUNK_OUT на TRUNK_IN приходит ответ 200 OK, значение заголовка Test_header копируется в переменную var1:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.113.195:5060;rport;branch=z9hG4bKPjf8b5024e-425d-434c-83ea-
e315a365644a
From: "24001" <sip:24001@192.168.113.195>;tag=9ca5b465-69ee-4d19-8891-6d99cc8d7b2c
To: "23002" <sip:23002@192.168.113.190>;tag=1
Call-ID: e5e140176f2119a643b0504e9afa69bb
CSeq: 1446 INVITE
Contact: <sip:23002@192.168.113.190:5063;transport=UDP>
Allow: INVITE, ACK, BYE, CANCEL, UPDATE
Test_header: 200 OK
Warning: warning
Min-Expires: 300
Supported: 100rel, timer, replaces, qwe
Require: 100rel, answermode
Content-Type: application/sdp
Content-Length:   166

[SDP]...
```

Модифицированный запрос, отправленный с TRUNK_IN на TRUNK_OUT, содержит заголовок Test_header_200 со значением из переменной var1:

```
BYE sip:23002@192.168.113.190:5063;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.113.195:5060;rport;branch=z9hG4bKPj091e9a98-54a1-4631-
a024-6fe968fa0ba5
Max-Forwards: 70
From: "24001" <sip:24001@192.168.113.195>;tag=9ca5b465-69ee-4d19-8891-6d99cc8d7b2c
To: "23002" <sip:23002@192.168.113.190>;tag=1
Call-ID: e5e140176f2119a643b0504e9afa69bb
CSeq: 1447 BYE
Allow: INVITE, ACK, BYE, CANCEL
Test_header_200: 200 OK
Content-Length:  0

[SDP]...
```

Использование системных переменных

В ESBC поддержано использование системных переменных в модификаторах **replace** (поле) и **add** (поле header value).

Список системных переменных, которые можно использовать при модификации:

- *LOCAL_DOMAIN* — локальный домен;
- *LOCAL_ADDR* — локальный IP-адрес, сейчас то же самое, что LOCAL_HOST;
- *LOCAL_HOST* — локальный домен или IP-адрес;
- *LOCAL_PORT* — локальный порт;
- *REMOTE_DOMAIN* — домен удалённой стороны;
- *REMOTE_ADDR* — IP-адрес удалённой стороны;
- *REMOTE_HOST* — домен или IP-адрес удалённой стороны;
- *REMOTE_PORT* — порт удалённой стороны;
- *IFACE_TYPE* — тип интерфейса (TRUNK или USER);
- *IFACE_ID* — числовой идентификатор интерфейса;
- *IFACE_NAME* — имя интерфейса;
- *VERSION* — версия ESBC (x.y.z.patch);
- *TIMESTAMP* — текущее время в секундах (заполняется на момент применения модификации).

Синтаксис обращения к системным переменным:

```
${VAR_NAME}
```

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. Требуется, чтобы в запросе INVITE, который отправляется в TRUNK_OUT, добавлялся заголовок Call-Info с информацией об имени транка, на который отправляется запрос, и версией ESBC.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_OUT:
vesbc(config-esbc)# mod-table sip MODTABLE_OUT
vesbc(esbc-mod-table-sip)#

#Создание модификатора add для добавления заголовка Call-Info:
vesbc(esbc-mod-table-sip)# mod 0 add
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет использоваться модификатор add (в данном случае INVITE):
vesbc(esbc-mod-table-sip-modification)# method type Invite

#Указать название заголовка, который будет добавлен:
vesbc(esbc-mod-table-sip-modification)# header name "Call-Info"

#Указать содержимое заголовка с использованием системных переменных:
vesbc(esbc-mod-table-sip-modification)# header value "call to ${IFACE_NAME}; ESBC version: $
{VERSION}"
vesbc(esbc-mod-table-sip-modification)# exit

#Привязать таблицу модификаторов к исходящему транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip out MODTABLE_OUT

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:23002@192.168.80.135:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.27:5061;rport;branch=z9hG4bK-2122485-1-1
From: "24001" <sip:24001@192.168.80.27:5061>;tag=1
To: "23002" <sip:23002@192.168.80.135:5060>
Call-ID: 1-2122485@192.168.80.27
Cseq: 1 INVITE
Contact: <sip:24001@192.168.80.27:5061>
Max-Forwards: 70
Subject: Performance Test
Allow: INVITE, ACK, BYE, CANCEL
Content-Type: application/sdp
Content-Length:   138

[SDP]...
```

На TRUNK_OUT отправляется уже модифицированный INVITE с заголовком Call-Info, который содержит имя вызываемой стороны и версию ESBC:

```
INVITE sip:23002@192.168.80.27:5063 SIP/2.0
Via: SIP/2.0/UDP 192.168.80.135:5060;rport;branch=z9hG4bKPj69d21930-f472-4e64-8555-6b68a532deae
Max-Forwards: 70
From: "24001" <sip:24001@192.168.80.135>;tag=f3db1c01-0c06-45cf-8b4d-a233070ae693
To: "23002" <sip:23002@192.168.80.27>
Contact: <sip:24001@192.168.80.135:5060;transport=udp>
Call-ID: 977eea09afecfc44932d4d9c1b2eeb15
CSeq: 6757 INVITE
Allow: INVITE, ACK, BYE, CANCEL
Supported: 100rel, replaces, ice, timer
Call-Info: call to TRUNK_OUT; ESBC version: 1.6.0.0085
Content-Type: application/sdp
Content-Length:   141

[SDP]...
```

Использование условий в SIP-модификаторах

В ESBC поддержано использование условий во всех типах SIP-модификаторов.

Условия можно задать в настройках модификатора. Поддержано 2 типа условий:

- По значению заголовка – определяется именем заголовка, содержимое которого будет проверяться (name-pattern) и значением заголовка, которое необходимо для выполнения условия (value-pattern);
- По тегу – определяется шаблоном, который сравнивается с любым из тегов сессии или со всеми тегами.

Если в настройках модификатора задано условие, то перед применением модификации SIP-сообщение проверяется на соответствие условию. Если условий несколько, то для применения модификации необходимо выполнение всех условий.

Пример работы условий по значению заголовка:

Схема:

В конфигурации есть 2 транка. Вызов, поступающий с TRUNK_IN, маршрутизируется в TRUNK_OUT. Требуется скопировать последнюю цифру номера из From и вставить её в заголовок X-Test-Header при условии, что вторая цифра в номере из To равна 8.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание таблицы модификаторов MODTABLE_OUT:
vesbc(config-esbc)# mod-table sip MODTABLE_OUT
vesbc(esbc-mod-table-sip)#

#Создание модификатора copy для копирования последней цифры из заголовка From в переменную:
vesbc(esbc-mod-table-sip)# mod 0 copy
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, где будет производиться поиск заголовка и копирование значения (любой запрос):
vesbc(esbc-mod-table-sip-modification)# method pattern ".*"

#Указать заголовок, содержимое которого будет скопировано:
vesbc(esbc-mod-table-sip-modification)# header name-pattern "From"

#Указать содержимое заголовка, которое нужно скопировать в переменную (последняя цифра номера):
vesbc(esbc-mod-table-sip-modification)# header value-pattern "(.)@"

#Указать название переменной, в которой будет храниться скопированное значение:
vesbc(esbc-mod-table-sip-modification)# variable-str "var"
vesbc(esbc-mod-table-sip-modification)# exit

#Создание модификатора add для добавления заголовка X-Test-Header:
vesbc(esbc-mod-table-sip)# mod 1 add
vesbc(esbc-mod-table-sip-modification)#

#Выбор запроса, в котором будет использоваться модификатор add (любой запрос):
vesbc(esbc-mod-table-sip-modification)# method pattern ".*"

#Указать название заголовка, который будет добавлен:
vesbc(esbc-mod-table-sip-modification)# header name X-Test-Header

#Указать содержимое заголовка с использованием системных переменных:
vesbc(esbc-mod-table-sip-modification)# header value "${var}"

#Создать условие модификации:
vesbc(esbc-mod-table-sip-modification)# condition 0 header

#Указать заголовок, содержимое которого будет проверяться:
vesbc(esbc-mod-table-sip-modification-condition)# name-pattern "To"

#Указать заголовок, содержимое которого будет проверяться (вторая цифра в номере должна быть
равна 8):
vesbc(esbc-mod-table-sip-modification-condition)# value-pattern ".*sip:.(8).*"
vesbc(esbc-mod-table-sip-modification-condition)# exit

#Привязать таблицу модификаторов к исходящему транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# mod-table sip out MODTABLE_OUT

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
```

```
Configuration has been confirmed. Commit timer canceled.
```

После внесения изменений в конфигурацию с TRUNK_IN приходит следующий INVITE:

```
INVITE sip:18345@192.168.113.195 SIP/2.0
Via: SIP/2.0/UDP 192.168.113.190:5062;rport;branch=z9hG4bKxrnczyxm
Max-Forwards: 70
To: <sip:18345@192.168.113.195>
From: "23001" <sip:23001@192.168.113.195>;tag=bmxas
Call-ID: 977eea09afecfc44932d4d9c1b2eeb15
CSeq: 161 INVITE
Contact: <sip:23001@192.168.113.190:5062;transport=udp>
Content-Type: application/sdp
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,REFER,NOTIFY,SUBSCRIBE,INFO,MESSAGE
Supported: replaces,norefersub
User-Agent: Twinkle/1.10.2
Content-Length: 179

[SDP]...
```

Вторая цифра в номере из заголовка To равна 8, условие выполнено, модификация выполняется. На TRUNK_OUT отправляется уже модифицированный INVITE с заголовком X-Test-Header, который содержит последнюю цифру номера A (1):

```
INVITE sip:18345@192.168.113.190:5461 SIP/2.0
Via: SIP/2.0/UDP
192.168.113.195:5060;rport;branch=z9hG4bKPj5bfa1162-6ccf-41e1-9aff-4d3cf8fe9511
Max-Forwards: 70
From: "23001" <sip:23001@192.168.113.195>;tag=9b1f5ca8-001a-4165-98ba-c7603d2bc0b2
To: <sip:18345@192.168.113.190>
Contact: <sip:23001@192.168.113.195:5060;transport=udp>
Call-ID: 771e99b84559038bd9a784c3a7360db3
CSeq: 1753 INVITE
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, REFER, NOTIFY, SUBSCRIBE, INFO, MESSAGE
Supported: 100rel, replaces, ice, norefersub
X-Test-Header: 1
Content-Type: application/sdp
Content-Length:   181

[SDP]...
```

Пример работы условий по тегу приведен в разделе Общие модификаторы в примере Модификатор удаления тега.

Логика применения условий

По умолчанию все условия применяются с логикой "И", то есть модификация выполнится только при успешной проверке всех условий, если хотя бы одно условие не выполняется, то ESBC переходит к обработке следующего модификатора.

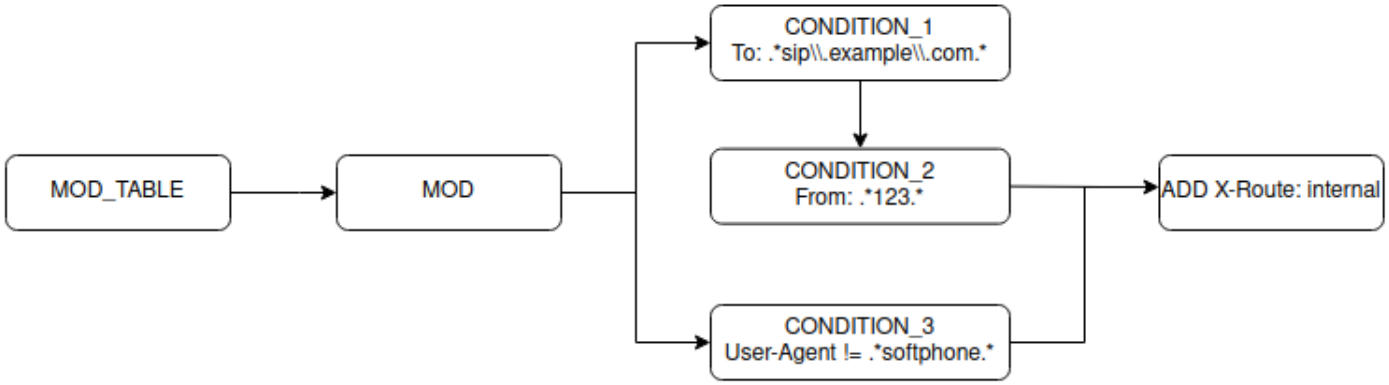
Есть возможность с помощью выражения задавать логику применения условий, доступные логические операторы:

- & — логическое И;
- | — логическое ИЛИ;
- ! — логическое отрицание.

Для обращения к условию необходимо использовать его индекс. В выражении можно использовать скобки для группировки условий.

Пример изменения логики применения условий

Схема:



Есть таблица модификаций, которая добавляет заголовок X-Route: internal, если заголовок To содержит "sip.example.com" и заголовок From содержит "123", либо, если эти условия не выполняются, то для добавления заголовка X-Route: internal заголовок User-Agent не должен содержать "softphone".

Конфигурация будет выглядеть следующим образом:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

vesbc(config-esbc)# mod-table sip MOD_TABLE

#Добавление модификатора с условиями CONDITION_1, CONDITION_2, CONDITION_3, логическим
выражением и действием добавить заголовок X-Route:
vesbc(esbc-mod-table-sip)# mod 0
vesbc(esbc-mod-table-sip-modification)# condition 1 header
vesbc(esbc-mod-table-sip-modification-condition)# name-pattern To
vesbc(esbc-mod-table-sip-modification-condition)# value-pattern ".*sip\\.example\\.com.*"
vesbc(esbc-mod-table-sip-modification-condition)# exit
vesbc(esbc-mod-table-sip-modification)# condition 2 header
vesbc(esbc-mod-table-sip-modification-condition)# name-pattern From
vesbc(esbc-mod-table-sip-modification-condition)# value-pattern ".*123.*"
vesbc(esbc-mod-table-sip-modification-condition)# exit
vesbc(esbc-mod-table-sip-modification)# condition 3 header
vesbc(esbc-mod-table-sip-modification-condition)# name-pattern User-Agent
vesbc(esbc-mod-table-sip-modification-condition)# value-pattern ".*softphone.*"
vesbc(esbc-mod-table-sip-modification-condition)# exit
vesbc(esbc-mod-table-sip-modification)# condition expression "(1&2)|!3"
vesbc(esbc-mod-table-sip-modification)# do commit
vesbc(esbc-mod-table-sip-modification)# do confirm
```

Благодаря настройке **condition expression "(1&2)|!3"** изменена логика применения условий для модификации.

Синтаксис для написания условий

Для написания условий используются регулярные выражения PCRE.

⚠ Регулярное выражение для условия по заголовку или тегу применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать значению "20405". В данном случае шаблон "^20.*" будет корректным, а шаблон "20" — нет.

При вводе выражения в одинарных и двойных кавычках различается логика экранирования символа обратного слеша - «\».

- В одинарных кавычках:

При вводе регулярного выражения в одинарных кавычках символы обратного слеша и двойные кавычки экранируются обратным слешем автоматически.

Пример ввода шаблона:

```
vesbc(config-esbc-mod-table-sip-modification)# response-pattern '\d'
```

Пример отображения шаблона в выводе конфигурации:

```
  mod-table sip MOD
    mod 0 add
      response-pattern "\\d"
    exit
  exit
```

В данном случае будет использоваться поиск совпадения по метасимволу «\d», то есть по совпадению с любой цифрой.

- В двойных кавычках:

При вводе регулярного выражения в двойных кавычках символ обратного слеша используется для экранирования следующего символа, поэтому для использования обратного слеша как символ его нужно дополнительно экранировать обратным слешем.

1. Пример ввода шаблона с одним обратным слешем:

```
vesbc(config-esbc-mod-table-sip-modification)# response-pattern "\d"
```

Пример отображения шаблона в выводе конфигурации:

```
  mod-table sip MOD
    mod 0 add
      response-pattern "d"
    exit
  exit
```

В данном случае будет использоваться поиск совпадения по букве «d».

2. Пример ввода шаблона с двумя обратными слешами:

```
vesbc(config-esbc-mod-table-sip-modification)# response-pattern "\\d"
```

Пример отображения шаблона в выводе конфигурации:

```
  mod-table sip MOD
    mod 0 add
      response-pattern "\\d"
    exit
  exit
```

В данном случае будет использоваться поиск совпадения с использованием метасимвола «\d», то есть по совпадению с любой цифрой.

9.10 Настройка SIP-профилей

В SIP-профиле настраивается конфигурация общих параметров SIP. Профиль используется в транках, транк-группах и абонентских интерфейсах.
В текущей версии ПО поддержаны следующие настройки:

- Контроль доступности направления;
- Список причин отбоя для перехода на следующее направление;
- Поведение при перенаправлении;
- Игнорирование OPTIONS;
- Таймеры SIP-сессий (RFC 4028);
- Транзит сообщений ISUP для работы в режиме SIP-T/SIP-I

Описание всех команд для настройки SIP-профилей приведено в разделе Настройки SIP-профиля.

9.10.1 Контроль доступности направления

Используется для периодической отправки keep-alive сообщений для контроля состояния встречной стороны.

В текущей версии ПО в качестве keep-alive сообщений используется метод OPTIONS.

По умолчанию keep-alive не используется. Для включения необходимо использовать команду *keepalive enable* в SIP-профиле.

Контроль осуществляется путем отправки сообщений OPTIONS с заданными интервалами *success-interval* (по умолчанию 60 сек.) и *failed-interval* (по умолчанию 20 сек.).

Описание всех команд для настройки контроля доступности направления приведено в разделе Настройки SIP-профиля Справочника команд CLI.

Алгоритм работы:

Сообщение OPTIONS отправляется только в случае, когда в транке отсутствует активность SIP после окончания периода success-interval. Т. е. в случае если через транк проходят вызовы с большей частотой, чем указано в настройке success-interval, то сообщения OPTIONS не будут отправляться на встречную сторону, т. к. очевидно, что направление доступно. Если после последнего отправленного или полученного сообщения SIP прошел период равный success-interval, то отправляется OPTIONS. При получении ответа на него (с любым статус-кодом) направление считается доступным. Сообщения OPTIONS будут отправляться с периодом success-interval до того момента, пока либо не появится активность SIP, либо не будут получены ответы на отправленные OPTIONS. Если не будет ответов на OPTIONS, транк считается недоступным, и сообщения OPTIONS будут отправляться с интервалом failed-interval до тех пор, пока транк снова не станет доступным.

Пример настройки:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создать SIP-профиль NEW_SIP_PROFILE:
vesbc(config-esbc)# sip profile NEW_SIP_PROFILE
vesbc(config-esbc-sip-profile)#

#Включить контроль доступности:
vesbc(config-esbc-sip-profile)# keepalive enable
vesbc(config-esbc-sip-profile)#

#Настроить интервалы контроля:
vesbc(config-esbc-sip-profile)# keepalive success-interval 120
vesbc(config-esbc-sip-profile)# keepalive failed-interval 30
vesbc(config-esbc-sip-profile)#

vesbc(config-esbc-sip-profile)# exit
vesbc(config-esbc)#

#Привязать SIP-профиль к транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# sip profile NEW_SIP_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

При использовании SIP-профиля с включенным контролем доступности, для абонентских интерфейсов отправка OPTIONS осуществляться не будет. Данная настройка используется только для контроля транков.

9.10.2 Список причин отбоя для перехода на следующее направление

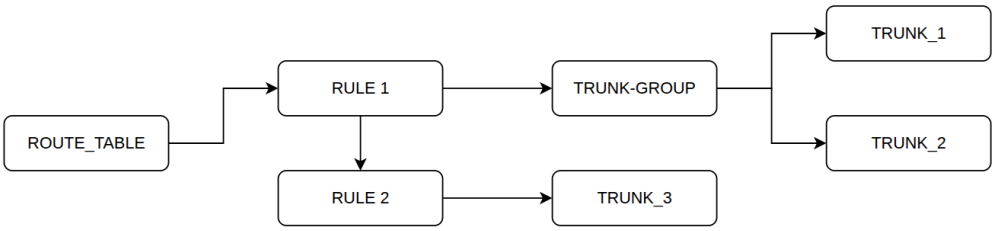
Список причин отбоя для указания статус-кодов ответов SIP, по которым будет осуществляться перемаршрутизация вызовов и регистраций на альтернативное направление (следующий транк в транковой группе/следующее правило в таблице маршрутизации).

При создании маски для списка можно использовать регулярные выражения PCRE.

Регулярное выражение для маски применяется с логикой полного совпадения(exact match). Например, необходимо написать шаблон, который будет соответствовать значению кода "200". В данном случае шаблон "^20.*" будет корректным, а шаблон "20" — нет.

Описание всех команд для настройки причин отбоя для перехода на следующее направление приведено в разделе Настройки SIP-профиля Справочника команд CLI.

Пример использования:



В таблице маршрутизации два правила, первое — направляет вызов в TRUNK_GROUP, второе — направляет вызов в TRUNK_3.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создать список ответов:
vesbc(config-esbc)# cause-list sip LIST
vesbc(config-esbc-cause-list-sip)#

#Создать маску, по которой будут отбираться ответы для перемаршрутизации:
vesbc(config-esbc-cause-list-sip)# cause-mask 404
vesbc(config-esbc-cause-list-sip)# exit

#Создать SIP-профиль, привязать список к SIP-профилю:
vesbc(config-esbc)# sip profile SIP-PROFILE
vesbc(config-esbc-sip-profile)# cause-list LIST
vesbc(config-esbc-sip-profile)# exit

#Привязать к транковой группе TRUNK-GROUP SIP-профиль:
vesbc(config-esbc)# trunk-group TRUNK-GROUP
vesbc(config-esbc-trunk-group)# sip profile SIP-PROFILE
vesbc(config-esbc-trunk-group)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-group)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-group)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Входящий вызов начинает маршрутизироваться по таблице маршрутизации (ROUTE_TABLE). В результате вызов маршрутизируется по правилу RULE_1 на TRUNK_GROUP и оттуда в TRUNK_1. TRUNK_1 недоступен, вызов отбивается по истечении Timer B, и происходит перемаршрутизация на TRUNK_2 (следующий транк в транковой группе). Из TRUNK_2 приходит ответ 404 Not Found, и т. к. код ответа совпадает с маской из списка, который используется в TRUNK-GROUP, то происходит маршрутизация на следующее направление. Поскольку в транковой группе больше нет транков, маршрутизация переходит к RULE_2, и вызов маршрутизируется в TRUNK_3.

Без использования списка причин отбоя, перемаршрутизация происходит только по недоступности транка.

Для абонентских интерфейсов, использование списка причин отбоя не влияет на маршрутизацию. Перемаршрутизация осуществляться не будет.

Перемаршрутизация абонентов

Вызов с зарегистрированного абонента будет направлен в тот транк, через который осуществлялась его регистрация. В случае неуспешного вызова перемаршрутизация осуществляться не будет.
При вызове с незарегистрированного абонента сначала идёт проверка, разрешены ли с этого абонентского интерфейса вызовы без регистрации (allow_unreg_call), если проверка успешна, то вызов смаршрутизируется по привязанной таблице маршрутизации и в случае **недоступности транка/ совпадении ответа с маской из списка** произойдёт маршрутизация на следующее направление.

9.10.3 Поведение при перенаправлении

Настройка поведения при перенаправлении позволяет использовать разные режимы обработки сообщений 3XX.

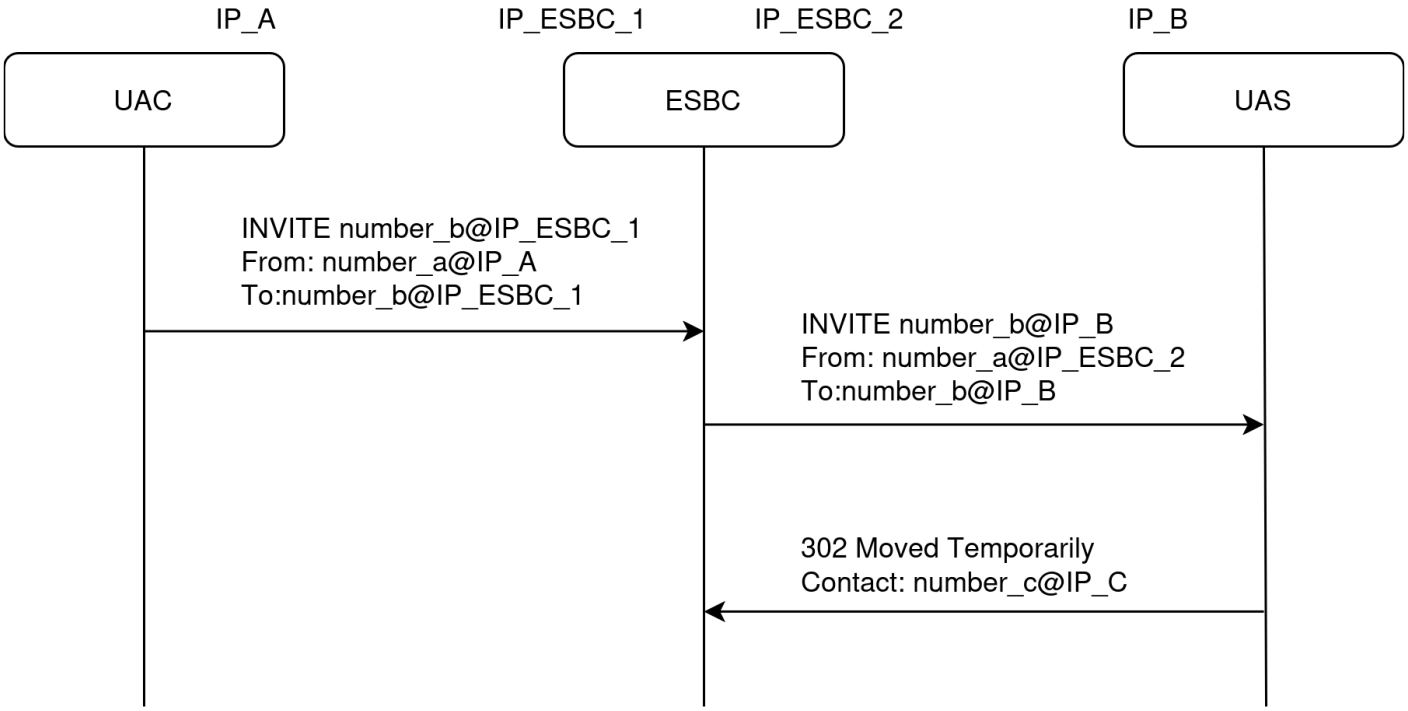
- forbidden — при получении 3xx ответа вызов завершается;
- transit — 3xx передаётся на другое плечо без изменений контакта;
- process — локальная обработка 3xx ответа.

Описание всех команд для настройки поведения при перенаправлении приведено в разделе Настройки SIP-профиля Справочника команд CLI.

Пример локальной обработки 3xx ответа

Схема:

Из транка UAC с IP_A приходит инициирующий INVITE на ESBC с номера number_a на номер number_b. Этот INVITE пересылается на сторону UAS, который может быть транком или абонентским интерфейсом, откуда приходит 302 ответ с номером number_c и адресом IP_C в заголовке Contact.



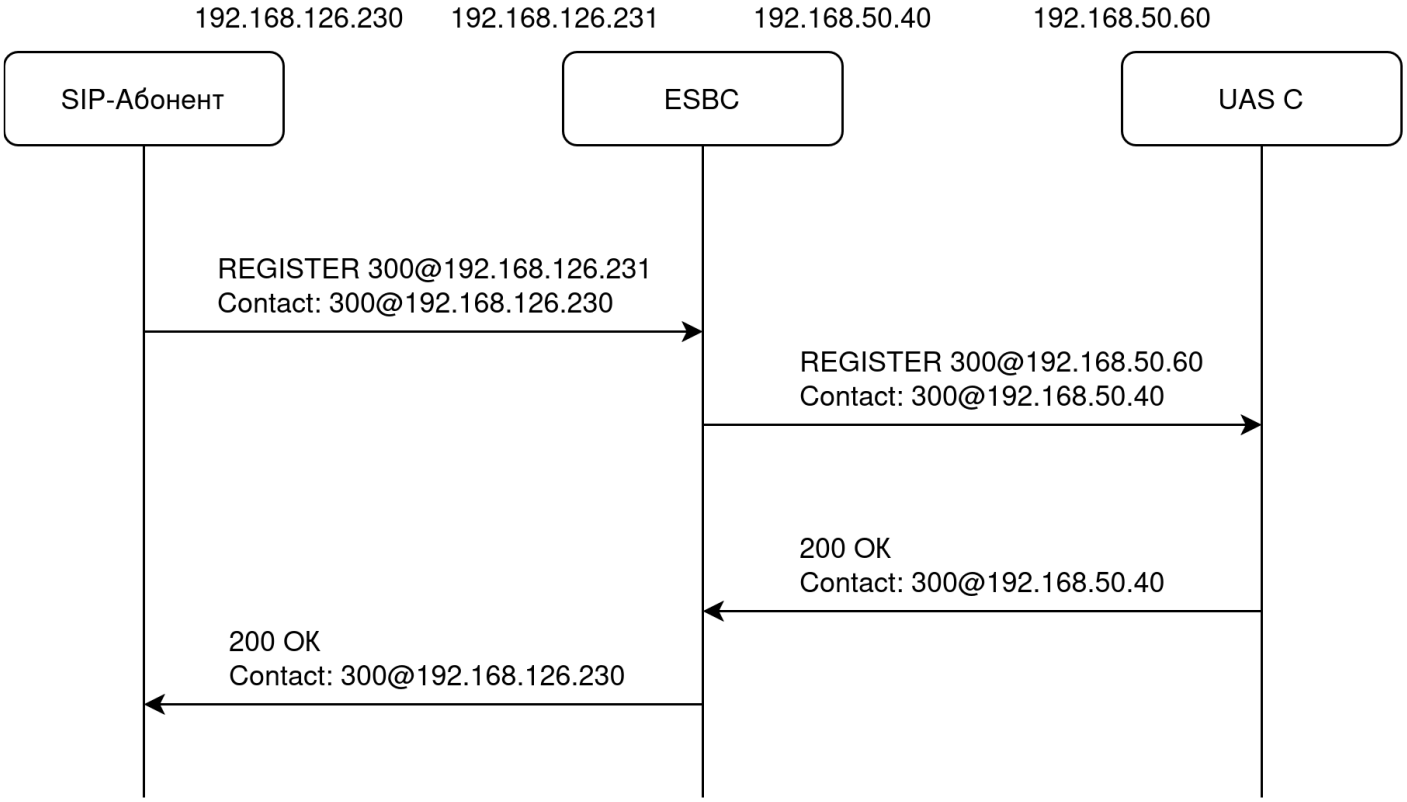
1) **IP_C** соответствует **IP_ESBC_2** и на ESBC зарегистрирован абонент с номером **number_c**.

Если IP-адрес из заголовка Contact пришедшего 302 ответа соответствует IP-адресу транспорта ESBC, с которого был отправлен запрос, и на ESBC существует зарегистрированный абонент с номером,

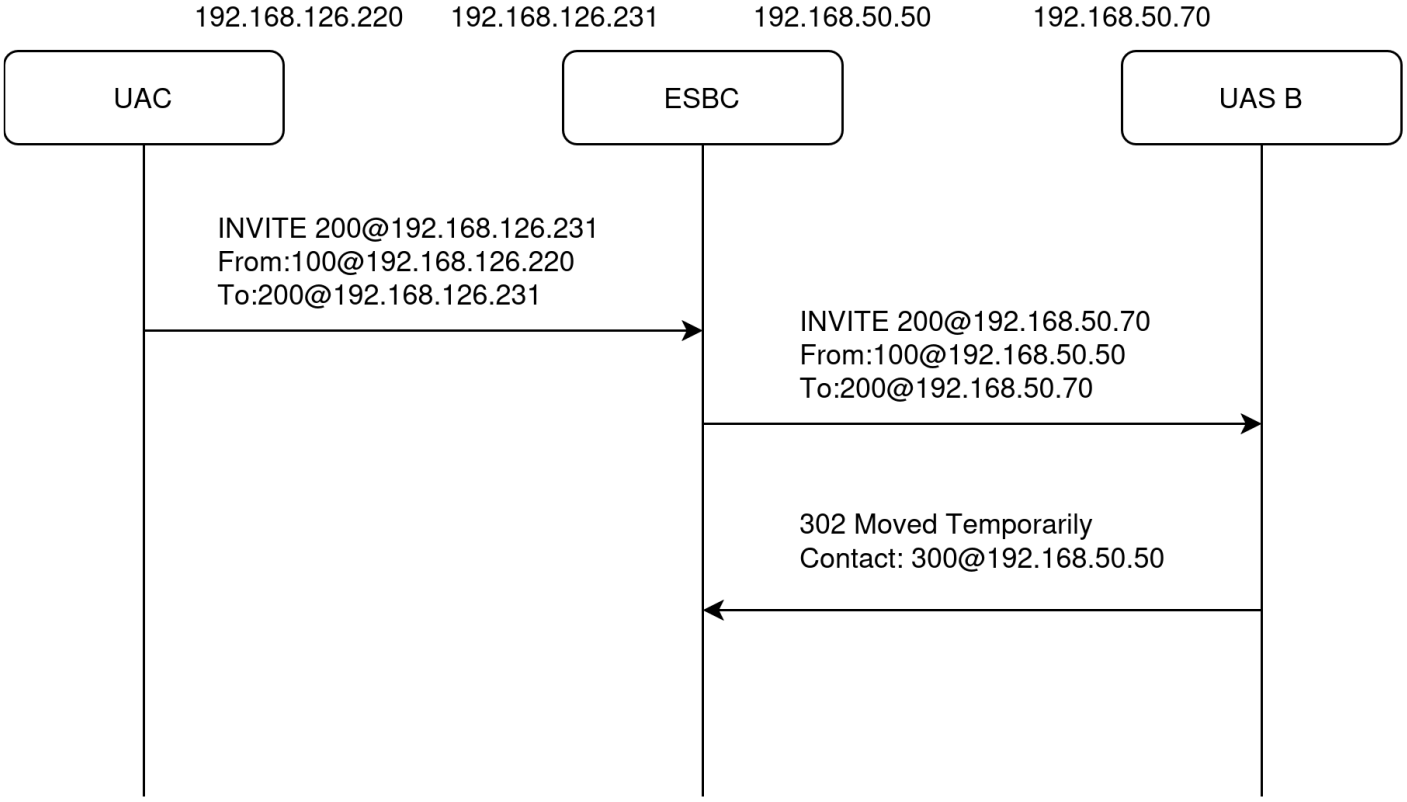
указанным в заголовке Contact, то ESBC отправит INVITE на тот транк, где этот абонент зарегистрирован и 181 в сторону инициатора вызова.

Пример работы:

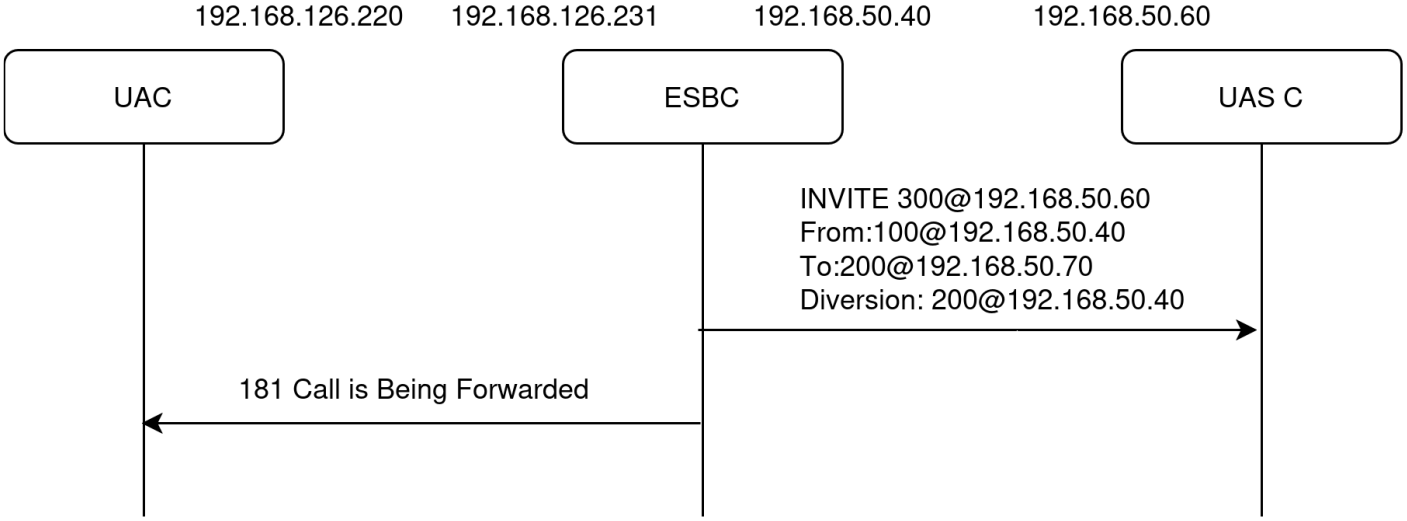
1. Регистрация SIP-абонента с номером 300 на UAS C:



2. С UAC поступает вызов на UAS B, он отвечает 302, в Contact указывает 300@192.168.50.50:



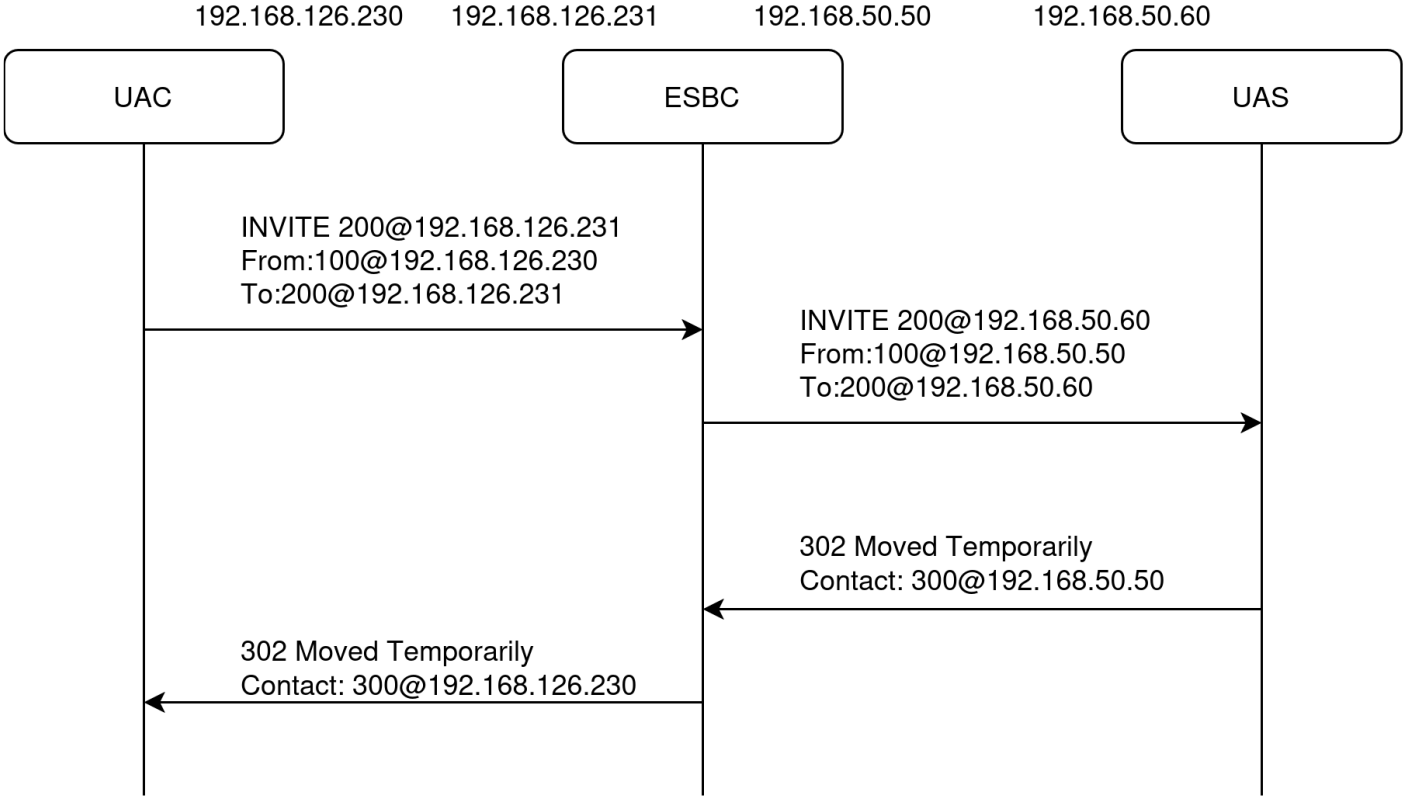
3. Так как существует зарегистрированный абонент с номером 300, то ESBC отправляет INVITE на тот транк, где этот абонент зарегистрирован, то есть на UAS C, а на UAC отправляет 181 ответ:



2) **IP_C** соответствует **IP_ESBC_2** и на ESBC не зарегистрирован абонент с номером **number_c**.

Если IP-адрес из заголовка Contact пришедшего 302 ответа соответствует IP-адресу транспорта ESBC, с которого был отправлен запрос, но на ESBC не существует зарегистрированного абонента с номером, указанным в заголовке Contact, то ESBC перешлёт ответ на другое плечо и заменит адрес в заголовке Contact на IP-адрес UAC.

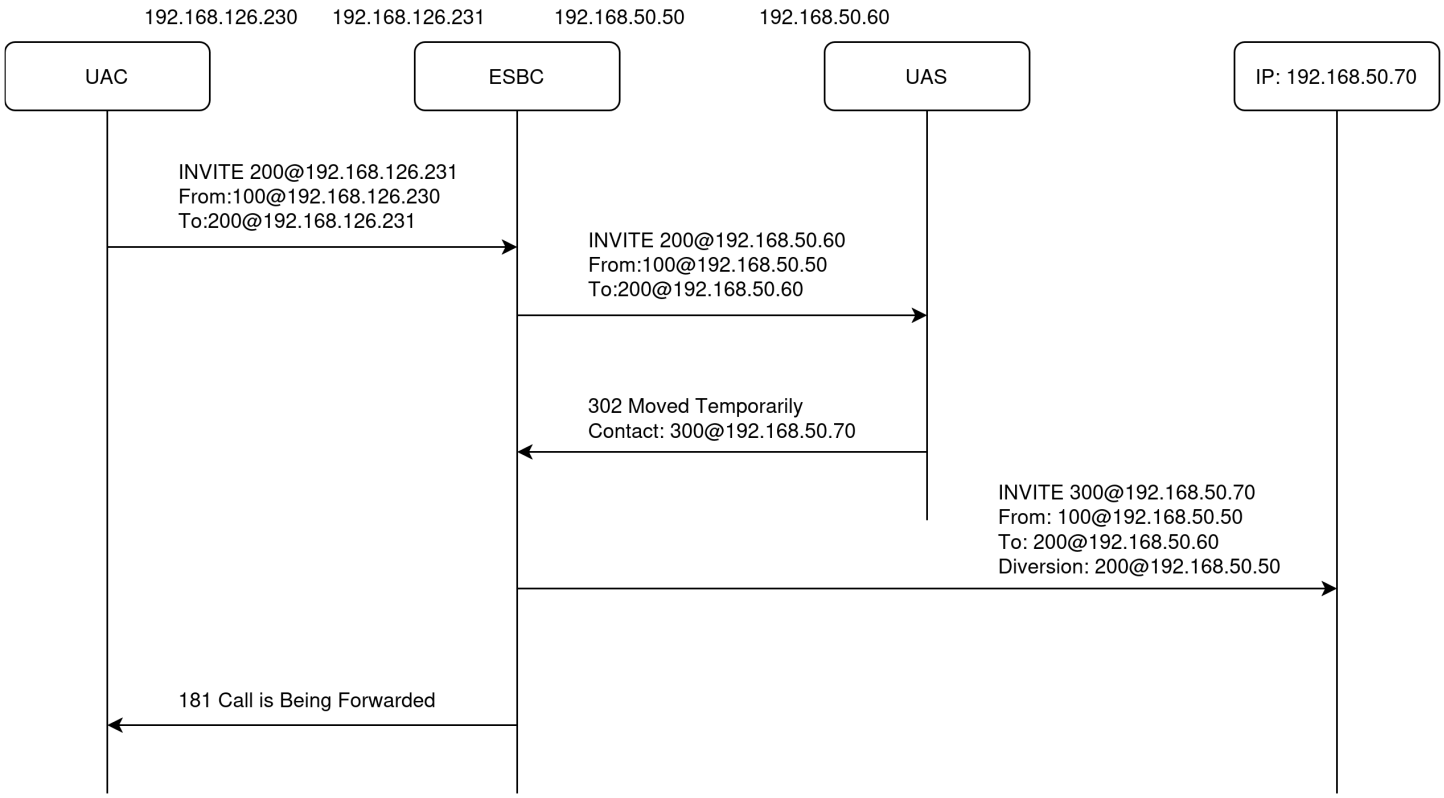
Пример работы:



3) **IP_C** не соответствует **IP_ESBC_2** и UAS — доверенный транк.

Если IP-адрес из заголовка Contact пришедшего 302 ответа не соответствует IP-адресу транспорта ESBC, с которого был отправлен запрос, и транк UAS — доверенный транк, то ESBC отправит INVITE на указанный адрес и 181 в сторону UAC.

Пример работы:



> Для того чтобы транк считался доверенным, необходимо включить опцию **trusted-network** в конфигурации транка.

В прочих случаях — вызов завершается.

9.10.4 Игнорирование OPTIONS

Данный режим используется для обработки входящих сообщений OPTIONS.

- ignore options enable — игнорирование запросов OPTIONS. На входящие запросы OPTIONS не будут отправляться ответы;
- no ignore options enable — отключение игнорирования запросов OPTIONS. На входящие запросы OPTIONS будут отправляться ответы 200 OK.

Описание всех команд для настройки игнорирования OPTIONS приведено в разделе Настройки SIP-профиля Справочника команд CLI.

> **Игнорирование** входящих запросов OPTIONS по умолчанию **включено**.

> Если к user-interface привязан sip profile с включенным игнорированием OPTIONS, то при получении OPTIONS от зарегистрированных абонентов ESBC будет обрабатывать эти запросы и отвечать 200 OK (только если в запросе указан заголовок Contact).
> Прочие входящие запросы OPTIONS будут игнорироваться.

9.10.5 Таймеры SIP-сессий (RFC 4028)

Использование таймеров SIP-сессий (RFC 4028) предназначено для контроля состояния сеансов связи и предотвращения зависания разговорных сессий, в случае возникновения каких-либо проблем, например с сетью.

Обновление сессии поддерживается путем передачи запросов re-INVITE или UPDATE в течение сеанса связи.

Запрашиваемый период контроля сессии (Session Expires) — это период времени в секундах, по истечении которого произойдет принудительное завершение сессии в случае, если сессия не будет вовремя обновлена. Принимает значения от 90 до 64800 с, значение по умолчанию — 1800 с.

Минимальный период контроля сессии (Min SE) — это минимальный интервал проверки работоспособности соединения. Принимает значения от 90 до 32000 с, значение по умолчанию — 90 с. Данное значение не должно превышать таймаут принудительного завершения сессии Sessions expires.

Сторона обновления сессии (Refresher) — определяет сторону, которая будет осуществлять обновление сессии (uac — сторона клиента (вызывающая), uas — сторона сервера (вызываемая)). Значение по умолчанию — uac.

Поведение по умолчанию — поддержка таймеров SIP-сессий отключена. Для включения используется команда *session timer enable*.

Описание всех команд для настройки таймеров SIP-сессий приведено в разделе Настройки SIP-профиля Справочника команд CLI.

Пример настройки

Задача:

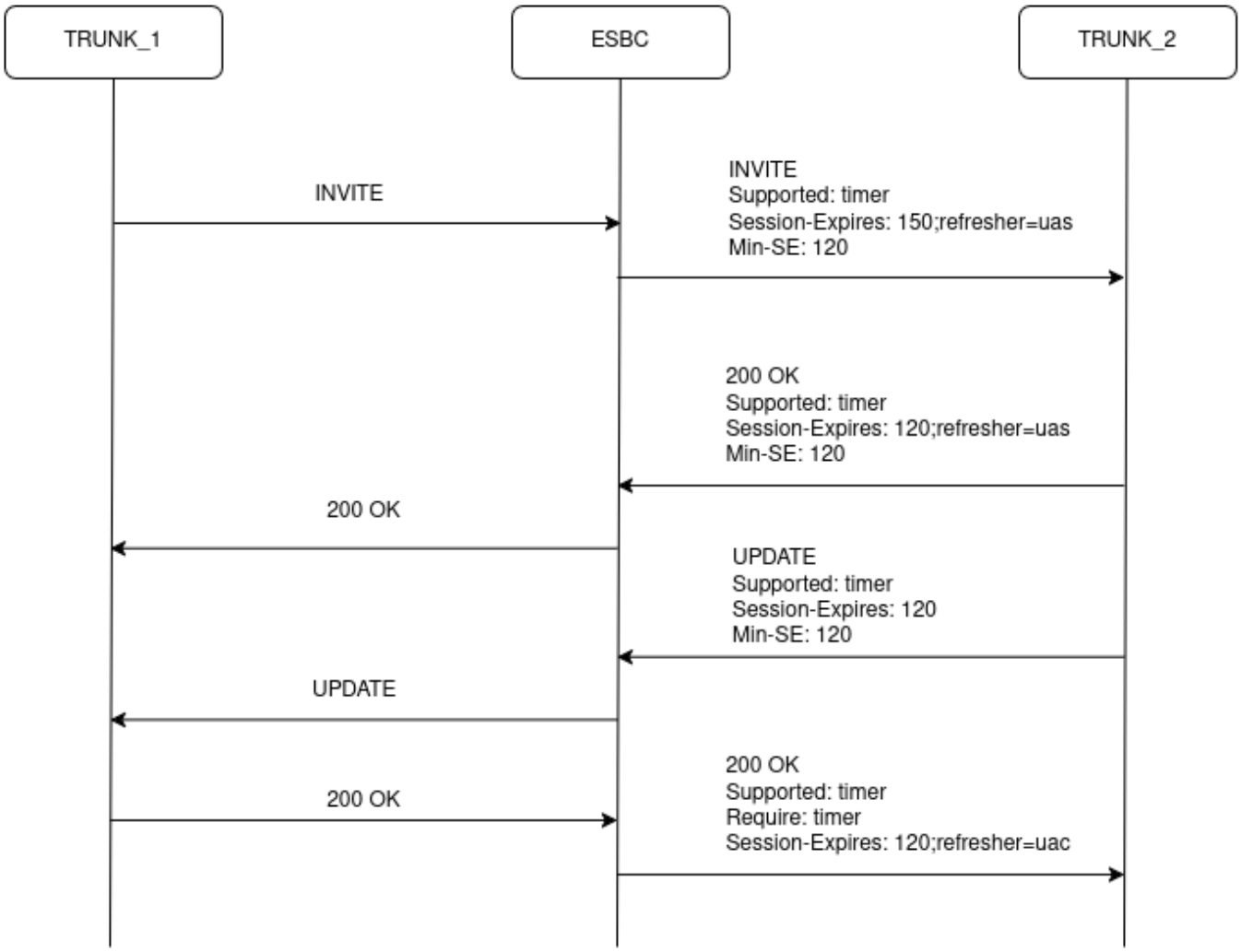
Включить поддержку таймеров SIP-сессий на транке TRUNK_2 с параметрами Session Expires – 120, Min SE – 120, Refresher – UAS. На транке TRUNK_1 таймеры не используются.

Решение:

Настроить SIP-профиль и привязать его к транку TRUNK_2:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создать SIP-профиль NEW_SIP_PROFILE:
vesbc(config-esbc)# sip profile NEW_SIP_PROFILE

#Включить поддержку таймеров SIP-сессий (RFC4028):
vesbc(config-esbc-sip-profile)# session timer enable
vesbc(config-esbc-sip-profile)#

#Настроить минимальный период контроля сессии 120 секунд:
vesbc(config-esbc-sip-profile)# session timer min-se 120
vesbc(config-esbc-sip-profile)#

#Настроить запрашиваемый период контроля сессии 150 секунд:
vesbc(config-esbc-sip-profile)# session timer session-expires 150
vesbc(config-esbc-sip-profile)#

#Настроить сторону обновления сессии:
vesbc(config-esbc-sip-profile)# session timer refresher uas
vesbc(config-esbc-sip-profile)#

vesbc(config-esbc-sip-profile)# exit
vesbc(config-esbc)#

#Привязать SIP-профиль к транку TRUNK_2:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# sip profile NEW_SIP_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

При исходящем вызове в транк TRUNK_2, в сообщение INVITE будут добавлены заголовки в соответствии с настройкой. Если сторона Б также поддерживает RFC 4028, то будет включен контроль сессии:



Т. к. refresher=uas, то в данном примере сторона Б будет отправлять сообщения UPDATE каждые 60 сек. Если по истечении 90 сек. от стороны Б не будет получено сообщение UPDATE, вызов будет разрушен (ESBC отправит BYE в обе стороны).

> ⚠ При использовании таймеров SIP-сессий на одном из плеч вызова, запросы re-INVITE или UPDATE, полученные от удаленной стороны, в рамках контроля сессии будут передаваться на второе плечо, независимо от настроек таймеров на этом плече.

9.10.6 Транзит сообщений ISUP для работы в режиме SIP-T/SIP-I

SIP-I и SIP-T — это два расширения протокола SIP, которые позволяют передавать сообщения телефонной сигнализации ISUP поверх сетей с коммутацией пакетов. Основная функция этих протоколов — инкапсуляция и трансляция сообщений ISUP, что обеспечивает взаимодействие между сетями SIP и традиционными телефонными сетями.

По умолчанию транзит SDP с ISUP запрещен. Для включения используется команда:

```
isup transit
```

Для транзита SDP с ISUP требуется включение данной опции на обоих (входящем и исходящем) направлениях (транках/пользовательских интерфейсах), через которые будет осуществляться маршрутизация сигнализации SIP.

В случае когда на каком-либо из направлений не включена поддержка транзита, на сообщения SIP с вложенным ISUP будет отправлен ответ 415 Unsupported Media Type.

> ⚠ ESBC не обрабатывает ISUP в SDP, а только осуществляет транзит данного вложения.

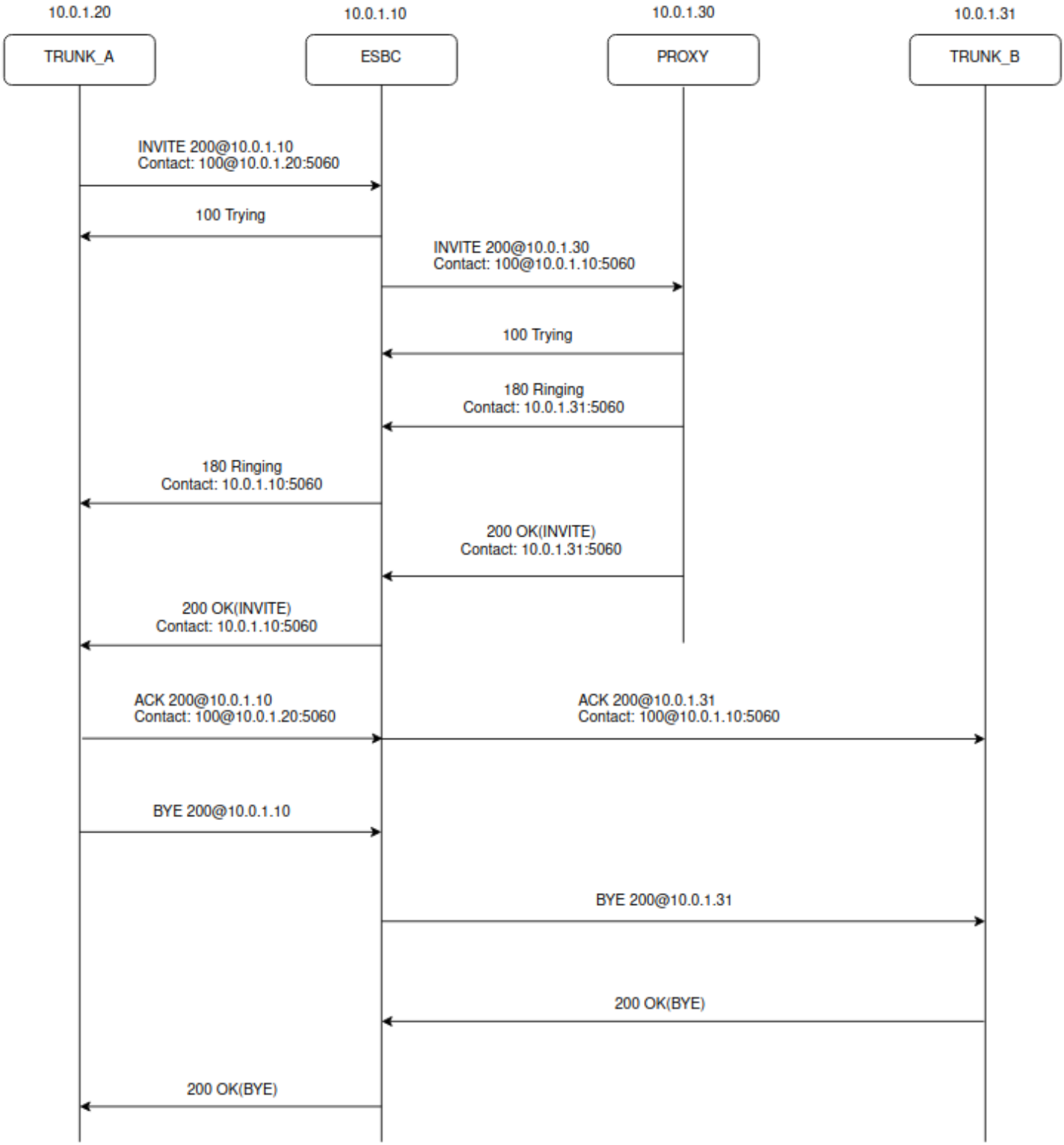
Описание всех команд для настройки транзита сообщений ISUP приведено в разделе Настройки SIP-профиля Справочника команд CLI.

9.10.7 Опционирование обработки заголовка Contact в запросах и ответах SIP

По умолчанию поведение ESBC реализовано в соответствии с п. 4 RFC 3261 (SIP session setup example with SIP trapezoid), поэтому при установлении диалога, запрос ACK будет направлен не на IP-адрес и порт, настроенные в транке (remote address, remote port), а на полученные из заголовка Contact встречной стороны.

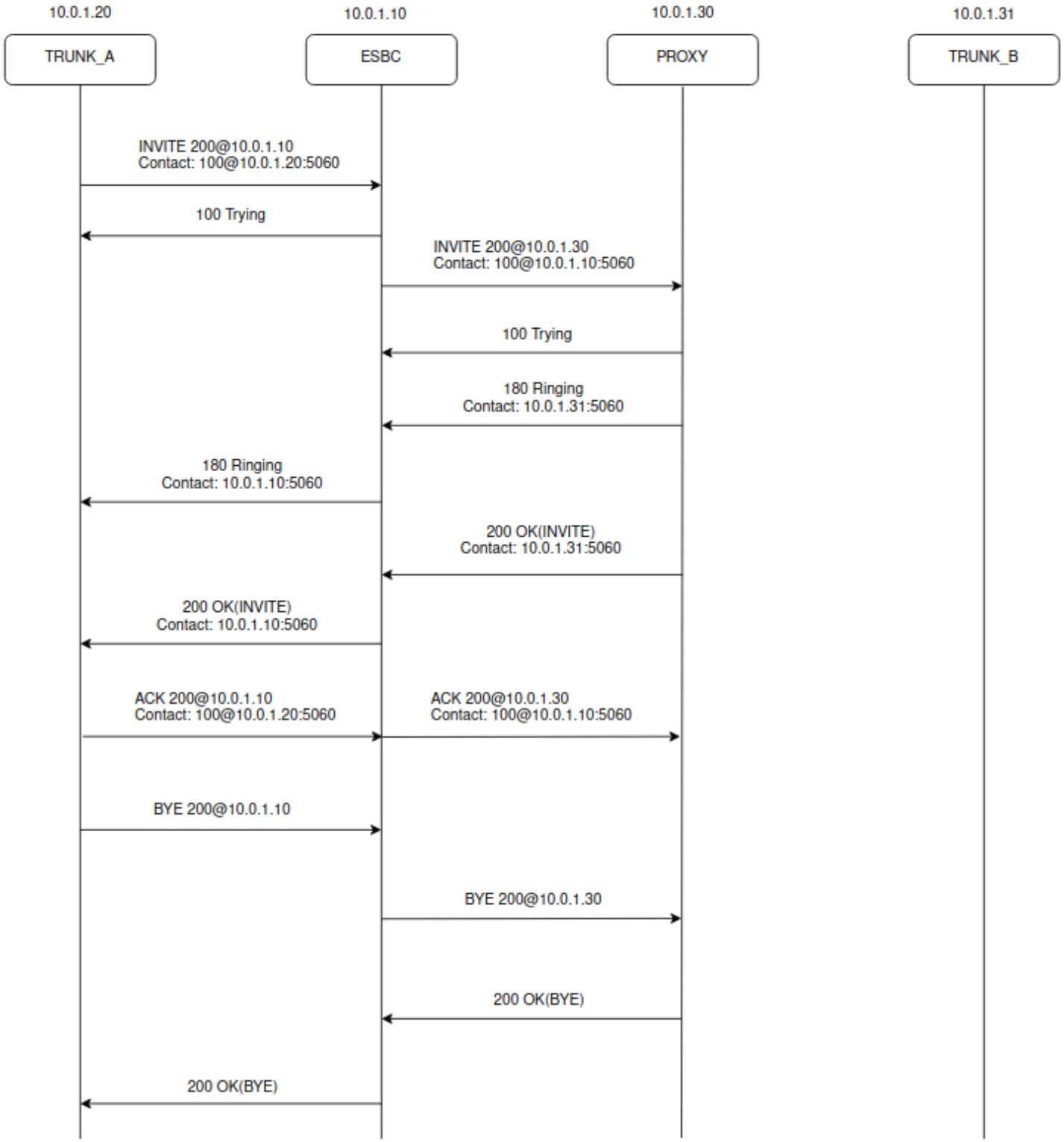
Пример

Вызов из транка А в транк Б, между ESBC и транком Б находится SIP-прокси, ESBC отправляет INVITE на адрес SIP-прокси (10.0.1.30), в 180 и 200 ответе в заголовке Contact SIP-прокси отправляет адрес транка Б (10.0.1.31). При отключенной опции (поведение по умолчанию) ESBC будет отправлять последующие запросы на адрес и порт, указанные в ответах в заголовке Contact.



Если включить опцию **contact follow disable** в SIP-профиле, то изменение адреса и порта в заголовке Contact будет игнорироваться и запросы будут отправляться на адрес и порт, настроенные в конфигурации транка.

Пример диаграммы обмена сообщениями с включенной опцией на плече ESBC--PROXY(TRUNK_B):



⚠ Если на абонентском интерфейсе включен NAT comedia mode, то вне зависимости от настроек в SIP-профиле, запросы будут отправляться на реальный адрес и порт удалённой стороны, значения в заголовке Contact игнорируются.

9.11 Настройка медиапрофилей

Использование медиапрофилей позволяет гибко управлять типом медиаданных путем фильтрации медиасекций в SDP, транскодированием аудио и видео, шифрованием RTP-потока, контролем сессии по наличию RTP и RTCP-пакетов.

Медиапрофили используются в абонентских интерфейсах, транках и транк-группах. Медиапрофиль, используемый для транка, входящего в транк-группу, переопределяет настройки медиапрофиля, используемого в транк-группе.

9.11.1 Управление типом медиаданных и кодеками

Обработка медиапотоков осуществляется в двух режимах: проксирование и транскодирование.

По умолчанию ESBC работает в режиме проксирования медиатрафика без использования транскодирования. Список шаблонов кодеков, доступных для проксирования через ESBC, задается командой:

```
codec allow {all | <CODEC_PATTERN> [<PT>]}
```

<CODEC_PATTERN> — название кодека/часть названия кодека;

<PT> — payload type (необязательный параметр). При указании будет проводиться дополнительная проверка шаблона на полное совпадение кодека с указанным payload type.

Описание всех команд приведено в разделе Настройки медиапрофиля справочника команд CLI.

При создании медиапрофиля список шаблонов для наиболее известных кодеков IANA, доступных для проксирования, добавляется автоматически и выглядит следующим образом:

```
media profile MEDIA_PROFILE
  codec allow AMR
  codec allow CLEARMODE
  codec allow CN
  codec allow G72
  codec allow G722/ 9
  codec allow G723 4
  codec allow G728 15
  codec allow G729/ 18
  codec allow GSM 3
  codec allow H26
  codec allow H261 31
  codec allow H263 34
  codec allow ILBC
  codec allow L16/44100 11
  codec allow L16/44100/2 10
  codec allow OPUS
  codec allow PCMA 8
  codec allow PCMU 0
  codec allow SPEEX
  codec allow T38 t38
  codec allow VP
  codec allow telephone-event
exit
```

Т. к. использование шаблонов позволяет указывать не полное название кодека, а его часть, то запись вида "codec allow G72" означает, что кодеки G726-16, G726-24, G726-32, G726-40 будут доступны для проксирования.

Для кодеков со статическим payload type рекомендуется указывать номер payload type, иначе, если в SDP не будет указан атрибут rtpmap, вызов будет отбиваться кодом 488.

Для абонентских интерфейсов, транков и транковых групп, к которым не привязан ни один медиапрофиль, используется медиапрофиль по умолчанию, который не отображается в конфигурации. В данном медиапрофиле применяются шаблоны кодеков, доступных для проксирования, указанные выше.

Для очистки списка используется команда *no codec allow all*. При использовании данной команды будут удалены шаблоны кодеков, добавленные автоматически при создании профиля, и шаблоны кодеков, добавленные/измененные пользователем.

Управление списком кодеков и типом медиаданных (audio, video, image) SDP осуществляется путем добавления/удаления/изменения шаблонов codec allow. Максимальное количество шаблонов в одном медиапрофиле — 64.

Для успешного согласования кодеков в режиме проксирования, необходимо, чтобы на входящем и исходящем направлении в медиапрофилях, привязанным к этим направлениям, содержались шаблоны, позволяющие пропускать одни и те же кодеки. В случае когда согласование невозможно, на запросы INVITE ESBC будет отвечать сообщением 488.

Примеры использования медиапрофиля для управления кодеками и типами медиаданных в режиме проксирования

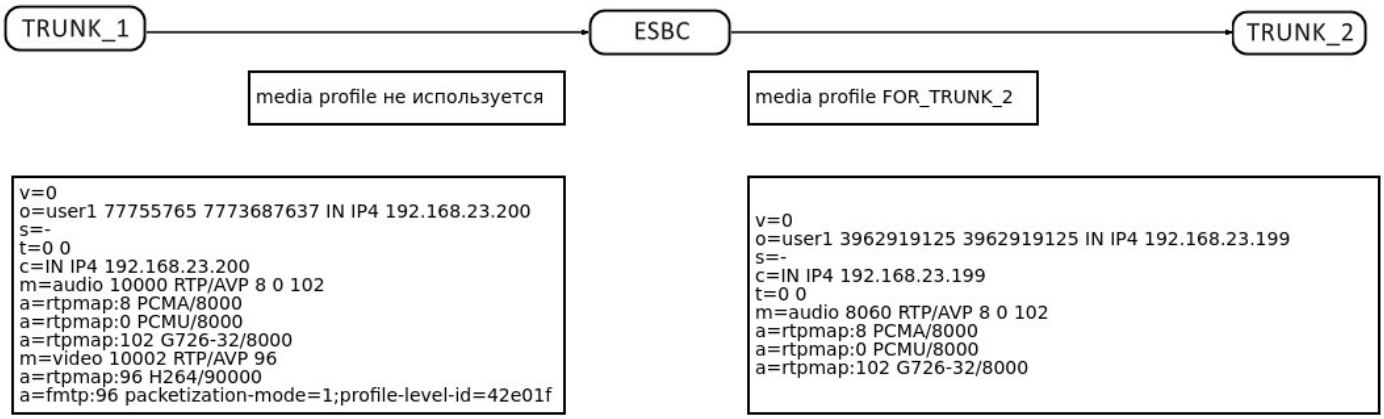
1. Запретить использование видео для транка TRUNK_2.

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для транка TRUNK_2:
vesbc(config-esbc)# media profile FOR_TRUNK_2

#Запретить использование всех видеокодеков:
vesbc(config-esbc-media-profile)# no codec allow H26
vesbc(config-esbc-media-profile)# no codec allow H261
vesbc(config-esbc-media-profile)# no codec allow H263
vesbc(config-esbc-media-profile)# no codec allow VP
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_2
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

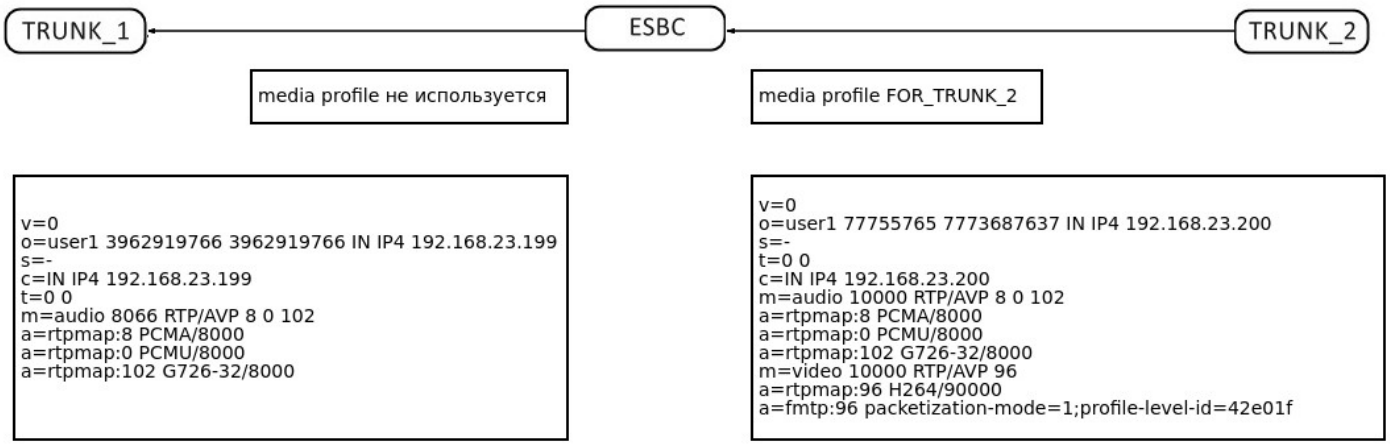
В результате конфигурация медиапрофиля будет выглядеть следующим образом:

```
  media profile FOR_TRUNK_2
    codec allow AMR
    codec allow CLEARMODE
    codec allow CN
    codec allow G72
    codec allow G722/ 9
    codec allow G728 15
    codec allow G729/ 18
    codec allow GSM 3
    codec allow ILBC
    codec allow L16/44100 11
    codec allow L16/44100/2 10
    codec allow OPUS
    codec allow PCMA 8
    codec allow PCMU 0
    codec allow SPEEX
    codec allow T38 t38
    codec allow telephone-event
  exit
```

В данном примере для транка TRUNK_1 не требуется использование отдельного медиапрофиля, т. к. при вызовах, поступающих в TRUNK_1, и, маршрутизируемых в TRUNK_2, все видеокодеки из SDP будут удалены в соответствии с медиапрофилем, используемым для транка TRUNK_2.



Для вызовов, поступающих в TRUNK_2, все видеокодеки из SDP будут удалены вне зависимости от направления маршрутизации.

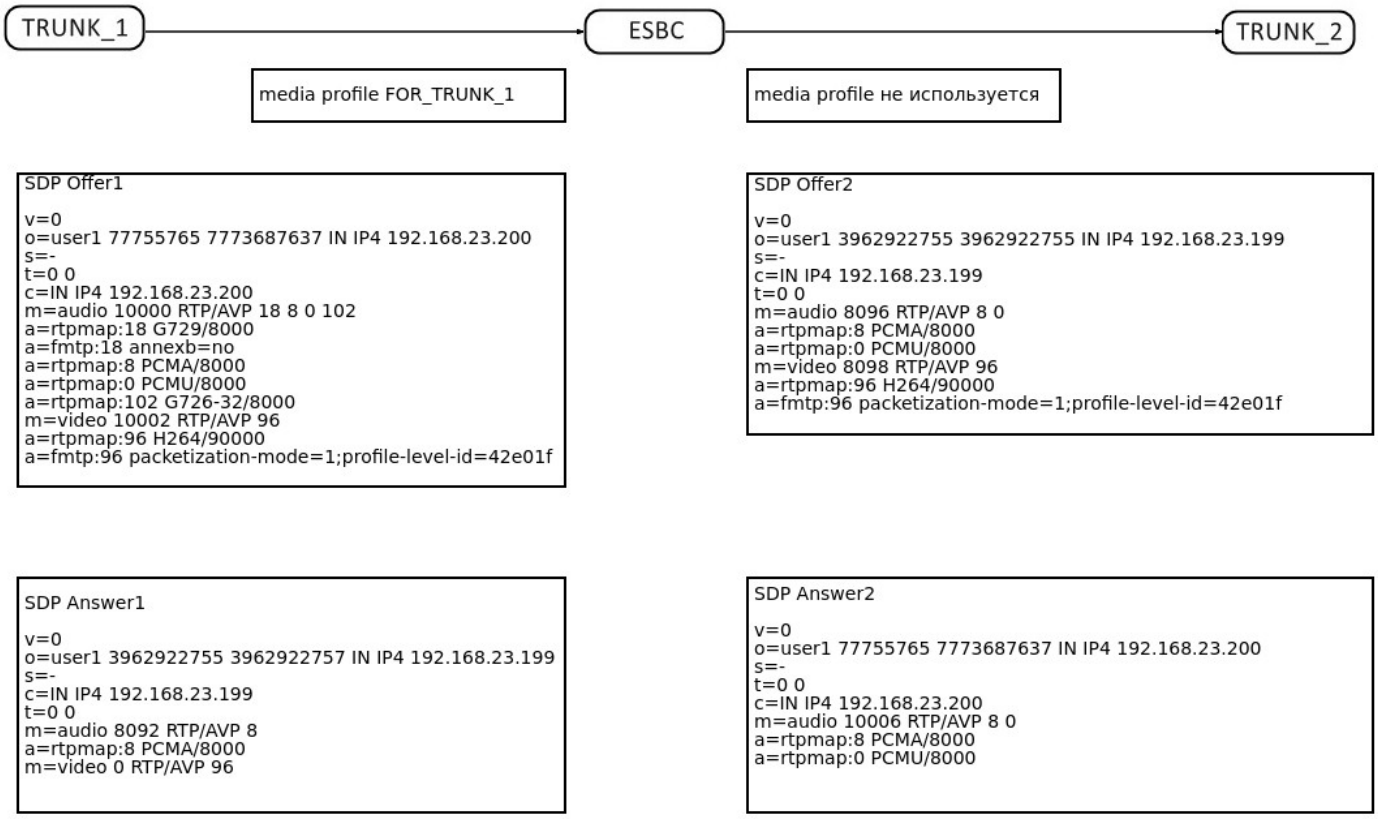


2. Запретить использование кодеков G729 и G726 для транка TRUNK_1.

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для транка TRUNK_1:
vesbc(config-esbc)# media profile FOR_TRUNK_1

#Запретить использование кодеков G729 и G726:
vesbc(config-esbc-media-profile)# no codec allow G729/
vesbc(config-esbc-media-profile)# no codec allow G72
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_1
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

В результате конфигурация медиапрофиля будет выглядеть следующим образом:

```
media profile FOR_TRUNK_1
  codec allow AMR
  codec allow CLEARMODE
  codec allow CN
  codec allow G722/ 9
  codec allow G728 15
  codec allow GSM 3
  codec allow H26
  codec allow H261 31
  codec allow H263 34
  codec allow ILBC
  codec allow L16/44100 11
  codec allow L16/44100/2 10
  codec allow OPUS
  codec allow PCMA 8
  codec allow PCMU 0
  codec allow SPEEX
  codec allow T38 t38
  codec allow VP
  codec allow telephone-event
exit
```



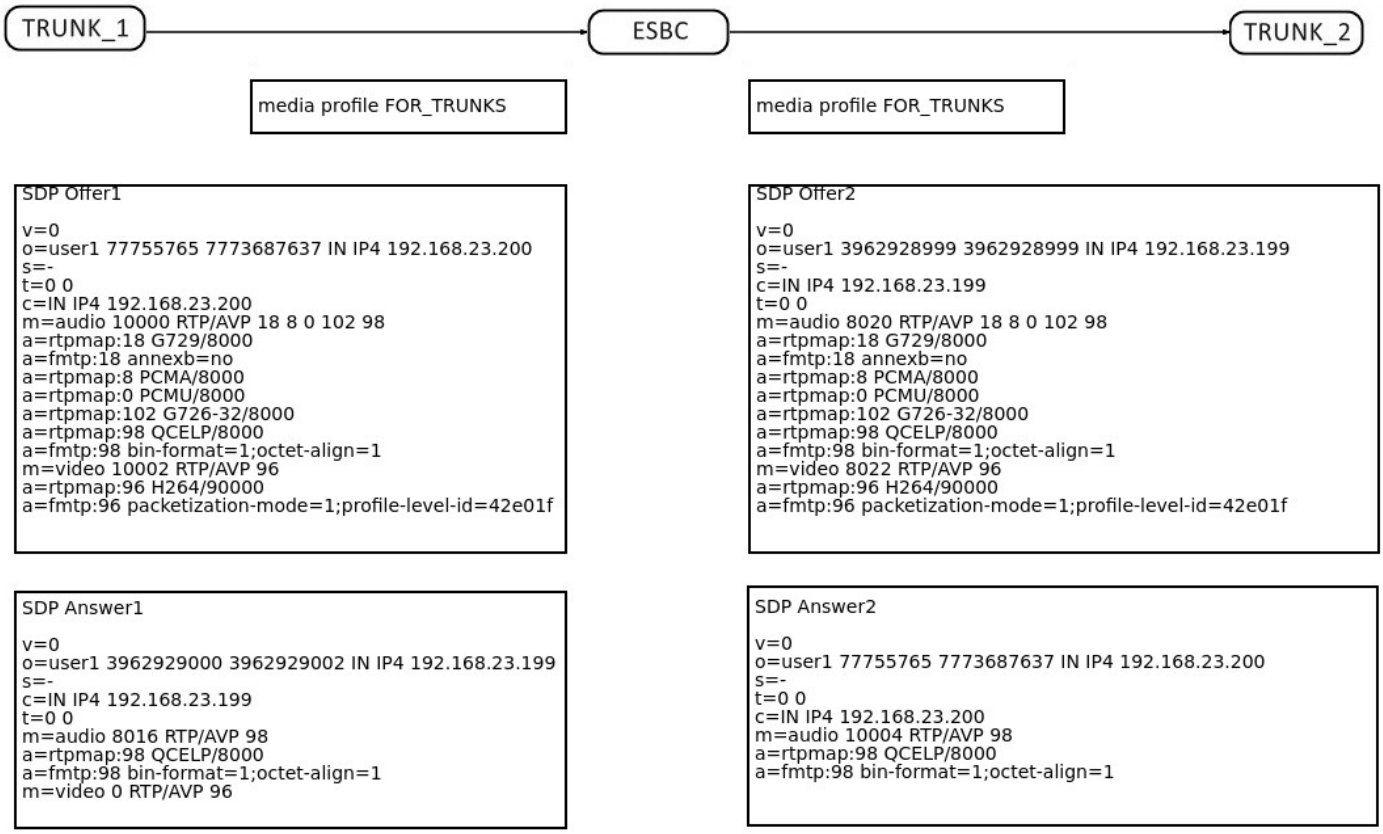
В данном примере в транк TRUNK_1 приходит INVITE с SDP Offer1, в котором наиболее приоритетным кодеком является G729, а также указан кодек G726, но т. к. настройками медиапрофиля FOR_TRUNK_1 данные кодеки запрещены, то в транк TRUNK_2 будет отправлен SDP Offer2 без данных кодеков. UA TRUNK_2 выбирает в качестве приоритетного кодек PCMA (SDP Answer2), и в результате ESBC отправляет в SDP Answer1 наиболее приоритетный кодек из SDP Offer1 (кроме G729) — PCMA.

3. Разрешить использование кодека QCELP для обоих транков (в дополнение к шаблонам по умолчанию).

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для использования в обоих транках:
vesbc(config-esbc)# media profile FOR_TRUNKS

#Добавить шаблон для кодека QCELP:
vesbc(config-esbc-media-profile)#codec allow QCELP
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к обоим транкам:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNKS
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNKS
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# do commit
vesbc(config-esbc)# do confirm
```

В результате конфигурация медиапрофиля будет выглядеть следующим образом:

```
 media profile FOR_TRUNKS
codec allow AMR
codec allow CLEARMODE
codec allow CN
codec allow G72
codec allow G722/ 9
codec allow G728 15
codec allow G729/ 18
codec allow GSM 3
codec allow H26
codec allow H261 31
codec allow H263 34
codec allow ILBC
codec allow L16/44100 11
codec allow L16/44100/2 10
codec allow OPUS
codec allow PCMA 8
codec allow PCMU 0
codec allow QCELP
codec allow SPEEX
codec allow T38 t38
codec allow VP
codec allow telephone-event
exit
```



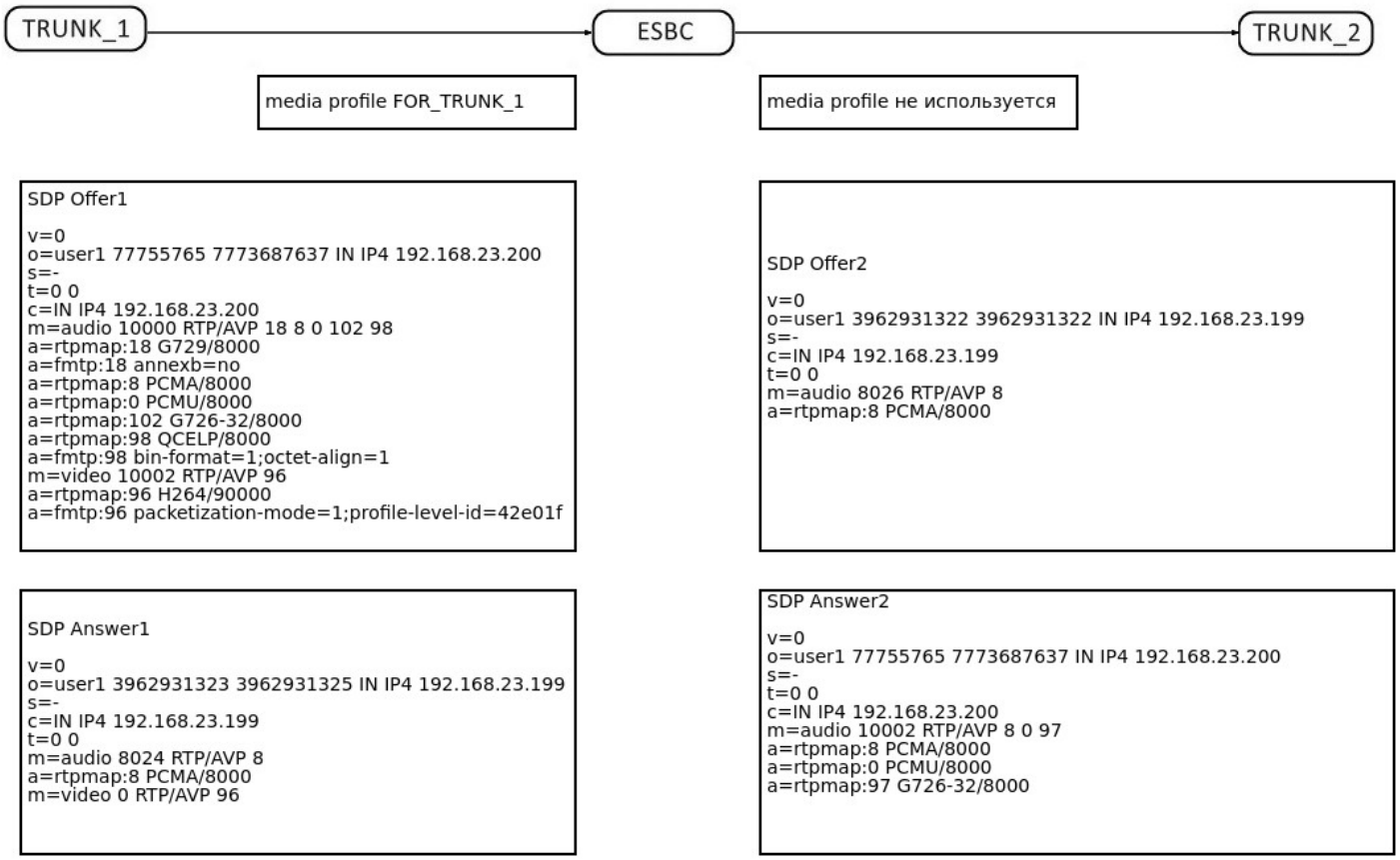
В данном примере в транк TRUNK_1 приходит INVITE с SDP Offer1, в котором содержится кодек QCELP, и т. к. настройками медиапрофиля FOR_TRUNKS этот кодек разрешен, то он будет передаваться SDP Offer2, отправляемый в транк TRUNK_2. UA TRUNK_2 выбирает кодек QCELP, и в результате он будет согласован в SDP Answer1.

4. Разрешить использование только кодека PCMA для TRUNK_1.

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для транка TRUNK_1:
vesbc(config-esbc)# media profile FOR_TRUNK_1

#Удалить все шаблоны:
vesbc(config-esbc-media-profile)# no codec allow all

#Добавить шаблон только для кодека PCMA:
vesbc(config-esbc-media-profile)# codec allow PCMA 8
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_1
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

В результате конфигурация медиапрофиля будет выглядеть следующим образом:

```
media profile FOR_TRUNK_1
  codec allow PCMA 8
exit
```



В данном примере в транк TRUNK_1 приходит INVITE с SDP Offer1 с набором кодеков. Т. к. настройками медиапрофиля FOR_TRUNK_1 запрещены все кодеки кроме PCMA, то в транк TRUNK_2 будет отправлен SDP Offer2, содержащий только кодек PCMA.

9.11.2 Транскодирование

Транскодирование — это реализация преобразования медиапотоков, основанных на разных кодеках.

Эта реализация позволяет:

- гибко справляться со сложными сетевыми соединениями и широким спектром медиакодеков;
- оптимизировать доступную полосу пропускания, принудительно используя различные кодеки сжатия;
- нормализовать трафик в сети предприятия, используя один кодек;
- заключать соглашения о взаимодействии между сетями VoIP для использования одобренных кодеков.

ESBC предоставляет реализацию транскодирования на границе сети вместо использования ресурсов сети предприятия для тех же функций.

Список кодеков, поддерживаемых в режиме транскодирования:

Аудиокодеки	Видеокодеки
AMR	H263-1998
AMR-WB	H264
G722	VP8
G7221-24	VP9
G7221-32	
G7221C-24	
G7221C-32	
G7221C-48	
G726-16	
G726-24	
G726-32	
G726-40	
G729	
GSM	
ILBC	
L16-MONO	
OPUS	
PCMA	
PCMU	
SPEEX-NB	
SPEEX-UWB	
SPEEX-WB	

Поддержка кодеков для транскодирования осуществляется командами:

```
codec {audio | video | image} {all | <CODEC>}
no codec {audio | video | image} {all | <CODEC>}
```

<CODEC> — название кодека. Указывается из списка поддержанных для транскодирования кодеков.

all — включает транскодирование всех доступных кодеков заданного типа медиаданных.

> Команда *codec image* в текущей версии ПО не поддерживается, данная команда аналогична команде *codec allow T38 t38*.

Описание всех команд приведено в разделе Настройки медиапрофиля справочника команд CLI.

Порядок обработки SDP для выбора режима работы:

1. Offer SDP фильтруется согласно разрешённым кодекам на плече A.
2. Offer SDP фильтруется согласно разрешённым кодекам на плече B.
3. Если в медиапрофиле на плече A включен транскодинг, и во входящем SDP присутствуют кодеки из списка разрешенных, то в конец Offer SDP добавляются недостающие кодеки, транскодинг которых включен в media profile на плече B.
4. Answer SDP фильтруется согласно разрешённым кодекам на плече B.
5. В конец Answer SDP добавляются недостающие кодеки, транскодинг которых включен в media profile на плече A.
6. Перед отправкой Answer SDP в плечо A производится согласование кодеков.

В результате транскодирование включается, если самые приоритетные кодеки из Offer и Answer SDP на двух плечах не совпадают. В таком случае в Answer SDP будет выбран наиболее приоритетный кодек, который был получен в Offer SDP, и для которого включена поддержка транскодирования в медиапрофиле на плече A.

Иначе при совпадении приоритетных кодеков в Offer SDP и Answer SDP, будет использоваться проксирование медиаданных.

С целью снижения нагрузки на ESBC, транскодирование включается только в случае, когда использовать проксирование медиатрафика невозможно.

Для включения транскодирования необходимо использовать медиапрофили с включенным транскодированием (codec audio/video) на обоих направлениях.

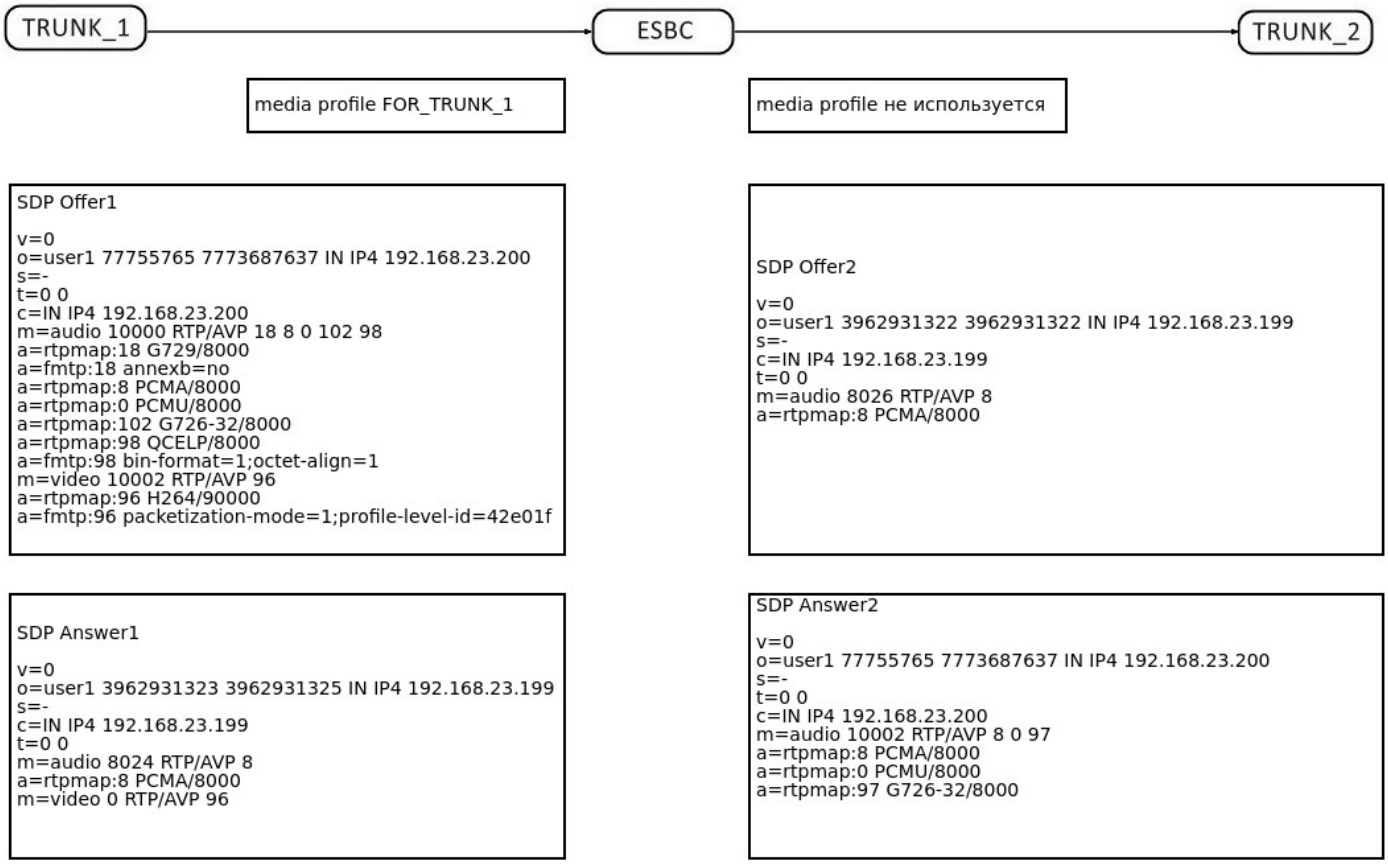
Если на одном из направлений не используется медиапрофиль (т. е. используется медиапрофиль по умолчанию), или в профиле не настроено ни одно правило codec audio/video, то транскодирование осуществляться не будет.

Пример:

В транке TRUNK_1 используется медиапрофиль FOR_TRUNK_1, в котором разрешены кодеки PCMA и PCMU для проксирования, и не указаны кодеки, разрешенные для транскодирования.

В транке TRUNK_2 используется медиапрофиль FOR_TRUNK_2, в котором также кодеки PCMA и PCMU разрешены для проксирования, и кодек G729 разрешен для транскодирования.

В SDP Offer1, полученном с транка TRUNK_1, указаны кодеки PCMA и PCMU, и т. к. в медиапрофиле FOR_TRUNK_1 отсутствуют кодеки, разрешенные для транскодирования, в SDP Offer2, который будет отправлен в TRUNK_2, кодек G729 не будет добавлен. Соответственно при вызовах из TRUNK_1 в TRUNK_2 (и в обратном направлении) возможно только проксирование медиаданных.



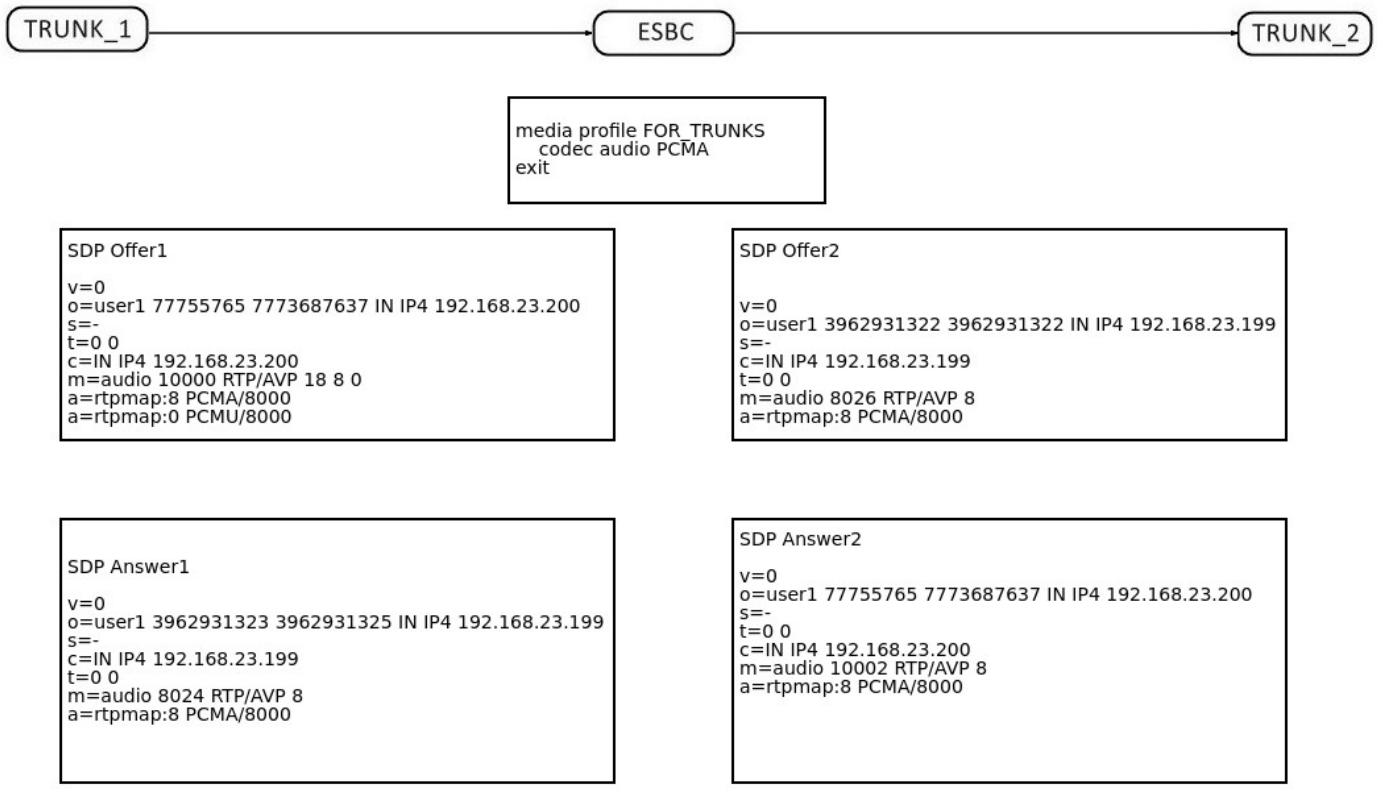
Если в медиапрофиле не содержится ни одного шаблона для проксирования кодеков, а указаны только кодеки, доступные для транскодирования, то при наличии одинаковых кодеков в медиапрофилях, используемых на входящем и исходящем направлениях, медиаданные будут передаваться в режиме проксирования.

Т. о. включение поддержки транскодирования для кодеков командами codec {audio | video | image} {all | <CODEC>} не означает, что передаваемые через ESBC медиаданные всегда будут транскодироваться.

Пример:

Для транков TRUNK_1 и TRUNK_2 используется один и тот же медиапрофиль FOR_TRUNKS, в котором указан только кодек PCMA, разрешенный для транскодирования, и отсутствуют шаблоны кодеков для проксирования.

В SDP Offer1, полученном с транка TRUNK_1, указаны кодеки PCMA и PCMU, и т. к. в медиапрофиле FOR_TRUNKS не указан кодек PCMU (ни для проксирования, ни для транскодирования), то в SDP Offer2, который будет отправлен в TRUNK_2, кодек PCMU не будет добавлен. При получении SDP Answer2 происходит согласование кодека PCMA, и в TRUNK_1 будет отправлен SDP Answer1 с кодеком PCMA. При этом медиаданные будут передаваться в режиме проксирования, т. к. наиболее приоритетные кодеки в SDP Offer и SDP Answer совпадают.



Примеры использования медиапрофилей для управления кодеками в режиме транскодирования

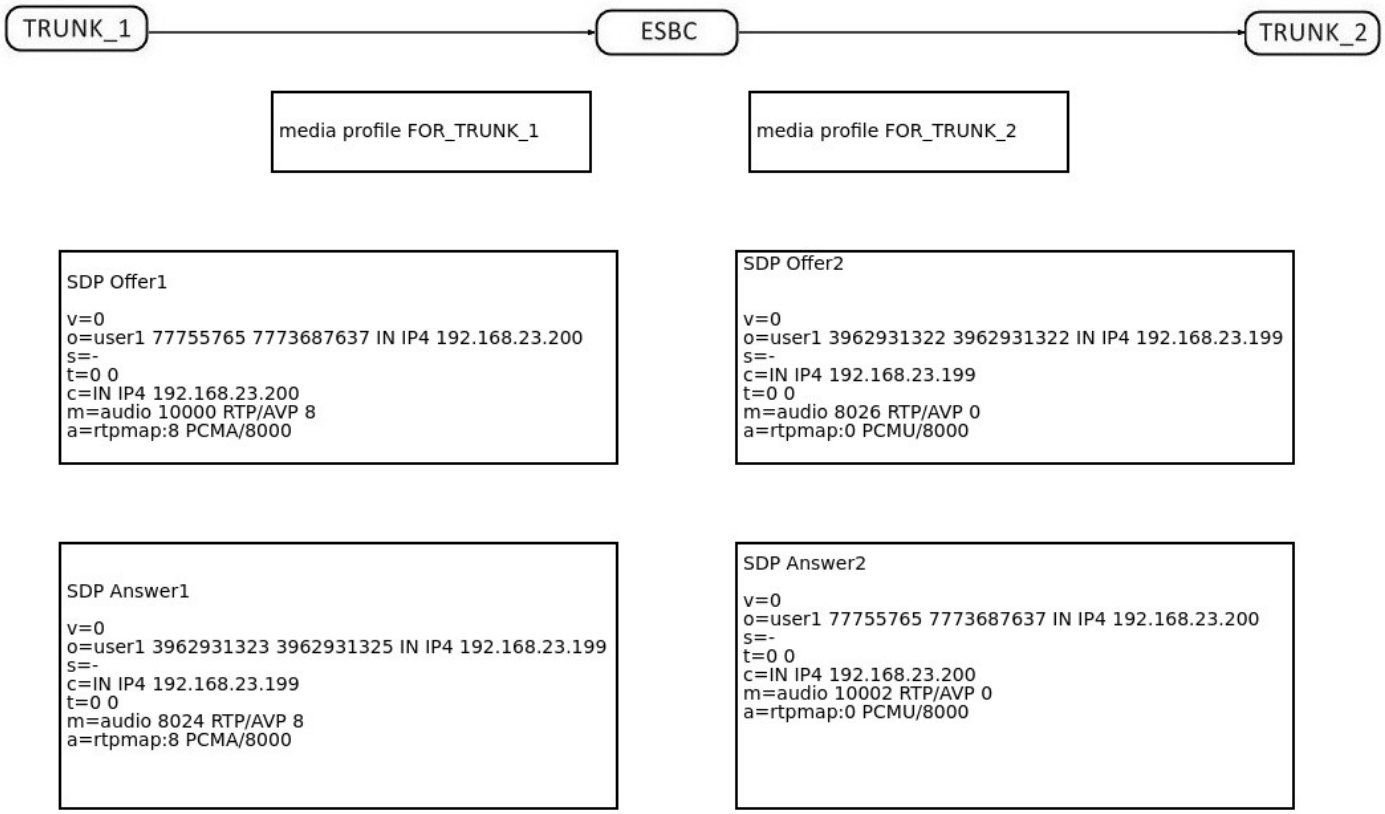
1. Настроить только режим транскодирования кодеков PCMA <---> PCMU между направлениями.

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для транка TRUNK_1:
vesbc(config-esbc)# media profile FOR_TRUNK_1

#Запретить использование всех кодеков в режиме проксирования:
vesbc(config-esbc-media-profile)# no codec allow all

#Включить поддержку кодека PCMA в режиме транскодирования:
vesbc(config-esbc-media-profile)# codec audio PCMA
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку TRUNK_1:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_1
vesbc(config-esbc-trunk-sip)# exit

#Создать медиапрофиль для транка TRUNK_2:
vesbc(config-esbc)# media profile FOR_TRUNK_2

#Запретить использование всех кодеков в режиме проксирования:
vesbc(config-esbc-media-profile)# no codec allow all

#Включить поддержку кодека PCMU в режиме транскодирования:
vesbc(config-esbc-media-profile)# codec audio PCMU
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку TRUNK_2:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_2
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

В результате конфигурация медиапрофилей будет выглядеть следующим образом:

```
  media profile FOR_TRUNK_1
    codec audio PCMA
  exit
  media profile FOR_TRUNK_2
    codec audio PCMU
  exit
```

В SDP Offer1, полученном с транка TRUNK_1, указан кодек PCMA, и т. к. в медиапрофиле FOR_TRUNK_2 указан только кодек PCMU для транскодирования, в SDP Offer2, который будет отправлен в TRUNK_2, кодек PCMA будет заменен на PCMU. Соответственно при вызовах из TRUNK_1 в TRUNK_2 (и в обратном направлении) возможно только транскодирование медиаданных.



Если в SDP Offer1, полученном с транка TRUNK_1, будут указаны любые кодеки кроме PCMA, то вызов не будут установлен, ESBC отправит на INVITE ответ 488.

2. Использование медиапрофилей для проксирования и транскодирования аудиоданных.

Для транков TRUNK_1 и TRUNK_2 используются медиапрофили, позволяющие проксировать все кодеки и транскодировать аудио G722 <---> G729 и GSM <---> G729.

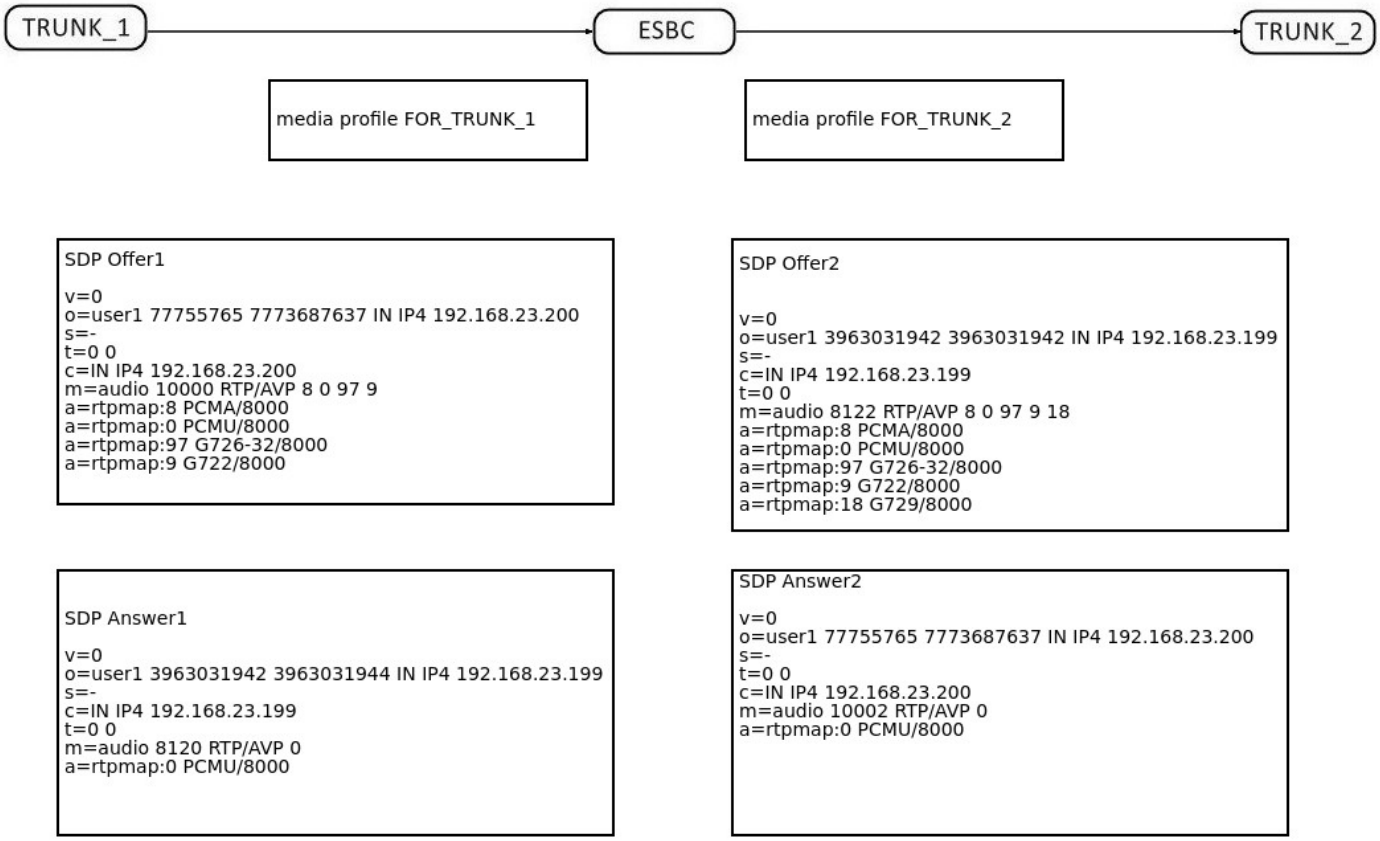
Настройка медиапрофилей:

```
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль для транка TRUNK_1:
vesbc(config-esbc)# media profile FOR_TRUNK_1

#Включить поддержку кодеков G722 и GSM в режиме транскодирования:
vesbc(config-esbc-media-profile)# codec audio G722
vesbc(config-esbc-media-profile)# codec audio GSM
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку TRUNK_1:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_1
vesbc(config-esbc-trunk-sip)# exit

#Создать медиапрофиль для транка TRUNK_2:
vesbc(config-esbc)# media profile FOR_TRUNK_2

#Включить поддержку кодека G729 в режиме транскодирования:
vesbc(config-esbc-media-profile)# codec audio G729
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку TRUNK_2:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)# media profile FOR_TRUNK_2
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

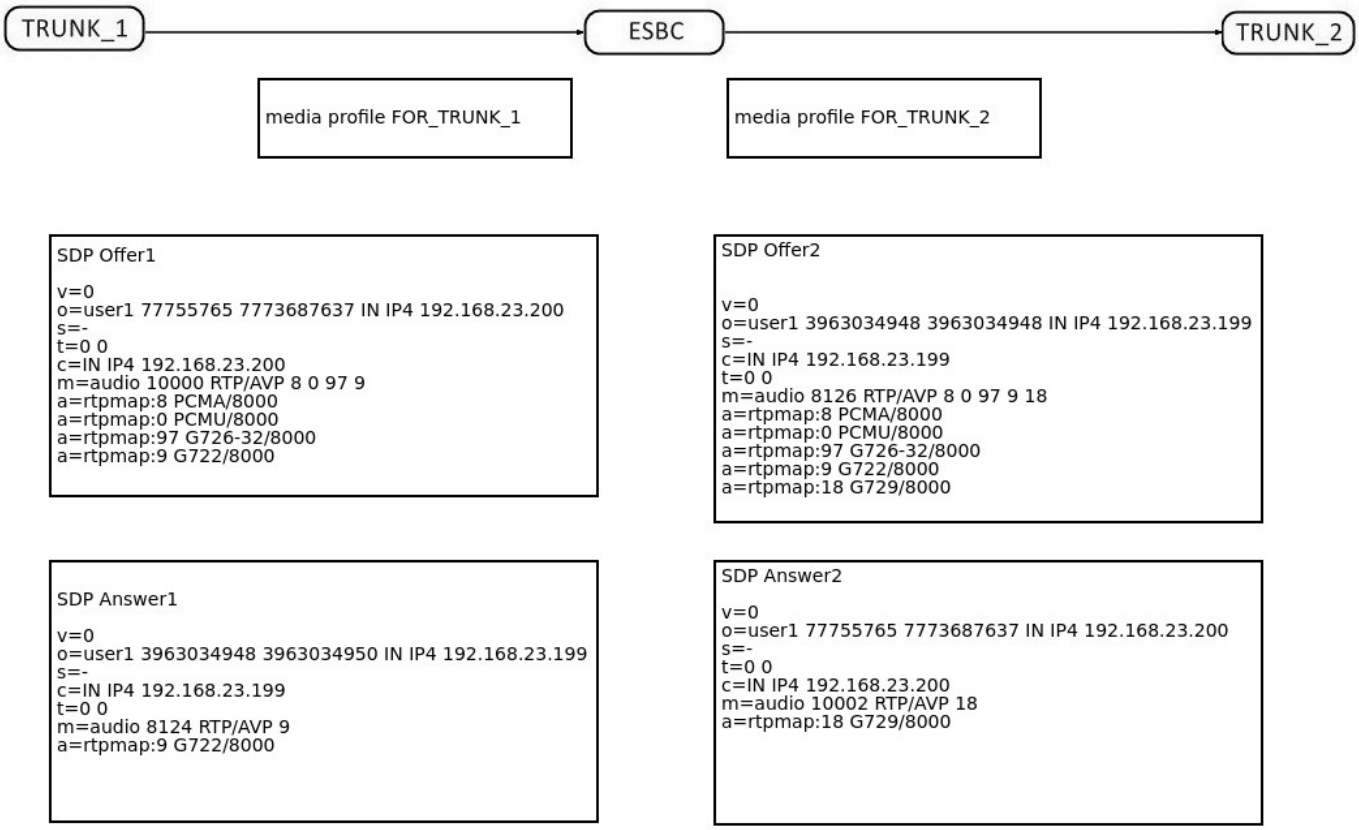
В результате конфигурация медиапрофилей будет выглядеть следующим образом:

```
media profile FOR_TRUNK_1
  codec audio GSM
  codec audio G722
  codec allow AMR
  codec allow CLEARMODE
  codec allow CN
  codec allow G72
  codec allow G722/ 9
  codec allow G728 15
  codec allow G729/ 18
  codec allow GSM 3
  codec allow H26
  codec allow H261 31
  codec allow H263 34
  codec allow ILBC
  codec allow L16/44100 11
  codec allow L16/44100/2 10
  codec allow OPUS
  codec allow PCMA 8
  codec allow PCMU 0
  codec allow SPEEX
  codec allow T38 t38
  codec allow VP
  codec allow telephone-event
exit

media profile FOR_TRUNK_2
  codec audio G729
  codec allow AMR
  codec allow CLEARMODE
  codec allow CN
  codec allow G72
  codec allow G722/ 9
  codec allow G728 15
  codec allow G729/ 18
  codec allow GSM 3
  codec allow H26
  codec allow H261 31
  codec allow H263 34
  codec allow ILBC
  codec allow L16/44100 11
  codec allow L16/44100/2 10
  codec allow OPUS
  codec allow PCMA 8
  codec allow PCMU 0
  codec allow SPEEX
  codec allow T38 t38
  codec allow VP
  codec allow telephone-event
exit
```

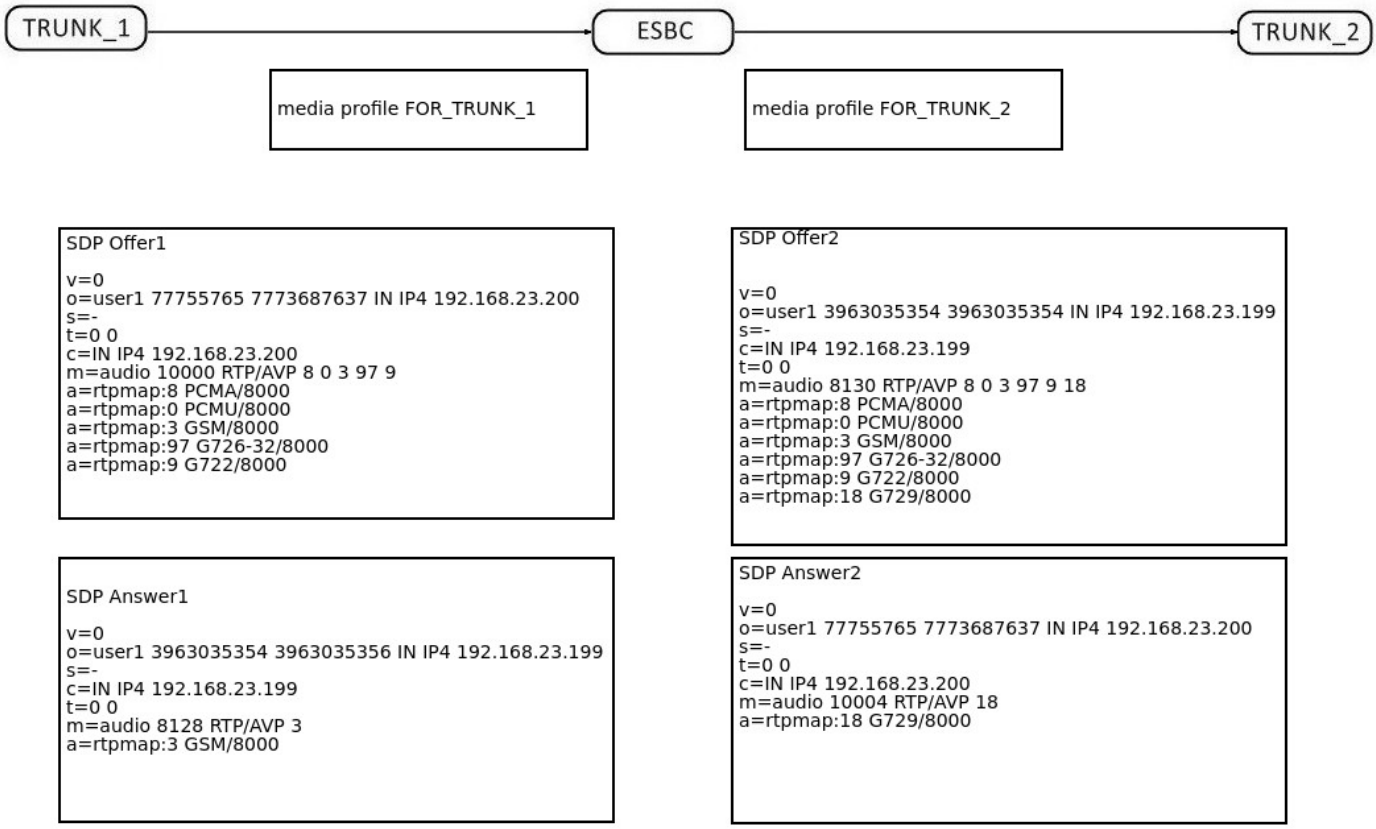
2.1 В SDP Offer1, полученном с транка TRUNK_1, указаны кодеки PCMA, PCMU, G726 и G722. Т. к. в медиапрофиле FOR_TRUNK_1 есть кодек G722, разрешенный для транскодирования, то в SDP Offer2, который будет отправлен в TRUNK_2, будет добавлен кодек G729. Остальные кодеки будут передаваться из SDP Offer1 в SDP Offer2, т. к. на обоих медиапрофилях настроены шаблоны, разрешающие проксирование этих кодеков. В SDP Answer2, полученном из TRUNK_2, указан кодек PCMU. Этот кодек будет согласован ESBC в SDP Answer1. Т. к. этот кодек был в SDP Offer1, то будет выбран режим проксирования медиаданных.



2.2 В SDP Offer1, полученном с транка TRUNK_1, указаны кодеки PCMA, PCMU, G726 и G722. Т. к. в медиапрофиле FOR_TRUNK_1 есть кодек G722, разрешенный для транскодирования, то в SDP Offer2, который будет отправлен в TRUNK_2, будет добавлен кодек G729. Остальные кодеки будут передаваться из SDP Offer1 в SDP Offer2, т. к. на обоих медиапрофилях настроены шаблоны, разрешающие проксирование этих кодеков. В SDP Answer2, полученном из TRUNK_2, указан кодек G729. Т. к. этого кодека не было в SDP Offer1, то будет согласован единственный возможный кодек для TRUNK_1 — G722. Т. к. кодеки на плечах TRUNK_1 и TRUNK_2 отличаются, будет включено транскодирование медиаданных G722 <---> G729.



2.3 В SDP Offer1, полученном с транка TRUNK_1, указаны кодеки PCMA, PCMU, GSM, G726 и G722. Т. к. в медиапрофиле FOR_TRUNK_1 есть кодеки G722 и GSM, разрешенные для транскодирования, то в SDP Offer2, который будет отправлен в TRUNK_2, будет добавлен кодек G729. Остальные кодеки будут передаваться из SDP Offer1 в SDP Offer2, т. к. на обоих медиапрофилях настроены шаблоны, разрешающие проксирование этих кодеков. В SDP Answer2, полученном из TRUNK_2, указан кодек G729. Т. к. этого кодека не было в SDP Offer1, то будет согласован наиболее приоритетный кодек для TRUNK_1 — GSM. Т. к. кодеки на плечах TRUNK_1 и TRUNK_2 отличаются, будет включено транскодирование медиаданных GSM <---> G729.



9.11.3 Таймаут ожидания RTP-пакетов

Функция контроля состояния разговорного тракта по наличию RTP-трафика от встречного устройства. Данный механизм, а также описанные ниже механизмы могут использоваться для предотвращения зависания разговорных сессий в случае возникновения неисправностей на сети передачи данных. Они могут использоваться в дополнение или в качестве альтернативы таймерам SIP-сессий (RFC 4028).

Контроль осуществляется следующим образом: если в течение заданного времени от встречного устройства не поступает ни одного RTP-пакета, то вызов завершается.
По умолчанию контроль выключен.

Пример настройки RTP-таймаута:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media-profile)#

#Включение таймера 60 секунд в медиапрофиле:
vesbc(config-esbc-media-profile)# rtp timeout 60
vesbc(config-esbc-media-profile)#

vesbc(config-esbc-media profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку NEW_TRUNK:
vesbc(config-esbc)# trunk sip NEW_TRUNK
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Если в вызове, который был установлен через транк NEW_TRUNK, в течение 1 минуты в ESBC не будут приходить RTP-пакеты с любой из сторон, то вызов будет принудительно завершён.

Таймаут ожидания RTP-пакетов после получения Comfort Noise

Эта функция является дополнением к функции "Таймаут ожидания RTP-пакетов" с учетом контроля состояния разговора по наличию RTP-трафика при использовании пакетов Comfort Noise в RTP-потоке.

Данный таймаут является множителем для ***rtp timeout***. Множитель — это коэффициент, который определяет, во сколько раз значение данного таймаута больше, чем таймаут ожидания RTP-пакетов. Принимает значения из диапазона 2–30.

Контроль осуществляется следующим образом: если в течение данного времени от встречного устройства не поступает ни одного RTP-пакета и последний пакет был пакетом CN (Comfort Noise), то вызов завершается.

По умолчанию множитель не установлен.

⚠ CN-множитель будет использоваться, только если настроен таймаут ожидания RTP-пакетов.

Пример настройки множителя:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media-profile)#

#Включение таймаута ожидания RTP-пакетов в 60 секунд:
vesbc(config-esbc-media-profile)# rtp timeout 60
vesbc(config-esbc-media-profile)#

#Установить множитель таймаута ожидания RTP-пакетов после получения Comfort Noise:
vesbc(config-esbc-media-profile)# rtp timeout cn 5
vesbc(config-esbc-media-profile)#

vesbc(config-esbc-media profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку NEW_TRUNK:
vesbc(config-esbc)# trunk sip NEW_TRUNK
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Если в вызове, который был установлен через транк NEW_TRUNK, в течение 1 минуты в ESBC не будут приходить RTP-пакеты с любой из сторон, а последний пакет был пакетом CN, то таймаут ожидания RTP-пакетов увеличится в 5 раз. Таким образом, если в течение 5 минут с момента получения CN-пакета не поступает ни одного RTP-пакета, то вызов будет принудительно завершён.

Таймаут ожидания RTCP-пакетов

Функция контроля состояния разговорного тракта по наличию RTCP-трафика от встречного устройства. Таймаут принимает значения из диапазона 10–300 с. Контроль осуществляется следующим образом: если в течение заданного времени от встречного устройства не поступает ни одного RTCP-пакета, то вызов завершается.

По умолчанию контроль выключен.

Пример настройки RTCP-таймаута:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media-profile)#

#Включение таймаута ожидания RTCP-пакетов в 30 секунд:
vesbc(config-esbc-media-profile)# rtcp timeout 30
vesbc(config-esbc-media-profile)#

vesbc(config-esbc-media profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку NEW_TRUNK:
vesbc(config-esbc)# trunk sip NEW_TRUNK
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Если в вызове, который был установлен через транк NEW_TRUNK, в течение 30 секунд в ESBC не будут приходить RTCP-пакеты с любой из сторон, то вызов будет принудительно завершён.

> ⚠ Рекомендуется настраивать таймауты ожидания RTP и RTCP-пакетов совместно.

Отсутствие RTP-пакетов не является явным признаком неисправности на сети передачи данных, т. к. разговорный канал может быть деактивирован временно какой-либо из сторон, например, через согласование SDP a=inactive. При этом пакеты RTCP будут передаваться сторонами в этом состоянии.

При использовании только RTCP-таймаута, ESBC не анализирует наличие RTP-пакетов, и в случае, когда стороны не используют RTCP, сессия завершится спустя время, равное таймауту ожидания RTCP-пакетов. Поэтому, если заранее неизвестно, поддерживает ли встречная сторона RTCP, необходимо настроить не только RTCP, но и RTP-таймаут.

Логика совместной работы таймеров

Если время потери RTP-пакетов больше, чем RTP-таймаут, то проверяется RTCP-таймаут. Если RTCP-таймаут не настроен или время ожидания RTCP-пакетов истекло, то сессия завершается, иначе сессия останется активной, пока не истечёт RTCP-таймаут. Если в это время придёт RTP-пакет, то таймаут ожидания RTP перезапустится. Таким же образом происходит проверка RTP-таймаута, если истекает RTCP-таймаут.

Пример совместной настройки RTP и RTCP-таймаута:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media-profile)#

#Включение таймаута ожидания RTP-пакетов в 60 секунд:
vesbc(config-esbc-media-profile)# rtp timeout 60
vesbc(config-esbc-media-profile)#

#Включение таймаута ожидания RTCP-пакетов в 120 секунд:
vesbc(config-esbc-media-profile)# rtcp timeout 120
vesbc(config-esbc-media-profile)#

vesbc(config-esbc-media profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку NEW_TRUNK:
vesbc(config-esbc)# trunk sip NEW_TRUNK
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Если в вызове, который был установлен через транк NEW_TRUNK, в течение 1 минуты не будут приходить RTP-пакеты, а таймаут ожидания RTCP-пакетов ещё не истёк, то сессия не завершится. Таким образом, соединение будет поддерживаться, пока встречная сторона отправляет RTCP-пакеты. Если за это время придёт RTP-пакет, таймаут ожидания RTP-пакетов перезапустится. Иначе, если RTCP-пакеты также перестанут приходить, и ни один RTP-пакет так и не поступит, сессия завершится через 120 секунд с момента последнего пришедшего RTCP-пакета.

9.11.4 Локальная обработка RTCP

Опция локальной обработки позволяет ESBC самостоятельно генерировать RTCP-пакеты в сессиях, где одно из плеч не поддерживает RTCP.

По умолчанию RTCP-пакеты не генерируются, а RTCP-пакеты, полученные на одном из плеч вызова, пересылаются на второе плечо. В режиме локальной обработки RTCP-пакеты генерируются и отправляются на оба плеча сессии, а полученные пакеты обрабатываются.

Локальная обработка RTCP включается, если хотя бы на одном плече привязан медиапрофиль с включенной опцией локальной обработки.

Интервал посылки RTCP-пакетов — это период времени в секундах, через который устройство отправляет контрольные пакеты по протоколу RTCP, принимает значения в диапазоне 1–255 секунд. По умолчанию — 5 сек.

Пример настройки локальной обработки RTCP:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media-profile)#

#Включение опции локальной обработки RTCP:
vesbc(config-esbc-media-profile)# rtcp local enable
vesbc(config-esbc-media-profile)#

#Настройка интервала посылки RTCP-пакетов в 10 секунд:
vesbc(config-esbc-media-profile)# rtcp local interval 10
vesbc(config-esbc-media-profile)#

vesbc(config-esbc-media profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку NEW_TRUNK:
vesbc(config-esbc)# trunk sip NEW_TRUNK
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Во время вызова, который был установлен через NEW_TRUNK, каждые 10 секунд RTCP-пакеты будут генерироваться и отправляться в сторону транка NEW_TRUNK, интервал отправки RTCP-пакетов на другое плечо будет зависеть от настроек медиапрофиля у этого направления, если медиапрофиль не установлен, то будет использован интервал по умолчанию – 5 секунд.

9.11.5 SRTP

SRTP (Secure Real-time Transport Protocol) — это расширенная версия протокола RTP с набором защитных механизмов. Протокол был опубликован организацией IETF в стандарте RFC 3711. SRTP обеспечивает конфиденциальность за счет шифрования RTP-нагрузки. Для шифрования медиапотока SRTP стандартизирует использование только единственного шифра — AES, который может использоваться в двух режимах:

- Сегментированный целочисленный счётчик — типичный режим, который осуществляет произвольный доступ к любым блокам, что является существенным для трафика RTP, передающегося в публичных сетях с непредсказуемым уровнем надежности и возможной потерей пакетов. Но стандарт для шифрования данных RTP — только обычное целочисленное значение счётчика. AES, работающий в этом режиме, является алгоритмом шифрования по умолчанию, с длиной шифровального ключа в 128 бит и ключом сессии длиной в 112 бит.
- f8-режим — вариант режима способа обратной связи, расширенного, чтобы быть доступным с изменённой функцией инициализации. Значения по умолчанию для шифровального ключа и ключа сессии — то же, что и в AES в режиме, описанном выше.

SRTP использует функцию формирования ключа для создания ключей на основе мастер-ключа. Протокол управления ключами создает все ключи в сессии с помощью мастер-ключа. За счет того, что у каждой сессии свой уникальный ключ, все сессии защищены. Поэтому, если одна сессия была скомпрометирована, то остальные по-прежнему под защитой.

В конфигурации доступны 2 метода обмена ключами:

- DTLS-SRTP (RFC 5763)
- SDES (RFC 4568)

и 3 режима использования SRTP:

- disable — SRTP запрещён;
- optional — SRTP не обязателен, но ключи будут подставлены в offer SDP второго плеча без изменения профиля транспорта в медиасекции SDP;
- mandatory — SRTP обязателен, профиль транспорта в медиасекции SDP будет изменён на соответствующий профиль SRTP.

Если выбран режим mandatory и включены оба метода, то на втором плече будет выбран DTLS-SRTP, как более приоритетный.

> ⚠ По умолчанию поддержка SRTP выключена.

По умолчанию при использовании DTLS-SRTP используются сертификаты, сгенерированные автоматически. Для использования сертификатов, загруженных пользователем, необходимо в медиапрофиле указать криптопрофиль с необходимыми сертификатами командой *crypto profile*. Подробное описание криптопрофилей приведено в разделе Настройка криптопрофилей.

Пример использования SRTP

Схема:

В конфигурации есть два транка, настроена маршрутизация. Вызов, который приходит из TRUNK_IN, уходит в TRUNK_OUT. На TRUNK_OUT включаем обязательное использование SRTP с методом обмена ключами — SDES.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля NEW_MEDIA_PROFILE:
vesbc(config-esbc)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-media profile)#

#Настройка SRTP (включили обязательный режим использования, метод обмена ключами – SDES):
vesbc(config-esbc-media-profile)# srtp keying
  dtls-srtp  Enable DTLS-SRTP keying method
  sdes       Enable SDES keying method

vesbc(config-esbc-media-profile)# srtp keying sdes
vesbc(config-esbc-media-profile)# srtp mode
  disable    SRTP is disabled
  mandatory  SRTP is mandatory
  optional   SRTP is optional

vesbc(config-esbc-media-profile)# srtp mode mandatory
vesbc(config-esbc-media-profile)#


vesbc(config-esbc-media-profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# media profile NEW_MEDIA_PROFILE
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

С TRUNK_IN приходит INVITE с SDP offer:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): 100 61 74 IN IP4 10.25.72.54
    Session Name (s): Talk
    Connection Information (c): IN IP4 10.25.72.54
    Time Description, active time (t): 0 0
    Session Attribute (a): rtcp-xr:rcvr-rtt=all:10000 stat-summary=loss,dup,jitt,TTL voip-
metrics
    Session Attribute (a): record:off
    Media Description, name and address (m): audio 7078 RTP/AVP 96 97 98 0 8 18 101 99 100
    Media Attribute (a): rtpmap:96 opus/48000/2
    Media Attribute (a): fmtp:96 useinbandfec=1
    Media Attribute (a): rtpmap:97 speex/16000
    Media Attribute (a): fmtp:97 vbr=on
    Media Attribute (a): rtpmap:98 speex/8000
    Media Attribute (a): fmtp:98 vbr=on
    Media Attribute (a): fmtp:18 annexb=yes
    Media Attribute (a): rtpmap:101 telephone-event/48000
    Media Attribute (a): rtpmap:99 telephone-event/16000
    Media Attribute (a): rtpmap:100 telephone-event/8000
    Media Attribute (a): rtcp-fb:* trr-int 5000
    Media Attribute (a): rtcp-fb:* ccm tmmbr
    [Generated Call-ID: l0XaoKkqav]
```

На второе плечо (TRUNK_OUT) пересылаем SDP offer с ключами:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): 100 3932018917 3932018917 IN IP4 192.168.23.199
    Session Name (s): Talk
    Connection Information (c): IN IP4 192.168.23.199
    Time Description, active time (t): 0 0
    Session Attribute (a): rtcp-xr:rcvr-rtt=all:10000 stat-summary=loss,dup,jitt,TTL voip-
metrics
    Session Attribute (a): record:off
    Media Description, name and address (m): audio 8064 RTP/SAVP 96 97 98 0 8 18 101 99 100
    Media Attribute (a): rtpmap:96 opus/48000/2
    Media Attribute (a): fmtp:96 useinbandfec=1
    Media Attribute (a): rtpmap:97 speex/16000
    Media Attribute (a): fmtp:97 vbr=on
    Media Attribute (a): rtpmap:98 speex/8000
    Media Attribute (a): fmtp:98 vbr=on
    Media Attribute (a): fmtp:18 annexb=yes
    Media Attribute (a): rtpmap:101 telephone-event/48000
    Media Attribute (a): rtpmap:99 telephone-event/16000
    Media Attribute (a): rtpmap:100 telephone-event/8000
    Media Attribute (a): rtcp-fb:* trr-int 5000
    Media Attribute (a): rtcp-fb:* ccm tmmbr
    Media Attribute (a): crypto:1 AES_256_CM_HMAC_SHA1_80
inline:FGdOolKfBlrQzUIedHcIqs9uauWEnUbqxXpop9PaI1dPIHVnO/vdb7JJHRLBLw==
    Media Attribute (a): crypto:2 AES_256_CM_HMAC_SHA1_32
inline:Galc9UfOqBFNmr3ICc3Fiuc3HgEXlj+p1dRw85LavzjWR1sGZUr1nsLQjfaTQA==
    Media Attribute (a): crypto:3 AES_CM_128_HMAC_SHA1_80 inline:jEjWFKpqdf6d94g/
ddSjj1i08dEWQA1tTI75Hqx3
    Media Attribute (a): crypto:4 AES_CM_128_HMAC_SHA1_32 inline:uFYI2UDA/
+woJJY4fWljfoxRROffXNtEO81bBnHJ
    [Generated Call-ID: 503d40e930910767a2dd95f88b483189]
```

9.11.6 Контроль источника RTP

Контроль источника RTP позволяет принимать медиатрафик только с IP-адреса и порта, указанного в SDP встречной стороны, повышая безопасность взаимодействия при использовании публичной сети передачи данных.

Включение/выключение режима осуществляется командами *rtp source-verification* и *no rtp source-verification* соответственно. При отключенной проверке IP-адрес и порт источника RTP не проверяется.

По умолчанию опция включена.

> ⚠ При использовании опции "Контроль источника RTP" совместно с включенной опцией "nat comedia" в транке или абонентском интерфейсе, RTP-трафик будет передаваться на IP-адрес и порт из входящего потока.

9.11.7 Поддержка RFC5168 (PFU)

Picture Fast Update (PFU) — это механизм быстрого восстановления видеоизображения за счет обновления поврежденных кадров, обеспечивающий минимальные задержки и стабильное качество видео даже при потере пакетов.

> ⚠ Опция работает только для вызовов с транскодированием видео, при проксировании полученные сообщения INFO пересылаются на другое плечо.

Включение/выключение режима осуществляется командами *rfc5168 enable* и *no rfc5168 enable* соответственно.

По умолчанию опция выключена.

При выключенной поддержке и получении INFO с XML, в котором есть PFU, ESBC отправляет ответ 200 OK и INFO с SDP, в котором содержится "Unsupported picture fast update".

Примеры

1. При проксировании видео и получении сообщения INFO с XML, в котором есть PFU, ESBC передает сообщение INFO на второе плечо

Схема:

В конфигурации есть два транка, настроена маршрутизация. На обоих транках используется один видеокодек (например, VP8). Вызов, который приходит из TRUNK_IN, маршрутизируется через TRUNK_OUT.

С TRUNK_IN приходит сообщение INFO с XML:

```
eXtensible Markup Language
    <?xml
        version="1.0"
        encoding="utf-8"
        ?>
    <media_control>
        <vc_primitive>
            <to_encoder>
                <picture_fast_update/>
                </to_encoder>
            </vc_primitive>
        </media_control>
```

Т. к. используется проксирование видео, ESBC прокидывает сообщение INFO на второе плечо (в TRUNK_OUT):

```
eXtensible Markup Language
    <?xml
        version="1.0"
        encoding="utf-8"
        ?>
    <media_control>
        <vc_primitive>
            <to_encoder>
                <picture_fast_update/>
                </to_encoder>
            </vc_primitive>
        </media_control>
```

Встречная сторона (TRUNK_OUT) отвечает, поддерживает ли она INFO с XML PFU или нет.

2. Поддержка RFC5168 отключена. При транскодировании видео и получении сообщения INFO с XML, в котором есть PFU, ESBC ответит 200 OK и отправит INFO с "Unsupported picture fast update" в XML

Схема:

В конфигурации есть два транка, настроена маршрутизация. На транках используются разные видеокодеки, т. е. ESBC работает в режиме транскодирования видео (в примере TRUNK_IN использует VP8, а TRUNK_OUT H264). Вызов, который приходит из TRUNK_IN, маршрутизируется через TRUNK_OUT. На обоих транках в настройках медиапрофиля поддержка использования RFC5168 отключена.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создать медиапрофиль MEDIA_PROFILE_TRUNK_IN (по умолчанию поддержка rfc5168 отключена):
vesbc(config-esbc)# media profile MEDIA_PROFILE_TRUNK_IN
vesbc(config-esbc-media-profile)# codec video VP8
vesbc(config-esbc-media-profile)# no codec allow H26
vesbc(config-esbc-media-profile)# no codec allow H261
vesbc(config-esbc-media-profile)# no codec allow H263
vesbc(config-esbc-media-profile)# exit
vesbc(config-esbc)#

#Создать медиапрофиль MEDIA_PROFILE_TRUNK_OUT (по умолчанию поддержка rfc5168 отключена):
vesbc(config-esbc)# media profile MEDIA_PROFILE_TRUNK_OUT
vesbc(config-esbc-media-profile)# codec video H264
vesbc(config-esbc-media-profile)# no codec allow VP
vesbc(config-esbc-media-profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль MEDIA_PROFILE_TRUNK_IN к транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_TRUNK_IN
vesbc(config-esbc-trunk-sip)#

#Привязать медиапрофиль MEDIA_PROFILE_TRUNK_OUT к транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_TRUNK_OUT
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

В TRUNK_IN приходит сообщение INFO с XML:

```
eXtensible Markup Language
    <?xml
        version="1.0"
        encoding="utf-8"
        ?>
    <media_control>
        <vc_primitive>
            <to_encoder>
                <picture_fast_update/>
                </to_encoder>
            </vc_primitive>
        </media_control>
```

Т. к. поддержка RFC5168 отключена, ESBC ответит 200 OK и отправит сообщение INFO с XML, в котором содержится "Unsupported picture fast update":

```
eXtensible Markup Language
    <?xml
        version="1.0"
        encoding="utf-8"
        ?>
    <media_control>
        <general_error>
            Unsupported picture fast update
            </general_error>
        </media_control>
```

3. Поддержка RFC5168 включена. При транскодировании видео и получении сообщения INFO с XML, в котором есть PFU, ESBC самостоятельно обрабатывает сообщение INFO

Схема:

В конфигурации есть два транка, настроена маршрутизация. На транках используются разные видеокодеки, т. е. ESBC работает в режиме транскодирования видео (в примере TRUNK_IN использует VP8, а TRUNK_OUT H264). Вызов, который приходит из TRUNK_IN, маршрутизируется через TRUNK_OUT. На обоих транках в настройках медиапрофиля указываем, что поддержка использования RFC5168 включена.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля MEDIA_PROFILE_TRUNK_IN с включенной поддержкой RFC5168:
vesbc(config-esbc)# media profile MEDIA_PROFILE_TRUNK_IN
vesbc(config-esbc-media-profile)# codec video VP8
vesbc(config-esbc-media-profile)# no codec allow H26
vesbc(config-esbc-media-profile)# no codec allow H261 31
vesbc(config-esbc-media-profile)# no codec allow H263 34
vesbc(config-esbc-media-profile)# rfc5168 enable
vesbc(config-esbc-media-profile)# exit
vesbc(config-esbc)#

#Создание медиапрофиля MEDIA_PROFILE_TRUNK_OUT с включенной поддержкой RFC5168:
vesbc(config-esbc)# media profile MEDIA_PROFILE_TRUNK_OUT
vesbc(config-esbc-media-profile)# codec video H264
vesbc(config-esbc-media-profile)# no codec allow VP
vesbc(config-esbc-media-profile)# rfc5168 enable
vesbc(config-esbc-media-profile)# exit
vesbc(config-esbc)#

#Привязать медиапрофиль к транку TRUNK_IN:
vesbc(config-esbc)# trunk sip TRUNK_IN
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_TRUNK_IN
vesbc(config-esbc-trunk-sip)#

#Привязать медиапрофиль к транку TRUNK_OUT:
vesbc(config-esbc)# trunk sip TRUNK_OUT
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_TRUNK_OUT
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled..
```

С TRUNK_IN приходит сообщение INFO с XML:

```
eXtensible Markup Language
    <?xml
        version="1.0"
        encoding="utf-8"
        ?>
    <media_control>
        <vc_primitive>
            <to_encoder>
                <picture_fast_update/>
                </to_encoder>
            </vc_primitive>
        </media_control>
```

Т. к. используется транскодирование, то ESBC самостоятельно обрабатывает INFO, отправляет 200 OK.

9.11.8 Настройка Media Offroad

Media Offroad (внетрактовый медиапоток) — это архитектурный подход в IP-телефонии (VoIP), при котором сигнальный трафик (SIP) проходит через пограничный контроллер сессий ESBC, а голосовой или видеопоток (медиатрафик RTP/SRTP) направляется напрямую между конечными точками (абонентами, IP АТС и т. д.) в обход ESBC.

При использовании Media Offroad, ESBC не изменяет IP-адреса и порты в SDP при обработки вызовов.

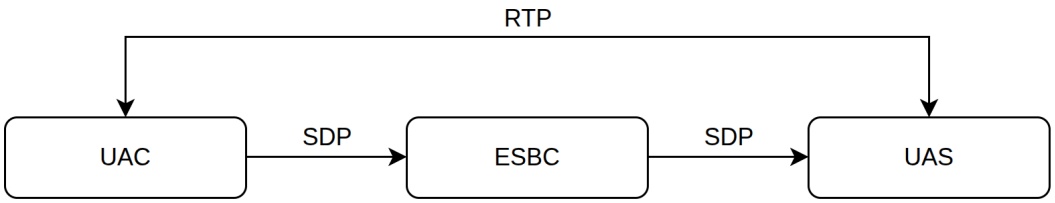
> ⚠ Если направления находятся в одной media offroad группе, то при вызове на эти направления будет работать только функция проксирование медиапрофиля. Остальные настройки медиапрофиля не будут влиять на SDP.
> Настройки public-ip и stun server на транках и на абонентских интерфейсах, использующих медиапрофиль с одинаковой media offroad группой, также не будут использоваться при работе режима Media Offroad.

> ⚠ Режим Media Offroad возможен только в случае, когда на входящем и исходящем направлении используются медиапрофили с **одинаковой** media offroad группой. Иначе вызов будет обрабатываться в режиме передачи медиапотоков через ESBC.

Пример использования Media Offroad

Нужно настроить передачу медии напрямую между клиентами.

Схема:



В конфигурации есть два транка, настроена маршрутизация. Для того чтобы медиапотоки ходили напрямую между UAC и UAS на транках, необходимо настроить медиапрофили с одной media offroad группой.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Создание медиапрофиля MEDIA_PROFILE_UAC с media offroad группой OFFROAD:
vesbc(config-esbc)# media profile MEDIA_PROFILE_UAC
vesbc(config-esbc-media-profile)# media offroad group OFFROAD
vesbc(config-esbc-media-profile)# exit

#Создание медиапрофиля MEDIA_PROFILE_UAS с media offroad группой OFFROAD:
vesbc(config-esbc)# media profile MEDIA_PROFILE_UAS
vesbc(config-esbc-media-profile)# media offroad group OFFROAD
vesbc(config-esbc-media-profile)# exit

#Привязать медиапрофиль к транку UAC:
vesbc(config-esbc)# trunk sip UAC
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_UAC
vesbc(config-esbc-trunk-sip)#

#Привязать медиапрофиль к транку UAS:
vesbc(config-esbc)# trunk sip UAS
vesbc(config-esbc-trunk-sip)# media profile MEDIA_PROFILE_UAS
vesbc(config-esbc-trunk-sip)#

#Применить и подтвердить изменения:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled..
```

При согласовании SDP на второе плечо будут передаваться адрес и порт, полученные во входящем плече. Поэтому медиапотоки будут передаваться напрямую между абонентами.

Пример отправки SDP

1. **С включенным Media Offroad.**

С UAC приходит INVITE с SDP offer:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): - 5000927421830469481 6519451137306200781 IN IP4 192.168.126
.235
    Session Name (s): Session SDP
    Connection Information (c): IN IP4 192.168.126.235
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 23236 RTP/AVP 8 18 0 96
    Media Attribute (a): rtpmap:8 PCMA/8000
    Media Attribute (a): rtpmap:18 G729/8000
    Media Attribute (a): fmtp:18 annexb=no
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:96 telephone-event/8000
    Media Attribute (a): fmtp:96 0-15
    Media Attribute (a): silenceSupp:off - - - -
    [Generated Call-ID: dc57f841-a743-1200-ca98-6813e20921bf]
```

На UAS передается SDP без подмены адреса и порта:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): - 5000927421830469481 6519451137306200781 IN IP4 192.168.126
.235
    Session Name (s): Session SDP
    Connection Information (c): IN IP4 192.168.126.235
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 23236 RTP/AVP 8 18 0 96
    Media Attribute (a): rtpmap:8 PCMA/8000
    Media Attribute (a): rtpmap:18 G729/8000
    Media Attribute (a): fmtp:18 annexb=no
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:96 telephone-event/8000
    Media Attribute (a): fmtp:96 0-15
    Media Attribute (a): silenceSupp:off - - - -
    [Generated Call-ID: a35f417e272093e280283749209df178]
```

2. С выключенным Media Offroad.

С UAC приходит INVITE с SDP offer:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): - 8474717719612293321 8743226203380917861 IN IP4 192.168.126
.235
    Session Name (s): Session SDP
    Connection Information (c): IN IP4 192.168.126.235
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 23240 RTP/AVP 8 18 0 96
    Media Attribute (a): rtpmap:8 PCMA/8000
    Media Attribute (a): rtpmap:18 G729/8000
    Media Attribute (a): fmtp:18 annexb=no
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:96 telephone-event/8000
    Media Attribute (a): fmtp:96 0-15
    Media Attribute (a): silenceSupp:off - - - -
    [Generated Call-ID: 5290521b-a758-1200-ca98-6813e20921bf]
```

На UAS передаём SDP с адресом и портом ESBC:

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): - 3996657384 3996657384 IN IP4 192.168.126.231
    Session Name (s): Session SDP
    Connection Information (c): IN IP4 192.168.126.231
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 8010 RTP/AVP 8 18 0 96
    Media Attribute (a): rtpmap:8 PCMA/8000
    Media Attribute (a): rtpmap:18 G729/8000
    Media Attribute (a): fmtp:18 annexb=no
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:96 telephone-event/8000
    Media Attribute (a): fmtp:96 0-15
    Media Attribute (a): silenceSupp:off - - - -
    [Generated Call-ID: 45d577c847420f66e60a155b69c815e4]
```

При согласовании SDP для использования SRTP на второе плечо будут передаваться ключи шифрования клиента, а не ESBC.

9.12 Управление безопасностью системы

В данном разделе описаны методы защиты и примеры настройки ESBC для защиты от различных SIP-атак.

Для более надежной защиты рекомендуется использовать дополнительные механизмы защиты от прочих сетевых атак, описанные в разделе Рекомендации по безопасной настройке.

9.12.1 Настройка профилей безопасности

Профили безопасности используются для управления механизмом защиты от SIP-атак. Использование профилей безопасности позволяет гибко управлять уровнями защиты для каждого направления.

Необходимый уровень защиты обеспечивается следующими настройками профиля:

- фильтрация SIP-флуда;
- блокировка по AOR/User-Agent;
- объединение ошибок по IP-адресу;
- защита от SIP-spoofing атак.

Профиль безопасности может использоваться в транках, транк-группах и абонентских интерфейсах.

Для обеспечения единой политики безопасности может быть использован один профиль для всех транков и один профиль для всех абонентских интерфейсов. Это может быть как один и тот же профиль безопасности, так и разные. Эти профили указываются в общих настройках ESBC.

Если для транка и транковой группы, в которую входит этот транк, используются разные профили безопасности, то будет применяться профиль, указанный в настройках транка. Если для транка/абонентского интерфейса и в общих настройках (для всех транков/абонентских интерфейсов) используются разные профили безопасности, то будет применяться профиль, указанный в настройках транка/абонентского интерфейса.

Описание всех команд для настройки профилей безопасности приведено в разделе Настройки профиля безопасности.

9.12.2 Общий принцип работы модуля fail2ban

Модуль занимается анализом возникающих ошибок для дальнейшей блокировки источников "подозрительного SIP-трафика". При возникновении ошибки в модуль отправляется информация о типе ошибки и об источнике. При накоплении достаточного количества ошибок источник блокируется.

Виды ошибок:

- ошибка регистрации;
- ошибка вызова;
- ошибка подписки;
- флуд SIP-пакетов;
- некорректный SIP-пакет;
- срабатывание флуд-фильтра;
- получение пакета вне транка/абонентского интерфейса.

Лимит количества ошибок зависит от нескольких факторов:

- вес ошибки;
- интервал времени между ошибками;
- количество ошибок одного вида;
- количество AOR, которые использовались в SIP-сообщениях с одного IP-адреса;
- количество IP-адресов, которые присылали SIP-сообщения с одним AOR.

При добавлении адреса в чёрный список указывается причина блокировки. Чёрный список можно просмотреть в CLI командой *show esbc black-list* или в WEB на странице Мониторинг → Списки доступа → Чёрный список.

Причины блокировки:

- ACCOUNT HACKING — превышен лимит по количеству ошибок с одинаковым AOR/User-Agent;
- PACKET FLOODING — превышен лимит по количеству ошибок с одного IP-адреса;
- BURST ERRORS — превышен глобальный лимит по количеству ошибок в секунду;
- GLOBAL RPS LIMIT — превышен глобальный лимит по количеству заблокированных запросов в секунду;
- IP RPS LIMIT — превышен лимит по количеству заблокированных запросов в секунду с одного IP-адреса;
- MONITORED ADDRESSES LIMIT — превышено максимальное количество IP-адресов с ошибками;
- DISTRIBUTED SPAM — превышено максимальное количество IP-адресов с одинаковым заблокированным атрибутом (AOR, User-Agent);
- BLOCKED ATTRIBUTES LIMIT — превышено максимальное количество заблокированных атрибутов (AOR, User-Agent);
- IP BLOCKED ATTRIBUTES LIMIT — превышено максимальное количество заблокированных атрибутов (AOR, User-Agent) с одного IP-адреса.

9.12.3 Фильтрация SIP-флуда

ESBC поддерживает создание флуд-фильтров для механизма конфигурируемой защиты от SIP-flood, а также для фильтрации клиентских приложений. Фильтр применяется ко всему SIP-сообщению (включая тело — SDP, XML и т. д.).

В настройках фильтра можно указать до 64 масок/шаблонов, по которым происходит поиск. В случае нахождения сообщение определяется как флуд и отбрасывается.

К профилю безопасности можно привязать до 8 флуд-фильтров.

При создании шаблона можно использовать регулярные выражения PCRE.

⚠ Регулярное выражение для флуд-фильтра применяется с логикой частичного совпадения (match).

Пример настройки флуд-фильтров

```
#Создание абонентских интерфейсов:
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip UI_1
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_UI_1
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_UI_1
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
vesbc(config-esbc-user-interface-sip)# exit
vesbc(config-esbc)# user-interface sip UI_2
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_UI_2
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_UI_2
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
vesbc(config-esbc-user-interface-sip)# exit
vesbc(config-esbc)# user-interface sip UI_3
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_UI_3
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_UI_3
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
vesbc(config-esbc-user-interface-sip)# exit

#Создание флуд-фильтра:
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# flood filter FLOOD_FILTER
vesbc(config-esbc-flood-filter)# pattern 0 7543546
vesbc(config-esbc-flood-filter)# pattern 1 flood
vesbc(config-esbc-flood-filter)# exit

#Привязка флуд-фильтра к профилю безопасности:
vesbc(config-esbc)# security profile SECURITY_PROFILE
vesbc(config-esbc-security-profile)# flood filter 0 FLOOD_FILTER
vesbc(config-esbc-security-profile)# exit

#Привязка профиля безопасности с флуд-фильтром ко всем абонентским интерфейсам:
vesbc(config-esbc)# general
vesbc(config-esbc-general)# security profile user-interface SECURITY_PROFILE

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

После применения изменений все входящие инициирующие запросы с абонентских интерфейсов UI_1, UI_2, UI_3 будут проверяться на наличие подстрок "7543546" и "flood". Если хотя бы одна подстрока будет найдена в сообщении, то оно отбросится, а в модуль fail2ban отправится уведомление о срабатывании флуд-фильтра. При накоплении достаточного количества ошибок источник блокируется.

ℹ Для того чтобы фильтр применялся ко всем сообщениям, а не только к инициирующим запросам, необходимо включить опцию apply-to-created.

Логика применения фильтров

По умолчанию все фильтры (шаблоны) применяются с логикой "ИЛИ", то есть сообщение отбросится, если будет найдено совпадение хотя бы по одному шаблону.

Есть возможность с помощью выражения задавать логику применения фильтров, доступные логические операторы:

- & — логическое И;
- | — логическое ИЛИ;
- ! — логическое отрицание.

Для обращения к фильтру необходимо использовать его индекс. В выражении можно использовать скобки для группировки фильтров.

Пример изменения логики применения фильтров

Конфигурация:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

vesbc(config-esbc)# flood filter FLOOD_FILTER

vesbc(config-esbc-flood-filter)# pattern 0 ".*scan.*"
vesbc(config-esbc-flood-filter)# pattern 1 ".*crack.*"
vesbc(config-esbc-flood-filter)# pattern 2 ".*trusted.*"
vesbc(config-esbc-flood-filter)# pattern expression "(0|1)&!2"
vesbc(config-esbc-flood-filter)# do commit
vesbc(config-esbc-flood-filter)# do confirm
```

Настройка **pattern expression "(0|1)&!2"** означает, что для срабатывания флуд-фильтра необходимо, чтобы SIP-сообщение содержало подстроку "scan" или "crack" и, при этом не содержало подстроку "trusted".

Фильтрация клиентских приложений

При помощи флуд-фильтров можно реализовать фильтрацию клиентских приложений, реализуется это добавлением шаблона следующего вида:

```
User-Agent:.*потенциально вредоносное клиентское приложение.*
```

В конфигурации флуд-фильтра есть команда, которая автоматически добавляет в конфигурацию фильтра 18 шаблонов с часто используемыми вредоносными клиентскими приложениями.

Соответствие запрещенных User-Agent и создаваемым дефолтным шаблоном представлено в таблице ниже.

Запрещенный User-Agent	Создаваемый дефолтный шаблон
scan	User-Agent:.*scan.*
crack	User-Agent:.*crack.*
flood	User-Agent:.*flood.*
kill	User-Agent:.*kill.*
sipcli	User-Agent:.*sipcli.*
sipv sipvicious	User-Agent:.*sipv.*
sipsak	User-Agent:.*sipsak.*
sundayaddr	User-Agent:.*sundayaddr.*
iWar	User-Agent:.*iWar.*
SIVuS	User-Agent:.*SIVuS.*
Gulp	User-Agent:.*Gulp.*
smap	User-Agent:.*smap.*
friendly-request	User-Agent:.*friendly-request.*
VaxIPUserAgent VaxSIPUserAgent	User-Agent:.*VaxS{0,1}IPUserAgent.*
siparmyknife	User-Agent:.*siparmyknife.*
Test Agent	User-Agent:.*Test Agent.*
SIPBomber	User-Agent:.*SIPBomber.*

Запрещенный User-Agent	Создаваемый дефолтный шаблон
Siprogue	User-Agent:.*Siprogue.*

Если в настройках фильтра недостаточно незаполненных шаблонов для создания всех default patterns, то они не создадутся.

9.12.4 Блокировка по AOR/User-Agent

По умолчанию при превышении лимита по ошибкам блокируется только адрес источника вредоносного трафика, в конфигурации профиля безопасности можно включить блокировку по AOR из заголовка From и блокировку по значению заголовка User-Agent.

При использовании одного и того же атрибута (AOR, User-Agent, IP-адрес) источником вредоносного трафика раньше всего сработает блокировка по AOR, потом по IP-адресу, потом по User-Agent. Из-за этого при ошибках с одинаковым AOR и IP-адресом может оказаться заблокированным только AOR, или при ошибках с одинаковым IP и User-Agent может оказаться заблокированным только IP-адрес.

Пример включения блокировки по AOR:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# security profile SECURITY_PROFILE

#Сбор ошибок по AOR:
vesbc(config-esbc-security-profile)# check aor
vesbc(config-esbc-security-profile)# exit

#Привязка профиля безопасности к абонентскому интерфейсу:
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# security profile SECURITY_PROFILE

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Пример блокировки:

С адреса 192.168.80.134 через абонентский интерфейс приходят сообщения INVITE с AOR 123@anonymous.invalid, вызовы без регистрации на интерфейсе запрещены, поэтому эти запросы отбиваются 403 Forbidden.

Если в профиле безопасности, привязанному к абонентскому интерфейсу, отключена блокировка по AOR, то через определенное количество запросов заблокируется только IP-адрес, и запросы с него больше обрабатываться не будут.

```
IP black-list:
--------------------------------------------------------------------------------
IP address      Ban reason   Address of Record (AOR)  Error   Blocking    Time of
blocking
                             User-Agent (UA)          count   timeout

                                                              in minutes

-------------  ------------ ------------------------  -----   --------
-----------------
192.168.80.134 IP RPS LIMIT                           16      1440
2026-09-10 04:54:43
```

Если к абонентскому интерфейсу привязан профиль безопасности с включенной блокировкой по AOR, то через какое-то время в чёрный список добавится AOR.
Все запросы с любого адреса, в котором будет заблокированный AOR во From, обрабатываться не будут.

```
AOR black-list:
-----------------------------------------------------------------------------------
AOR                     Ban reason         AOR        Forgive    Time of blocking
                                           error      time in

                                           count      minutes
---------------------   ----------------   ---------  --------  ------------------
123@anonymous.invalid   ACCOUNT HACKING    25         60       2026-09-10 05:08:57
```

Блокировка по AOR срабатывает раньше, чем блокировка по адресу, поэтому адрес не успеет попасть в черный список.

Пример включения блокировки по User-Agent:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# security profile SECURITY_PROFILE

#Сбор ошибок по User-Agent:
vesbc(config-esbc-security-profile)# check user-agent
vesbc(config-esbc-security-profile)# exit

#Привязка профиля безопасности к абонентскому интерфейсу:
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# security profile SECURITY_PROFILE

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Пример блокировки:

С нескольких адресов 192.168.80.13x через абонентский интерфейс приходят сообщения INVITE с User-Agent: sipflood, вызовы без регистрации на интерфейсе запрещены, поэтому эти запросы отбиваются 403 Forbidden.

Если в профиле безопасности, привязанному к абонентскому интерфейсу, отключена блокировка по User-Agent, то через определенное количество запросов заблокируются только IP-адреса, и запросы с них больше обрабатываться не будут:

```
IP black-list:
--------------------------------------------------------------------------------
IP address       Ban reason    Address of Record (AOR)  Error    Blocking     Time
of blocking
                               User-Agent (UA)          count    timeout

                                                                in minutes

---------------  ------------  -----------------------  ------   ----------
-------------------
192.168.80.134   IP RPS LIMIT                            16       1435
2026-09-10 04:49:48
192.168.80.135   IP RPS LIMIT                            16       1436
2026-09-10 04:50:11
192.168.80.136   IP RPS LIMIT                            16       1437
2026-09-10 04:51:49
192.168.80.137   IP RPS LIMIT                            16       1438
2026-09-10 04:52:25
192.168.80.138   IP RPS LIMIT                            16       1439
2026-09-10 04:53:23
192.168.80.139   IP RPS LIMIT                            16       1440
2026-09-10 04:54:43
```

Если к абонентскому интерфейсу привязан профиль безопасности с включенной блокировкой по User-Agent, то через какое-то время в чёрный список помимо IP-адресов добавится ещё и User-Agent.
Все запросы с любого адреса, в котором будет заблокированный User-Agent, обрабатываться не будут:

```
IP black-list:
--------------------------------------------------------------------------------
IP address        Ban reason     Address of Record (AOR)   Error    Blocking
Time of blocking
                                 User-Agent (UA)           count    timeout

                                                                    in minutes

---------------   -------------  -----------------------  -------   ----------
--------------------
192.168.80.134    IP RPS LIMIT       UA: sipflood          16        1435
2026-09-10 04:49:48
192.168.80.135    IP RPS LIMIT       UA: sipflood          16        1436
 2026-09-10 04:50:11
192.168.80.136    IP RPS LIMIT       UA: sipflood          16        1437
2026-09-10 04:51:49
192.168.80.137    IP RPS LIMIT       UA: sipflood          16        1438
 2026-09-10 04:52:25
192.168.80.138    IP RPS LIMIT       UA: sipflood          16        1439
 2026-09-10 04:53:23
192.168.80.139    IP RPS LIMIT       UA: sipflood          16        1440
 2026-09-10 04:54:43

User-agent black-list:
------------------------------------------------------------------------------------
---------
UA                          Ban reason         UA error    Forgive      Time of
blocking
                                               count       time in

                                                           minutes

-------------------------   ----------------   ---------   ----------
--------------------
sipflood                    ACCOUNT HACKING    61          60           2026-09-10
05:14:53
```

9.12.5 Объединение ошибок по IP-адресу

Данная опция позволяет объединять ошибки по IP-адресу.

Поведение по умолчанию – опция выключена, ошибки для каждого AOR/User-Agent считаются отдельно, блокируются при большом количестве ошибок с отдельного AOR/User-Agent.

При включенной опции, если ошибки имеют разный AOR/User-Agent, но одинаковый IP-адрес, то при блокировке адреса заблокируются все связанные AOR/User-Agent.

Настройка обеспечивает лучшую защиту от распределенных атак, но если много разных AOR/UA используют одинаковый IP-адрес, то могут быть ложные срабатывания.

Объединение ошибок работает, только если в профиле безопасности включена блокировка по AOR или User-Agent.

Пример включения объединения ошибок по IP-адресу:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# security profile SECURITY_PROFILE

#Сбор ошибок по AOR:
vesbc(config-esbc-security-profile)# check user-agent

#Объединение ошибок по адресу:
vesbc(config-esbc-security-profile)# errors aggregation
vesbc(config-esbc-security-profile)# exit

#Привязка профиля безопасности к абонентскому интерфейсу:
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# security profile SECURITY_PROFILE

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Пример блокировки:

С адреса 192.168.80.133 через абонентский интерфейс приходят сообщения INVITE с разными AOR, вызовы без регистрации на интерфейсе запрещены, поэтому эти запросы отбиваются 403 Forbidden.

Если в профиле безопасности, привязанному к абонентскому интерфейсу, отключено объединение ошибок по адресу, то через определенное количество запросов заблокируется только IP-адрес, и запросы с него больше обрабатываться не будут:

```
IP black-list:
 ---------------------------------------------------------------------------------
IP address        Ban reason         Address of Record (AOR)     Error
Blocking     Time of blocking
                                     User-Agent (UA)             count
timeout
                                                                                 in
minutes
---------------   ----------------   -------------------------   ---------
----------   --------------------
192.168.80.134    IP RPS LIMIT                                   16          1440
      2026-09-10 04:54:43
```

Если к абонентскому интерфейсу привязан профиль безопасности, в котором включено объединение ошибок по адресу, то через какое-то время в чёрный список помимо IP-адреса добавятся все связанные с этим адресом AOR.

Все запросы с любого адреса, в котором будут заблокированные AOR во From, обрабатываться не будут.

```
IP black-list:
-----------------------------------------------------------------------------------------
IP address     Ban reason     AOR                    AOR   Blocking  Time of blocking
                                                     error timeout
                                                     count in minutes
-------------- ------------- ----------------------- ----- ---------- -------------------
192.168.80.133 IP BLOCKED    24018@anonymous.invalid 1     1439       2025-07-31 07:11:32
               ATTRIBUTES LIMIT

AOR black-list:
-------------------------------------------------------------------------------
AOR                        Ban reason         AOR      Forgive   Time of blocking
                                              error    time in
                                              count    minutes
-------------------------  ----------------  ------   -------   --------------------
24001@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:29
24002@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:29
24003@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:29
24004@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:30
24005@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:30
24006@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:30
24007@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:30
24008@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:30
24009@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:31
24010@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:31
24011@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:31
24012@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:31
24013@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:31
24014@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:32
24015@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:32
24016@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:32
24017@anonymous.invalid    PACKET FLOODING    1        19        2025-07-31 07:11:32
24018@anonymous.invalid    PACKET FLOODING    1        58        2025-07-31 07:11:32
```

9.12.6 Защита от SIP-spoofing атак

Опция проверки IP-адреса источника SIP-сообщения позволяет защититься от SIP-spoofing атак.

IP-адрес и порт, с которого поступает SIP-сообщение от абонента, сравнивается с IP-адресом и портом, с которого ранее регистрировался этот абонент. Если абонент неизвестен или запрос пришёл с неизвестного направления, то запрос игнорируется, а в модуль fail2ban отправляется оповещение.

> Если в настройках абонентского интерфейса разрешены вызовы без регистрации, то создание сессии для неизвестного абонента разрешено. Если абонент зарегистрирован, то при получении SIP-сообщения с IP-адреса или порта, отличного от адреса и порта при регистрации, такое сообщение не будет обрабатываться.

Проверка работает на все запросы вне созданных сессий, если запрос пришёл на абонентский интерфейс и если запрос не является запросом регистрации.

В рамках созданной сессии адрес и порт, с которого пришёл запрос/ответ, сравнивается с адресом и портом источника инициирующего запроса. Если они не совпадают, то запрос/ответ игнорируется, а в модуль fail2ban отправляется оповещение. Сессия при этом не завершается.

При накоплении достаточного количества ошибок источник блокируется.

Поведение по умолчанию – опция выключена.

Пример включения проверки IP-адреса источника SIP-сообщения:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# security profile SECURITY_PROFILE

#Включение проверки адреса источника SIP-сообщения:
vesbc(config-esbc-security-profile)# check src-address
vesbc(config-esbc-security-profile)# exit

#Привязка профиля безопасности к абонентскому интерфейсу:
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# security profile SECURITY_PROFILE

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Примеры блокировки

1. С IP-адреса 192.168.23.242 и порта 5060 через абонентский интерфейс регистрируется абонент с AOR 123@domain.local, вызовы без регистрации на интерфейсе запрещены. Если на абонентский интерфейс придёт сообщение INVITE с AOR 123@domain.local, но с другого адреса или порта, например, с 192.168.23.242:5075, то данный запрос будет игнорирован, а в модуль fail2ban отправится оповещение.

2. С IP-адреса 192.168.23.242 и порта 5060 через абонентский интерфейс приходит сообщение INVITE с AOR 123@domain.local, вызов устанавливается. В профиле безопасности, привязанному к абонентскому интерфейсу, включена проверка адреса источника SIP-сообщения. Если в рамках сессии придёт запрос BYE с другого адреса или порта, например, с 192.168.80.35:5060, то данный запрос будет игнорирован, сессия не завершится, а IP-адрес 192.168.80.35 через определенное количество запросов будет заблокирован.

9.12.7 Настройка временных периодов

Временные периоды используются в случае блокировки объектов с целью гибкой настройки защиты от SIP-атак. Для этого в общих настройках реализованы конфигурации:

- времени блокировки (security block-timeout), которое используется при блокировке IP-адреса;
- времени прощения (security forgive-time), которое используется при блокировке атрибутов (AOR, user-agent и sip-user);
- времени хранения ошибок (security errors-window) – время, в течение которого накапливаются ошибки по каждому объекту блокировки.

> По умолчанию время блокировки составляет 1440 минут, время прощения — 60 минут, время хранения ошибок — 3600 секунд.

Описание всех команд выше приведено в разделе Общие настройки ESBC.

При блокировке атрибута и поступлении нового запроса, содержащего заблокированный атрибут, время прощения (Forgive time in minutes) восстановится в начальное значение.

При блокировке нескольких атрибутов и поступлении нового запроса, содержащего несколько заблокированных атрибутов, время прощения (Forgive time in minutes) восстановится в начальное значение только для самого приоритетного атрибута.

Пример:

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Изменение времени блокировки адресов на 2880 минут (2 дня):
vesbc(config-esbc-general)# security block-timeout 2880

#Изменение времени блокировки атрибутов на 600 минут (10 часов):
vesbc(config-esbc-general)# security forgive-time 600

#Изменение времени хранения ошибок на 7200 секунд (2 часа):
vesbc(config-esbc-general)# security errors-window 7200

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Пример блокировки адреса:

При получении невалидных сообщений с адреса 192.168.80.26 эти запросы отбиваются на втором плече 403 Forbidden.

При достижении определенного количества неудачных запросов этот адрес добавится в черный список с временем блокировки, которое было настроено в общих настройках.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Изменение времени блокировки адресов на 2880 минут (2 дня):
vesbc(config-esbc-general)# security block-timeout 2880

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.

Происходит блокировка адреса 192.168.80.134

vesbc# show esbc black-list ip
IP black-list:
--------------------------------------------------------------------------------
IP address      Ban reason    Address of Record (AOR)  Error   Blocking    Time of blocking
                              User-Agent (UA)          count   timeout
                                                               in minutes
--------------- ------------- -----------------------  ------  ----------  ------------------
192.168.80.134  IP RPS LIMIT                           16      2880        2026-09-10 04:54:43
```

Пример блокировки атрибута:

При получении невалидных сообщений с AOR 123@test.block эти запросы отбиваются на втором плече 403 Forbidden.

Если в привязанном профиле безопасности включен сбор ошибок по AOR, то при достижении определенного количества неудачных запросов AOR добавится в черный список с временем блокировки, которое было настроено в общих настройках.

```
vesbc#
vesbc# configure
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Изменение времени блокировки атрибутов на 600 минут (10 часов):
vesbc(config-esbc-general)# security forgive-time 600

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.

Происходит блокировка AOR 22222@192.168.80.26

vesbc# show esbc black-list aor
AOR black-list:
--------------------------------------------------------------------------------------------
AOR                         Ban reason         AOR         Forgive      Time of blocking
                                               error       time in
                                               count       minutes
-------------------------   ----------------   ---------   ----------   --------------------
22222@192.168.80.26-----------------------------------------------------------------------

AOR                    Ban reason       User        Forgive      Time of
                                        error       time in      blocking
                                        count       minutes
--------------------   --------------   ---------   ----------   ----------
101@192.168.126.217    TOOMANY USER     5           60           2026-09-10
                       REQUESTS                                  05:22:47        ACCOUNT
HACKING    25          600          2026-09-10 05:08:57
```

9.13 Настройка AAA

AAA (Authentication, Authorization, Accounting) — это базовая архитектура управления доступом, используемая в сетевых устройствах для идентификации клиентов, определения их прав и регистрации выполняемых действий.

В контексте ESBC механизмы AAA применяются для контроля доступа к устройству, управления сессиями и взаимодействия с внешними серверами аутентификации. Реализация AAA включает три ключевых компонента:

- Authentication (аутентификация) — процесс проверки подлинности клиента на основе предоставленных учетных данных.
- Authorization (авторизация, проверка полномочий, проверка уровня доступа) — определение разрешенных действий для прошедшего аутентификацию клиента.
- Accounting (учёт) — регистрация событий доступа, изменений и сессий для последующего аудита и анализа.

В данном разделе приведены параметры настройки, примеры конфигурации и рекомендации по применению различных механизмов AAA, используемых в ESBC.

9.13.1 Настройка аутентификации абонентов через RADIUS

ESBC поддерживает аутентификацию регистрирующихся через него абонентов на RADIUS-сервере.

Режимы работы digest-аутентификации:

- ***Draft Sterman*** — в данном режиме ESBC самостоятельно отправляет абоненту параметры для digest-аутентификации, далее эти параметры и digest response, полученный от абонента, передает на RADIUS-сервер для верификации. В сообщениях Access-Request используются атрибуты 206, 207 в соответствии с draft-sterman-aaa-sip-00.txt.
- ***RFC5090-no-challenge*** — в данном режиме ESBC самостоятельно отправляет абоненту параметры для digest-аутентификации, далее эти параметры и digest response, полученный от абонента, передает на RADIUS-сервер для верификации. В сообщениях Access-Request используются атрибуты 103–122, в соответствии с RFC 5090.
- ***RFC5090*** — в данном режиме параметры для digest-аутентификации (в сообщении ACCESS-CHALLENGE) ESBC получает от RADIUS-сервера и пересылает их абоненту.

> ⚠ В текущей версии ПО аутентификация через RADIUS-сервер поддерживается только для метода REGISTER.

Порядок настройки аутентификации через RADIUS-сервер:

1. Настроить RADIUS-сервер(ы).
2. Настроить radius profile.
3. Настроить aaa profile.
4. Использовать aaa profile в настройках абонентского интерфейса.

Настройка RADIUS-сервера

В таблице ниже приведены минимальные необходимые настройки RADIUS-сервера. Описание всех доступных настроек приведено в разделе Настройка AAA справочника команд CLI.

Шаг	Описание	Команда	Ключи
1	Добавить RADIUS-сервер в список используемых серверов и перейти в режим его конфигурирования.	**esbc(config)# radius-server host { <IP-ADDR> | <IPV6-ADDR> } [vrf <VRF>]**	<IP-ADDR> – IP-адрес RADIUS-сервера, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес RADIUS-сервера, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF] <VRF> – имя экземпляра VRF, задается строкой до 31 символа.
2	Задать пароль для аутентификации на удаленном RADIUS-сервере.	**esbc(config-radius-server)# key ascii-text { <TEXT> | encrypted <ENCRYPTED-TEXT> }**	<TEXT> – строка [8..16] ASCII-символов; <ENCRYPTED-TEXT> – зашифрованный пароль, размером [8..16] байт, задаётся строкой [16..32] символов.
3	Задать IPv4/IPv6-адрес, который будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых RADIUS-пакетах (можно не указывать при условии настройки п. 4).	**esbc(config-radius-server)# source-address { <ADDR> | object-group <NETWORK_OBJ_GROUP_NAME> }**	<ADDR> – IP-адрес источника, задается в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <NETWORK_OBJ_GROUP_NAME> – список адресов, которые будут использоваться в качестве source address.
4	Задать интерфейс или туннель маршрутизатора, IPv4/IPv6-адрес которого будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых RADIUS-пакетах. (можно не указывать при условии настройки п. 3).	**esbc(config-radius-server)# source-interface { <IF> | <TUN> }**	<IF> – имя интерфейса устройства, задаётся в виде, описанном в разделе Типы и порядок именования интерфейсов маршрутизатора; <TUN> – имя туннеля устройства, задаётся в виде, описанном в разделе Типы и порядок именования туннелей маршрутизатора.

Шаг	Описание	Команда	Ключи
5	Указать тип соединений, для аутентификации которых будет использоваться RADIUS-сервер.	**esbc(config-radius-server)# usage { all | aaa | auth | acct | pptp | l2tp | voip }**	Для аутентификации регистраций абонентов ESBC необходимо использовать тип соединения **usage voip**.

Возможно настроить до восьми RADIUS-серверов.

Настройка RADIUS-профиля

В таблице ниже приведены минимальные необходимые настройки RADIUS-профиля. Описание всех доступных настроек приведено в разделе Настройки RADIUS-профиля справочника команд CLI.

Шаг	Описание	Команда	Ключи
1	Перейти в раздел конфигурирования ESBC.	**esbc(config)# esbc**	
2	Добавить в конфигурацию RADIUS-профиль и перейти в режим его конфигурирования.	**esbc(config-esbc)# radius profile <NAME>**	<NAME> – название RADIUS-профиля, строка [1..64] символов.
3	Выбрать RADIUS-сервер, настроенный на предыдущем этапе для аутентификации. Допускается использование до восьми RADIUS-серверов в одном RADIUS-профиле.	**esbc(config-radius-profile)# radius-server host { <IP-ADDR> | <IPV6-ADDR> } [vrf <VRF>]**	<IP-ADDR> – IP-адрес RADIUS-сервера, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес RADIUS-сервера, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF] <VRF> – имя экземпляра VRF, задается строкой до 31 символа.

Настройка AAA-профиля

Шаг	Описание	Команда	Ключи
1	Перейти в раздел конфигурирования ESBC.	**esbc(config)# esbc**	
2	Добавить в конфигурацию AAA-профиль и перейти в режим его конфигурирования.	**esbc(config-esbc)# aaa profile <NAME>**	<NAME> – название AAA-профиля, строка [1..64] символов.
3	Выбрать RADIUS-профиль, настроенный на предыдущем этапе для аутентификации.	**esbc(config-aaa-profile)# auth radius profile <NAME>**	<NAME> – название RADIUS-профиля, строка [1..64] символов.

Для использования аутентификации абонентов через RADIUS-сервер необходимо указать AAA-профиль в конфигурации абонентского интерфейса (user-Interface), см. раздел Настройка абонентских интерфейсов.

Пример настройки:

Задача:

Использовать RADIUS-серверы 192.168.113.200 (основной) и 192.168.113.201 (резервный) для аутентификации регистраций абонентов, поступающих на абонентский интерфейс USERS_VIA_RADIUS в режиме RFC5090-no-challenge.

Решение:

1. Настроить основной RADIUS-сервер 192.168.113.200:

```
vesbc# configure
vesbc(config)# radius-server host 192.168.113.200
vesbc(config-radius-server)# key ascii-text password
vesbc(config-radius-server)# source-address 192.168.113.207
vesbc(config-radius-server)# usage voip
vesbc(config-radius-server)# exit
```

2. Настроить резервный RADIUS-сервер 192.168.113.201:

```
vesbc(config)# radius-server host 192.168.113.201
vesbc(config-radius-server)# key ascii-text password
vesbc(config-radius-server)# source-address 192.168.113.207
vesbc(config-radius-server)# usage voip
#Указываем более низкий приоритет, чем у основного сервера. По умолчанию приоритет 1. Чем
меньше значение, тем приоритетнее сервер.
vesbc(config-radius-server)# priority 100
vesbc(config-radius-server)# exit
```

3. Настроить RADIUS-профиль RP_USERS:

```
vesbc(config)# esbc
vesbc(config-esbc)# radius profile RP_USERS
#Указываем два RADIUS-сервера
vesbc(config-radius-profile)# radius-server host 192.168.113.200
vesbc(config-radius-profile)# radius-server host 192.168.113.201
#Указываем режим работы digest-аутентификации
vesbc(config-radius-profile)# auth digest rfc5090_no_chlng
vesbc(config-radius-profile)# exit
```

4. Настроить AAA-профиль AAA_USERS:

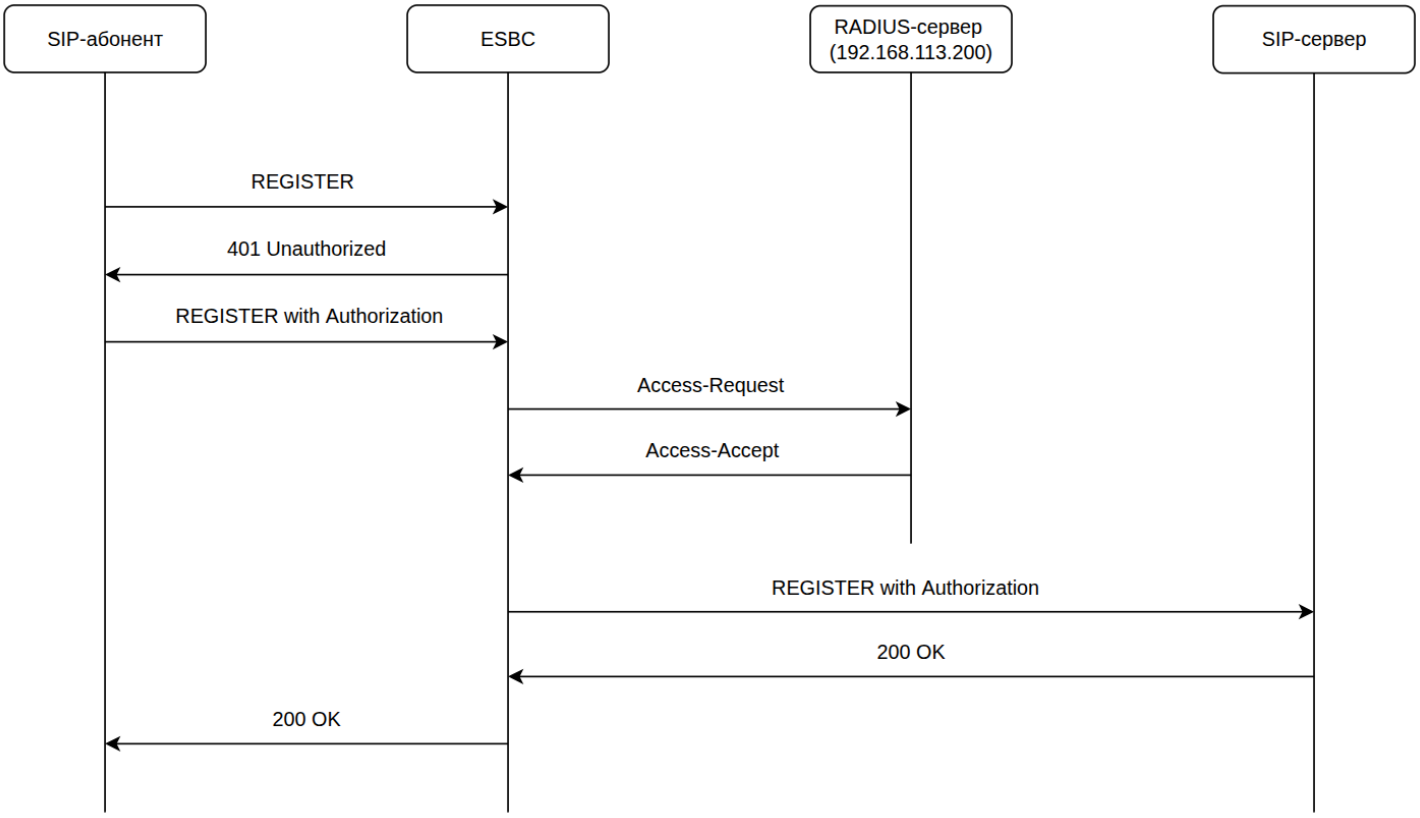
```
vesbc(config-esbc)# aaa profile AAA_USERS
#Указываем RADIUS-профиль
vesbc(config-aaa-profile)# auth radius profile RP_USERS
vesbc(config-aaa-profile)# exit
```

5. Использовать AAA-профиль AAA_USERS в настройках абонентского интерфейса USERS_VIA_RADIUS:

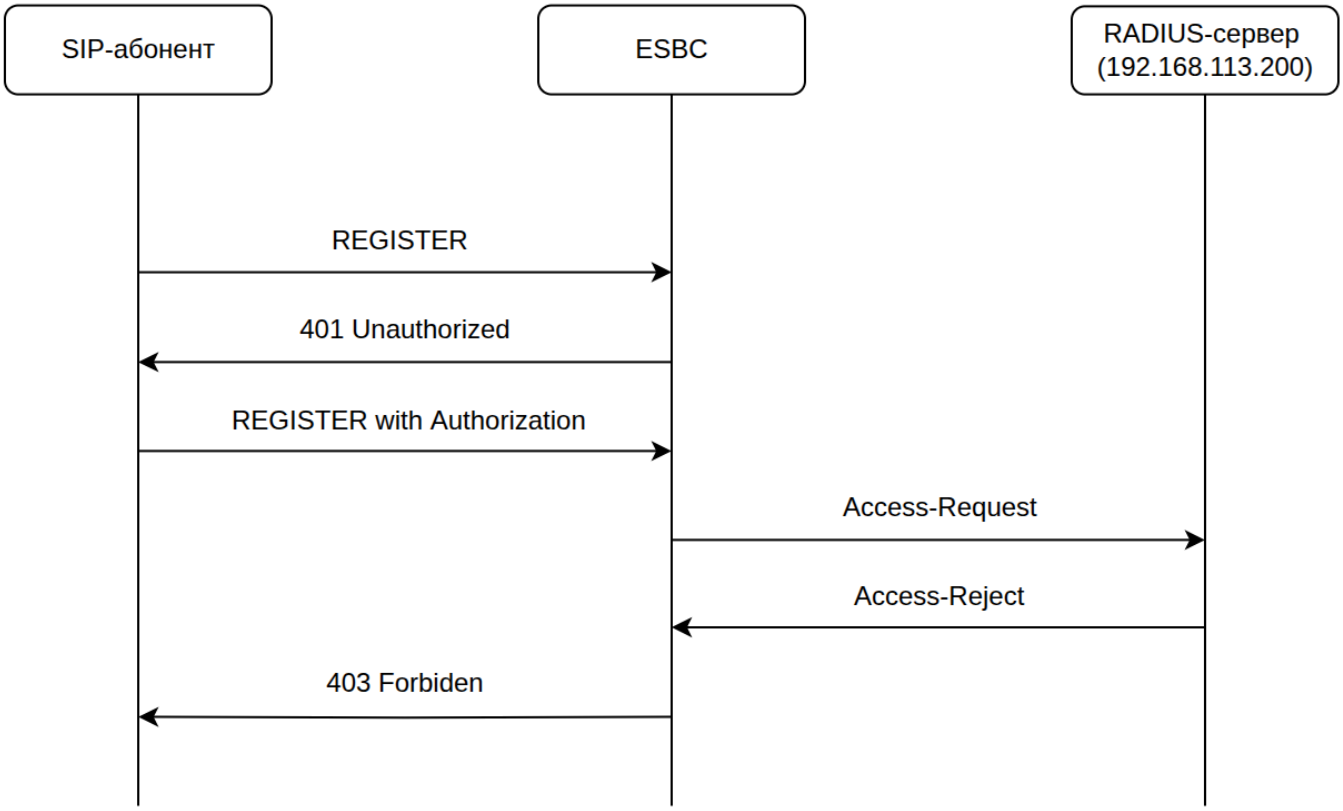
```
vesbc(config-esbc)# user-interface sip USERS_VIA_RADIUS
vesbc(config-esbc-user-interface-sip)# aaa profile AAA_USERS
vesbc(config-esbc-user-interface-sip)# exit
vesbc(config-esbc)# exit
vesbc(config)# exit
#Применяем и сохраняем все настройки
vesbc# commit
vesbc# confirm
```

При получении сообщений REGISTER на абонентский интерфейс USERS_VIA_RADIUS, ESBC будет отправлять абоненту параметры для digest-аутентификации в сообщении 401 Unauthorized, далее эти параметры и digest response, полученный от абонента, будет передаваться на RADIUS-сервер 192.168.113.200 в сообщении Access-Request.

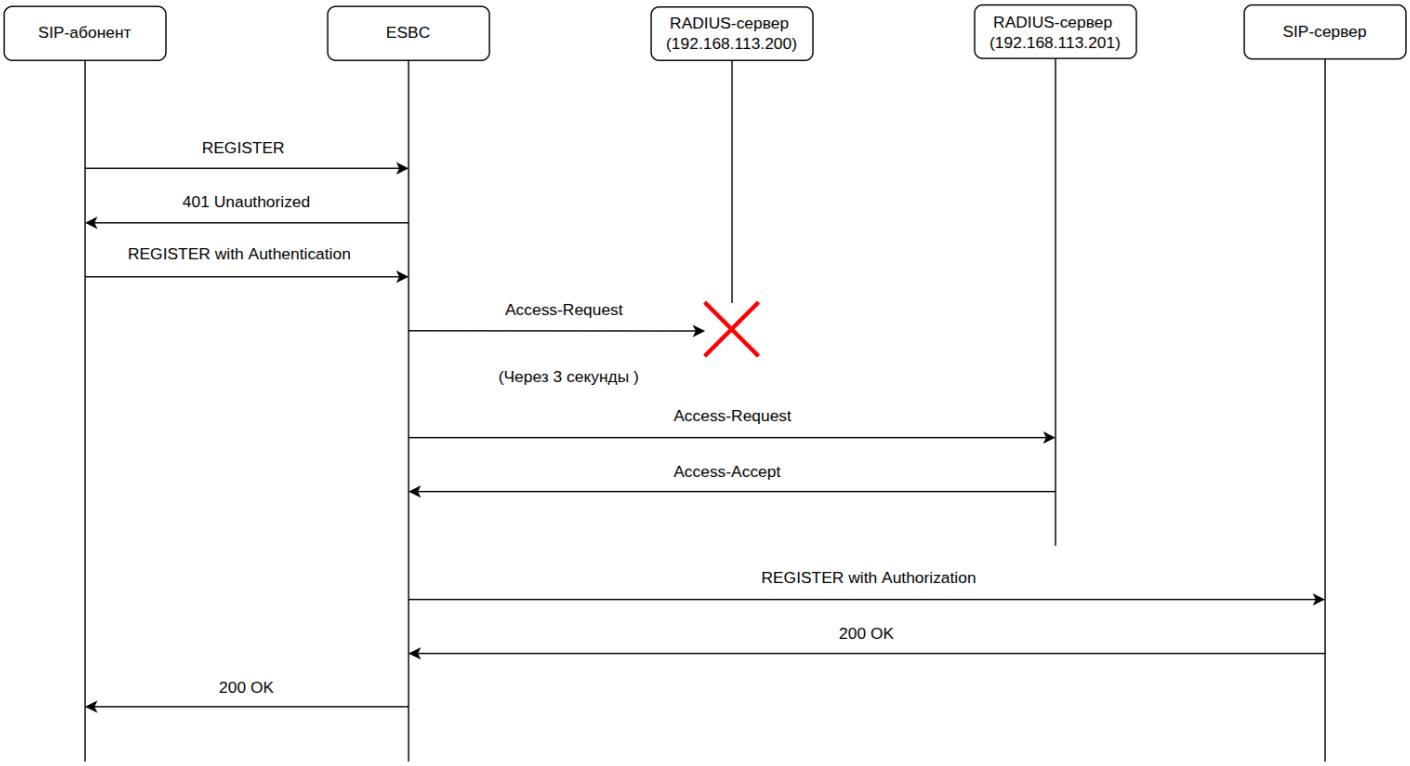
При получении от сервера Access-Accept, сообщение REGISTER от абонента будет направлено на вышестоящий SIP-сервер:



При получении Access-Reject, будет отправлено сообщение 403 Forbidden:



При недоступности RADIUS-сервера 192.168.113.200, после таймаута будет отправлено сообщение Access-Request на резервный RADIUS-сервер 192.168.113.201:



9.13.2 Настройка локальной аутентификации запросов

Локальная аутентификация на ESBC используется для аутентификации клиентов, используя учётные данные, привязанные к транку ESBC.

Алгоритм настройки транка для работы функции:

1. Добавить аутентификационные данные в профиль учётных данных.
2. Привязать профиль учётных данных к AAA-профилю.
3. Добавить правило выбора SIP-метода запроса, при совпадении с которым будет использоваться локальная аутентификация. Поддержана обработка 401 и 407 ответов на запросы: INVITE, REGISTER, UPDATE, INFO, REFER. При создании шаблона можно использовать регулярные выражения PCRE.

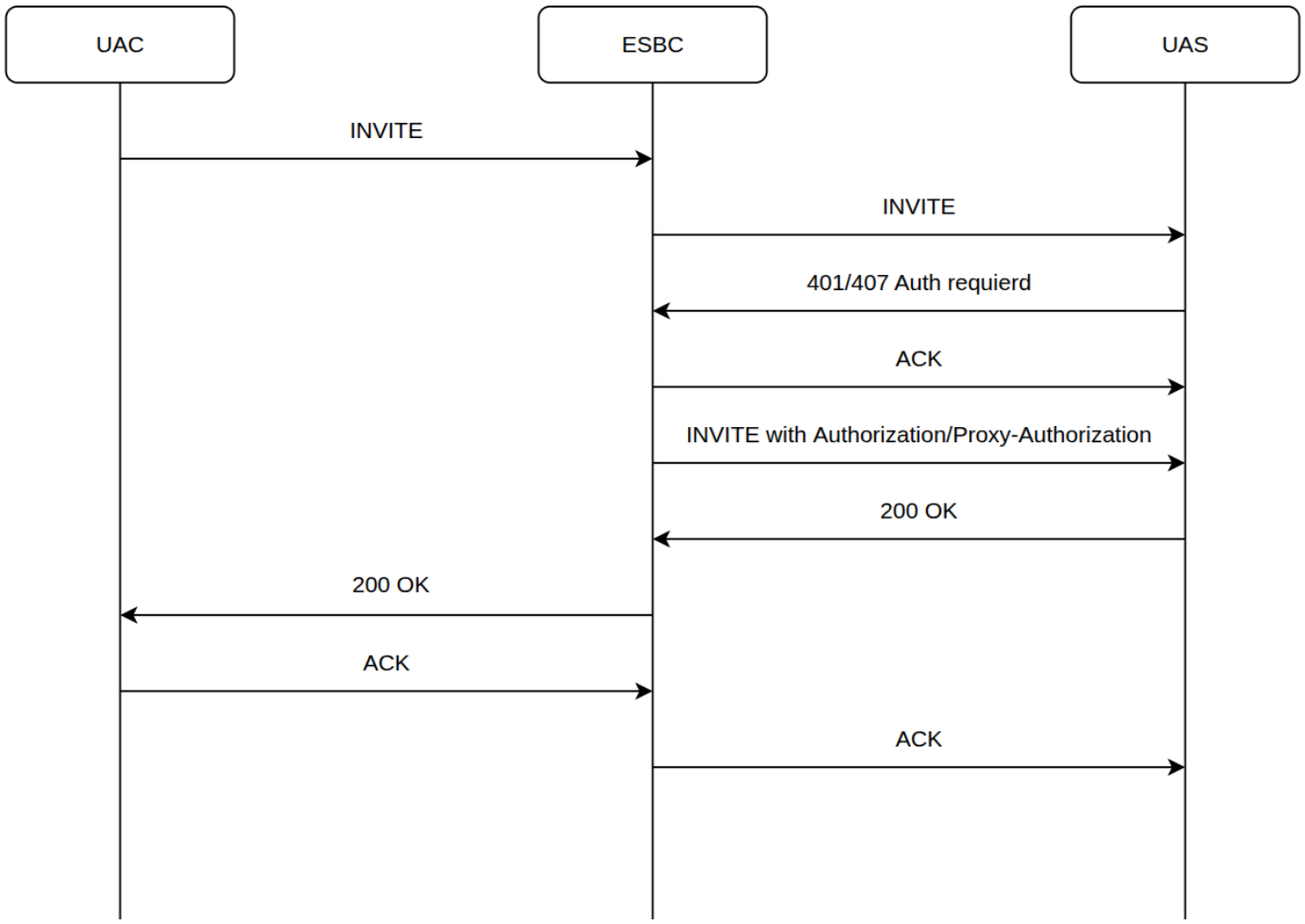
   > ⚠ По умолчанию установлено правило выбора метода REGISTER.

4. Привязать AAA-профиль к транку.

Если на отправленный запрос придёт ответ с требованием аутентификации, то ESBC обработает его и самостоятельно переотправит изначальный запрос на сервер вместе с аутентификационными данными.

> ⚠ Если в профиле учётных данных не будет подходящего номера, то ESBC перешлёт 401/407 ответ на первое плечо.

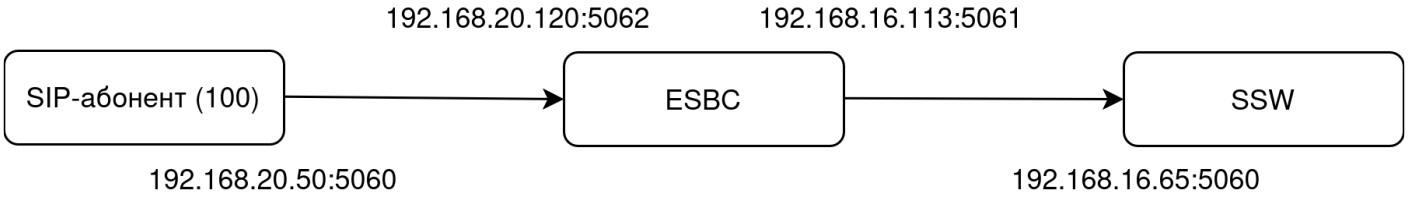
Схема работы:



Пример настройки локальной аутентификации абонента:

Задача:

Настроить локальную аутентификацию абонента с номером 100, логином 100 и паролем PASSWORD. При этом Softswitch запрашивает аутентификацию на запросы REGISTER и INVITE.



Решение:

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.16.113/24
vesbc(config-if-gi)# ip firewall disable
```

2. Настроить IP-адрес на внешнем интерфейсе в сторону абонентов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "ABONENTS"
vesbc(config-if-gi)# ip address 192.168.20.120/24
```

3. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5061
```

4. Создать SIP-транспорт в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# port 5062
```

5. Создать медиаресурсы для согласования и передачи голоса на плече ESBC — SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC — Абонентский шлюз/ SIP-абоненты:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_ABONENTS
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать профиль учётных данных с аутентификационными данными клиента:

```
vesbc#
vesbc# config
vesbc(config)# esbc
vesbc(config-esbc)# credential profile CREDENTIAL_PROFILE
vesbc(esbc-credential-profile)# number 100
vesbc(esbc-credential-profile-cred)# login 100
vesbc(esbc-credential-profile-cred)# password PASSWORD
```

8. Создать AAA профиль с созданным профилем учётных данных и шаблоном выбора обрабатываемых методов:

```
vesbc#
vesbc# config
vesbc(config)# esbc
vesbc(config-esbc)# aaa profile AAA_PROFILE
vesbc(config-aaa-profile)# credential local profile CREDENTIAL_PROFILE
vesbc(config-aaa-profile)# credential method-pattern INVITE|REGISTER
```

⚠ Регулярное выражение для credential method-pattern применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать методу "INVITE". В данном случае шаблон "INV.*" будет корректным, а шаблон "INV" — нет.

9. Создать SIP-транк в сторону SSW и добавить в него настроенный AAA профиль:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# aaa profile AAA_PROFILE
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.65
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

10. Создать таблицу маршрутизации и добавить туда правила, по которым вызовы, приходящие с абонентов, будут маршрутизироваться на SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW
```

11. Создать абонентский интерфейс в сторону абонентов и привязать к нему созданную таблицу маршрутизации:

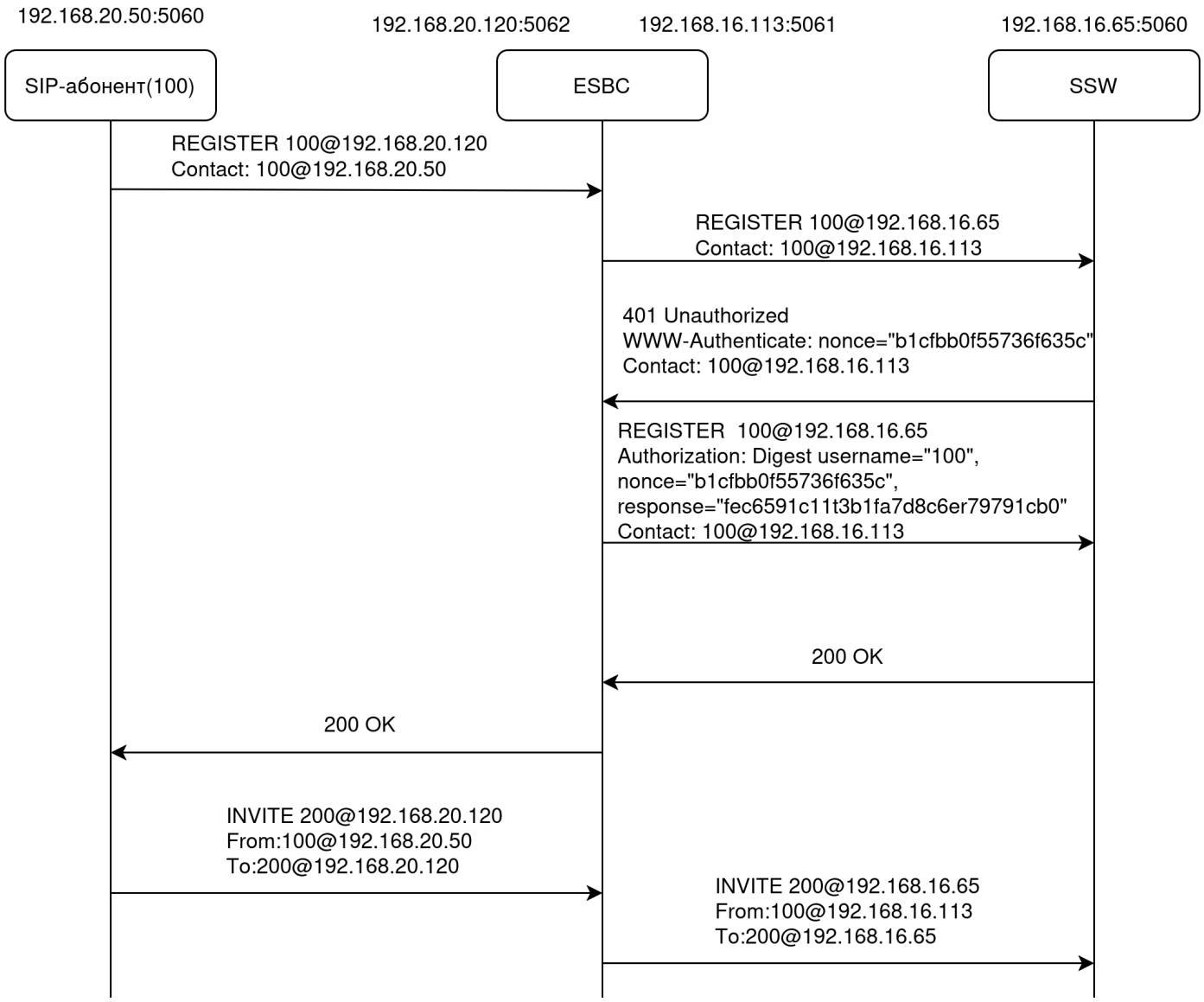
```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_ABONENTS
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW

#Если абоненты находятся за NAT, выполнить команду:
vesbc(config-esbc-user-interface-sip)# nat comedia-mode on
```

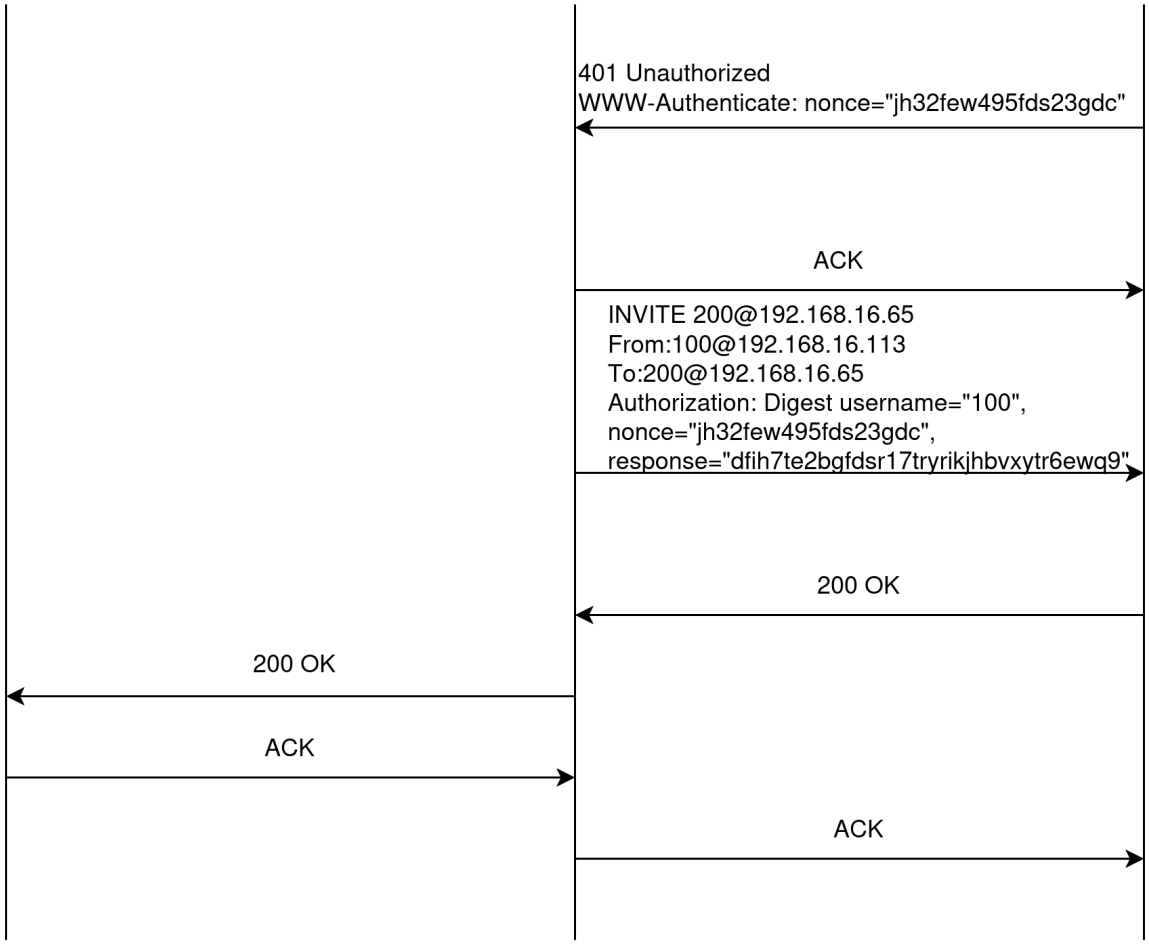
12. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

Теперь при регистрации этого абонента, если с Softswitch придёт 401 ответ, то ESBC локально его обработает и отправит новый запрос регистрации с аутентификационными данными из профиля учётных данных. То же самое произойдет при вызове.

Пример работы:





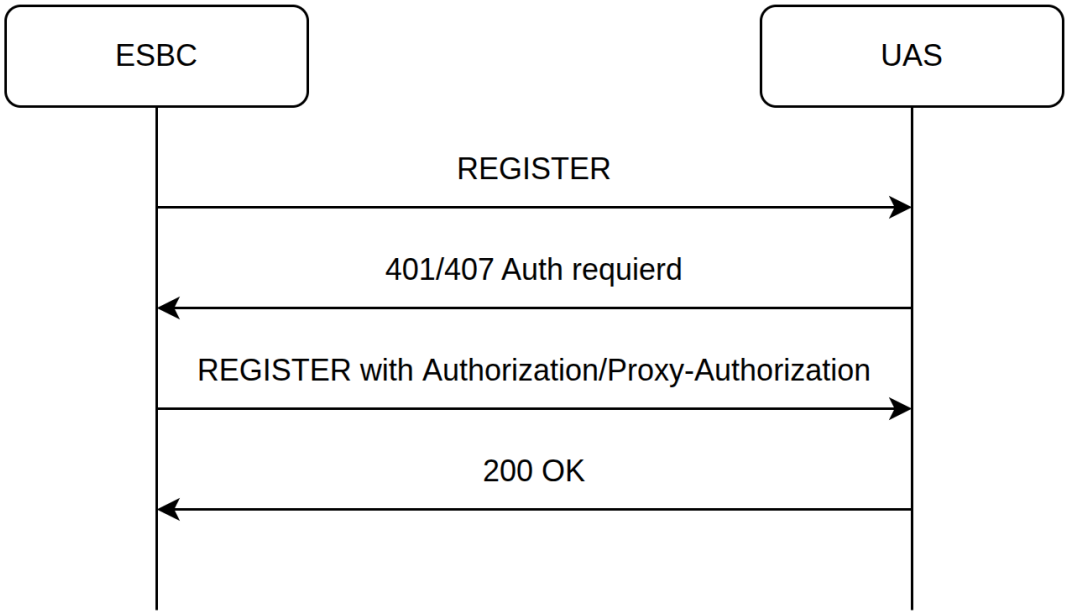
9.13.3 Настройка клиентской регистрации транка

Для повышения безопасности подключения к внешней среде используется клиентская регистрация транка ESBC на сервере регистрации провайдера. Для этого используются аутентификационные данные, которые выдаёт провайдер.
При этом через транк разрешаются:

1. Исходящие вызовы с зарегистрированных номеров на любые.
2. Входящие вызовы с любых номеров на зарегистрированные.

При отсутствии регистраций направление будет считаться недоступным.

Схема работы:



Алгоритм настройки транка для работы функции:

1. Добавить аутентификационные данные в профиль учётных данных.
2. Привязать профиль учётных данных к AAA-профилю.
3. Привязать AAA-профиль к транку.
4. Включить клиентский режим работы регистрации транка.

ESBC начнёт самостоятельно отправлять запросы регистрации, используя данные из привязанного профиля учётных данных.

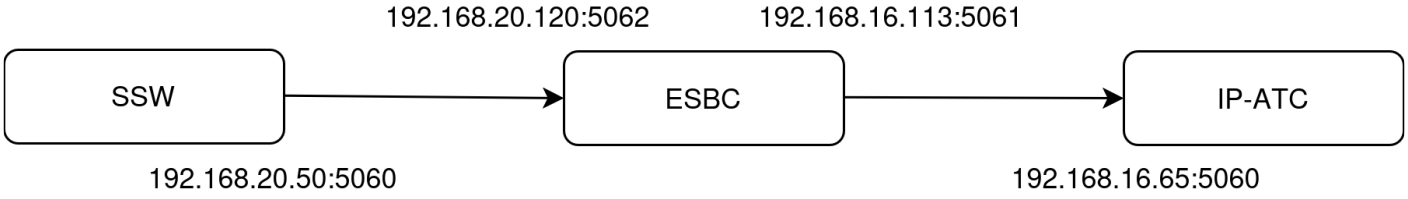
Рекомендуется использовать режим только для схемы передачи транк-транк.

Если настроен абонентский интерфейс с маршрутизацией в транк, на котором включена клиентская регистрация, то через него не будут проходить запросы регистрации от абонентов.

Пример настройки клиентской регистрации транка:

Задача:

Настроить связь между Softswitch и IP-АТС провайдера через ESBC. При этом провайдер выдал аутентификационные данные: номер 100, логин 100, пароль PASSWORD, домен DOMAIN.loc.



Для решения задачи на ESBC нужно настроить транк для Softswitch и транк с клиентской регистрацией для IP-АТС провайдера.

Решение:

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на внешнем интерфейсе в сторону IP-АТС провайдера:

```
vesbc#
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "IP"
vesbc(config-if-gi)# ip address 192.168.16.113/24
```

2. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.20.120/24
vesbc(config-if-gi)# ip firewall disable
```

3. Создать SIP-транспорт в сторону сервера регистрации провайдера:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_IP
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5061
```

4. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# port 5062
```

5. Создать медиаресурсы для согласования и передачи голоса на плече ESBC — IP-АТС:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_IP
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC — SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать профиль учётных данных с аутентификационными данными транка:

```
vesbc#
vesbc# config
vesbc(config)# esbc
vesbc(config-esbc)# credential profile CREDENTIAL_PROFILE
vesbc(esbc-credential-profile)# number 100
vesbc(esbc-credential-profile-cred)# login 100
vesbc(esbc-credential-profile-cred)# password PASSWORD
```

Подробное описание команд настройки профиля учётных данных на ESBC представлено в настройках профиля учетных данных в CLI.

> В один профиль можно добавить до 2500 номеров.

8. Создать AAA профиль с созданным профилем учётных данных:

```
vesbc#
vesbc# config
vesbc(config)# esbc
vesbc(config-esbc)# aaa profile AAA_PROFILE
vesbc(config-aaa-profile)# credential local profile CREDENTIAL_PROFILE
```

9. Создать SIP-транк в сторону IP-АТС и привязать к нему настроенный AAA профиль, а также выданный провайдером домен:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_IP
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_IP
vesbc(config-esbc-trunk-sip)# aaa profile AAA_PROFILE
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.65
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_IP
vesbc(config-esbc-trunk-sip)# domain DOMAIN.loc
```

10. Создать SIP-транк в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.20.50
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

11. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с SSW, будут маршрутизироваться на IP-АТС:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_IP
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_IP
```

12. Привязать созданную таблицу маршрутизации к транку смотрящему в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# route-table TO_IP
```

13. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с IP-АТС, будут маршрутизироваться на SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW
```

14. Привязать созданную таблицу маршрутизации к транку, смотрящему в сторону IP-АТС:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_IP
vesbc(config-esbc-trunk-sip)# route-table TO_SSW
```

15. На транке, смотрящем на IP-АТС, задать параметры регистрации и включить клиентскую регистрацию:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_IP

#Выбираем время, на которое будет регистрироваться транк (по умолчанию 3600 секунд):
vesbc(config-esbc-trunk-sip)# registration expires 900

#Выбираем время, через которое начнётся повторная отправка регистрации после неудачной попытки
 (по умолчанию 60 секунд):
vesbc(config-esbc-trunk-sip)# registration retry-after 40

#Включаем режим клиентской регистрации на транке:
vesbc(config-esbc-trunk-sip)# registration mode client
```

16. Применить конфигурацию и подтвердить изменения:

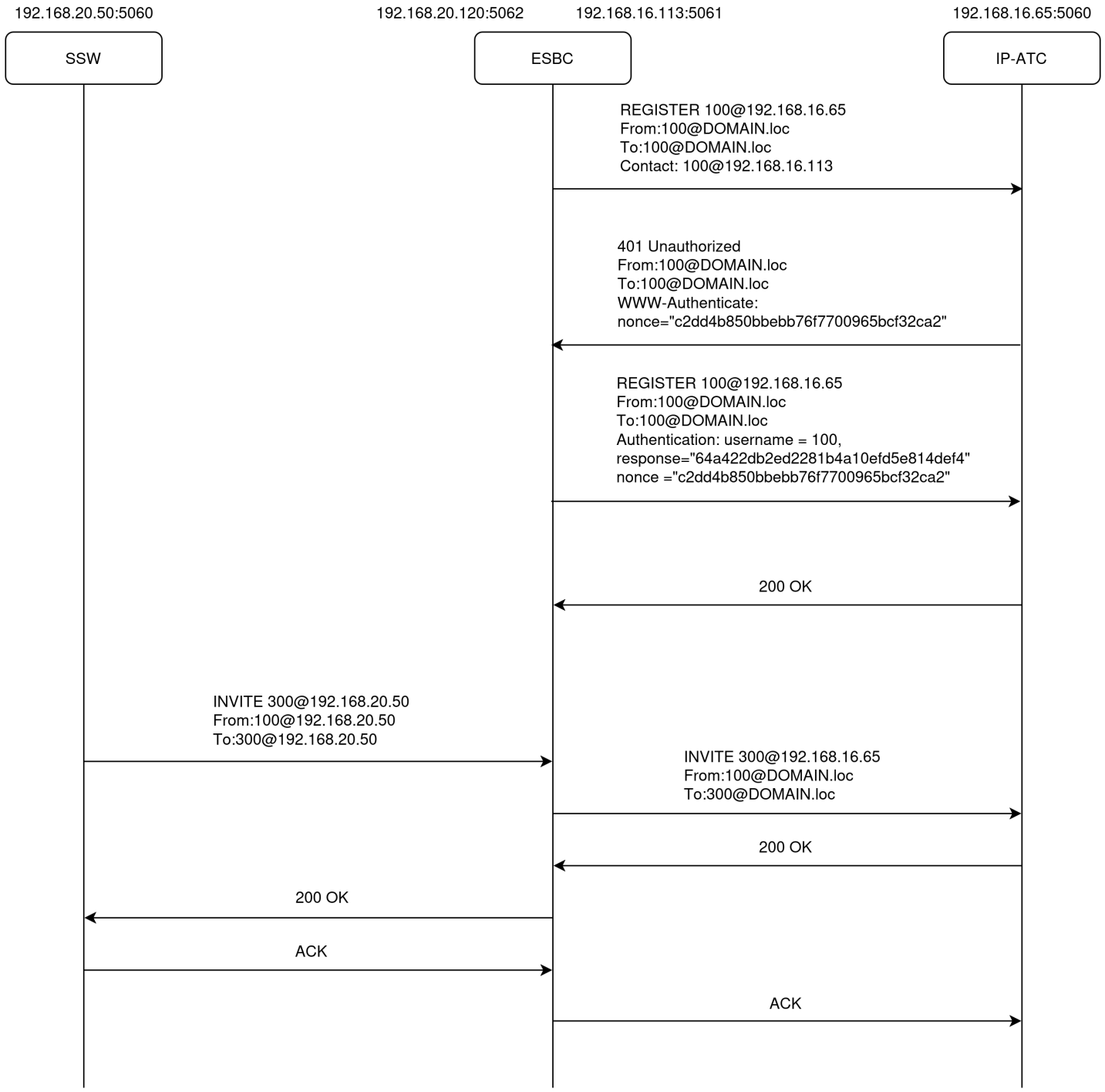
```
vesbc# commit
vesbc# confirm
```

17. Посмотреть активные регистрации транка можно командой show esbc trunks:

```
vesbc# show esbc trunks sip TRUNK_IP registration

Registration type: Client
Total registrations: 1
Active registrations: 1
AOR              Contact                          Status           Expires    Expires in
--------------   ------------------------------   --------------   --------   ----------
100@DOMAIN.loc   sip:100@192.168.16.113:5061      Registered       900        867
```

После применения конфигурации с транка отправится запрос регистрации (REGISTER) на IP-АТС провайдера. Если придёт ответ, требующий аутентификацию (401,407), то ESBC использует локальную аутентификацию запросов и ответит на него, используя данные из профиля учётных данных. После успешной регистрации транка, исходящие вызовы с него возможны на любые номера только с **зарегистрированного номера**. Входящие вызовы будут обработаны с любого номера только на **зарегистрированный номер**.

Пример работы:



9.14 Настройка NAT

NAT (Network Address Translation — «преобразование сетевых адресов») — это технология, которая позволяет преобразовывать локальные IP-адреса внутренней сети в один или несколько глобальных IP-адресов и обратно. NAT обеспечивает механизм подключения областей с приватными адресами к внешним областям, в которых используются уникальные в глобальном масштабе зарегистрированные адреса. Кроме того, NAT осуществляет преобразование портов — то есть меняет номера портов в пакетах данных, что помогает однозначно сопоставлять исходящие и входящие соединения.

В данном разделе описаны различные функции для работы с NAT, а также настройки ESBC для его обхода, такие как:

1. Настройка NAT comedia-mode для прохождения медиатрафика через устройства NAT;
2. Настройка Public-IP;
3. Настройка внешнего STUN-сервера;
4. Настройка локального STUN-сервера.

9.14.1 Настройка NAT comedia-mode

С целью преодоления соединений через устройства NAT, в ESBC реализована поддержка nat comedia-mode для транков и абонентских интерфейсов.

Механизм NAT comedia-mode используется в том случае, когда **встречное** устройство находится за NAT.

Настройка и принцип работы nat comedia-mode для транков (trunk)

Включение режима nat comedia-mode осуществляется в настройках транка:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip
vesbc(config-esbc-trunk-sip)# nat comedia-mode
    Select NAT comedia mode for trunk:
        off
        on
        flexible

vesbc(config-esbc-trunk-sip)# nat comedia-mode on
```

Реализована работа в двух режимах:

- *flexible* — проверяет источник во входящем RTP-потоке и транслирует исходящий поток на IP-адрес и UDP-порт, с которого принимается медиапоток. В случае прерывания входящего RTP-потока более чем на 1 секунду, исходящий поток продолжает транслироваться;
- *on* — проверяет источник во входящем RTP-потоке и транслирует исходящий поток на IP-адрес и UDP-порт, с которого принимается медиапоток. В случае прерывания входящего RTP-потока более чем на 1 секунду, исходящий поток перестает транслироваться.

Принцип работы

При включении режима nat comedia-mode on/flexible отправка и получение сообщений сигнализации SIP осуществляется с/на IP-адрес и порт, указанные в настройках транка.

Отправка RTP будет осуществляться на IP-адрес и порт, с которого был получен первый RTP-пакет от встречной стороны. До получения RTP от встречной стороны, медиатрафик **не будет передаваться** от ESBC в эту сторону.

Настройка и принцип работы nat comedia-mode для абонентов (user-interface)

Включение режима nat comedia-mode осуществляется в настройках абонентского интерфейса:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip USERS
vesbc(config-esbc-user-interface-sip)# nat comedia-mode
    Select NAT comedia mode for user-interface:
        off
        on
        flexible
vesbc(config-esbc-user-interface-sip)# nat comedia-mode on
```

Реализована работа в двух режимах:

- *flexible* — проверяет источник во входящем RTP-потоке и транслирует исходящий поток на IP-адрес и UDP-порт, с которого принимается медиапоток, в случае прерывания входящего RTP-потока более чем на 1 секунду, исходящий поток продолжает транслироваться;
- *on* — проверяет источник во входящем RTP-потоке и транслирует исходящий поток на IP-адрес и UDP-порт, с которого принимается медиапоток, в случае прерывания входящего RTP-потока более чем на 1 секунду, исходящий поток перестает транслироваться.

Принцип работы.

При включении режима nat comedia-mode on/flexible отправка сообщений сигнализации SIP осуществляется симметрично (на IP-адрес и порт, с которого был принят запрос) в случае, если клиент в инициирующем запросе не использовал параметр RPORT.

Отправка сообщений сигнализации SIP на зарегистрированного абонента будет осуществляться на IP-адрес и порт, с которого был принят запрос REGISTER, а не на данные абонента, указанные в заголовке Contact.

Отправка RTP будет осуществляться на IP-адрес и порт, с которого был получен первый RTP-пакет от встречной стороны. До получения RTP от встречной стороны, медиатрафик **не будет передаваться** от ESBC в эту сторону.

Команда *nat keep-alive-interval* в настройках абонентского интерфейса используется для настройки интервала для поддержки соединения за NAT. При включении опции, абоненту с заданным интервалом будут отправляться пакеты с содержанием "0d0a" для предотвращения разрушения сессии на NAT-маршрутизаторе.

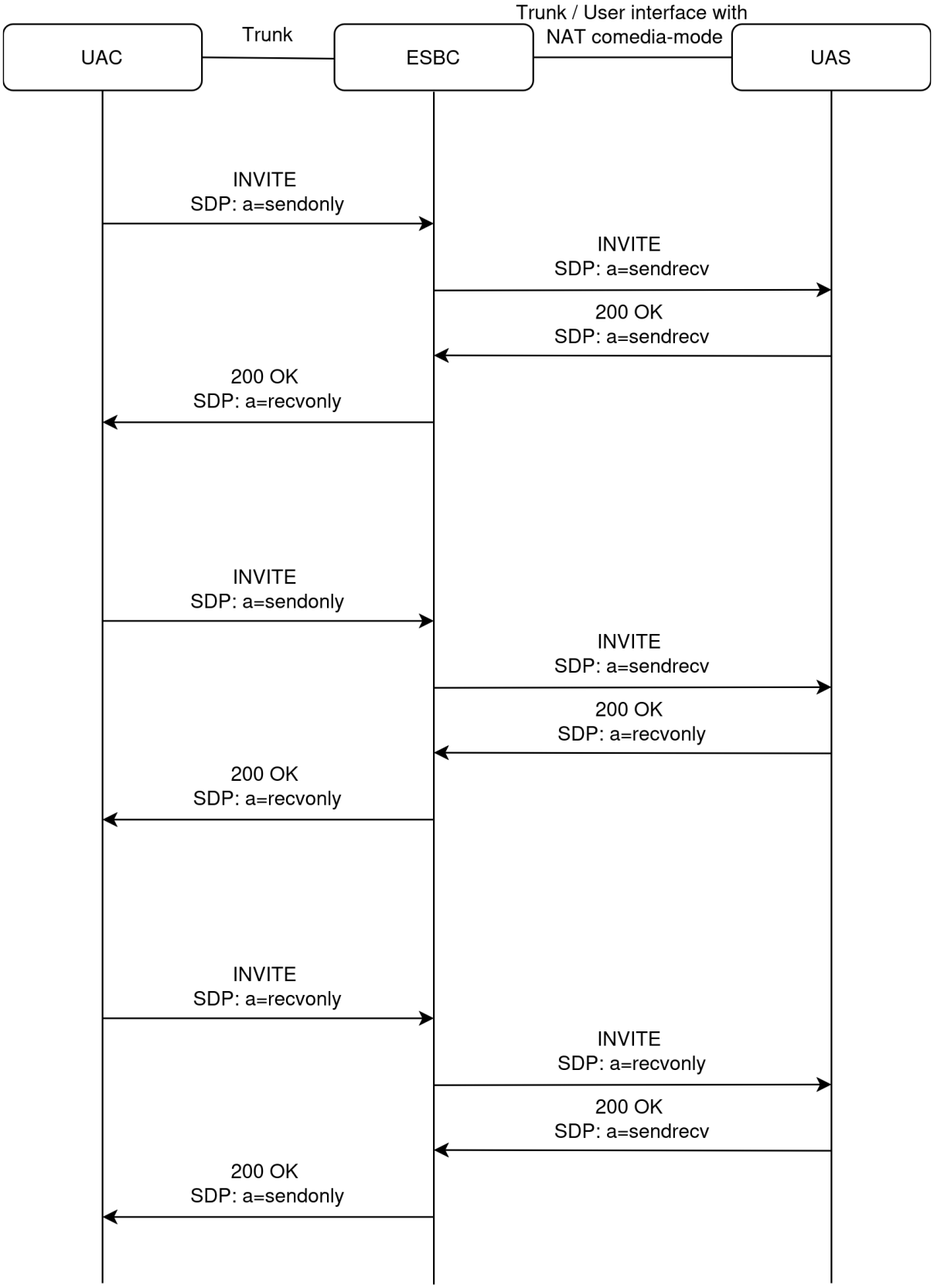
Подмена атрибутов direction в SDP

При включении опции *nat comedia-mode* атрибут direction sendonly в SDP при отправке offer/answer sdp заменяется на sendrecv.

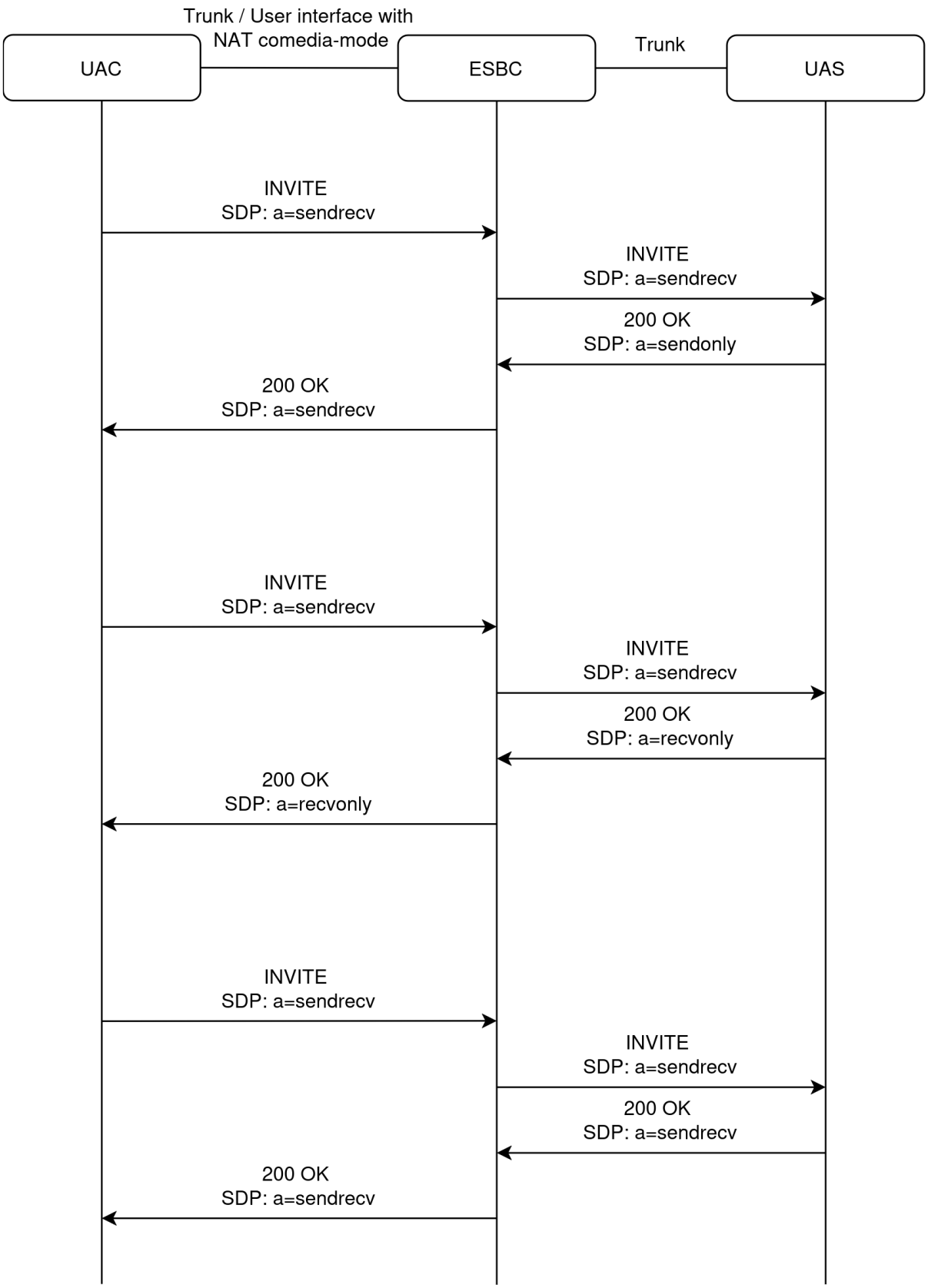
Данный механизм используется для предотвращения ситуации, в которой абонент за NAT не начнет первым отправку RTP-пакетов в сторону ESBC, и соответственно ESBC не начнет отправку встречного потока RTP к абоненту.

Примеры:

1. Замена атрибутов direction в offer sdp:



2. Замена атрибутов direction в answer sdp:



9.14.2 Настройка Public IP

Public IP (рус. «публичный IP-адрес») — это внешний IP-адрес, который используется при отправке запросов пользователю или удаленному адресу из внешней сети.

Настройка используется в случае, когда ESBC не имеет публичного IP-адреса и выход в публичную сеть осуществляется через NAT. В таком случае в качестве Public IP указывается адрес WAN-интерфейса NAT для подстановки в сигнальные сообщения протокола SIP.

Public IP можно настроить для абонентского интерфейса, транка и транковой группы.

> Если Public IP настроен в транке и в транковой группе, в которую входит этот транк, то будет использоваться Public IP из настроек транка.

> В качестве публичного адреса можно использовать как IPv4, так и IPv6 (не поддерживается в текущей версии ПО) адрес.

При наличии Public IP, адреса в SDP, заголовках Via и Contact будут заменены на значение public-ip из конфигурации объекта. Media будет работать в режиме NAT-comedia.

> Если на направлении настроен Public IP и внешний STUN-сервер, то в качестве публичного будет использоваться адрес, полученный от STUN-сервера.

> Для корректной работы опции Public IP необходимо организовать проброс портов для сигнализации SIP и медиапортов RTP на вышестоящем устройстве NAT "один к одному".

Пример настройки Public IP для транка:

```
#Настройка SIP-транспорта для транка:
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRUNK_TRANSPORT
vesbc(config-esbc-sip-transport)# ip address 192.168.1.1
vesbc(config-esbc-sip-transport)# exit

#Настройка медиаресурсов для транка:
vesbc(config-esbc)# media resource TRUNK_MEDIA
vesbc(config-esbc-media-resource)# ip address 192.168.1.1
vesbc(config-esbc-media-resource)# exit

#Настройка параметров транка:
vesbc(config-esbc)# trunk sip TRUNK_PUBLIC_IP
vesbc(config-esbc-trunk-sip)# sip transport TRUNK_TRANSPORT
vesbc(config-esbc-trunk-sip)# media resource 0 TRUNK_MEDIA
vesbc(config-esbc-trunk-sip)# remote address 192.168.1.3
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# public-ip 10.25.0.1

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Пример использования Public IP:

ESBC получает сообщение, которое должно быть смаршрутизировано в транк TRUNK_PUBLIC_IP:

```
INVITE sip:23002@192.168.1.1:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.4:5061;rport;branch=z9hG4bK-1914230-1-1
From: "24001" <sip:24001@192.168.1.4:5061>;tag=1
To: "23002" <sip:23002@192.168.1.1:5060>
Call-ID: 1-1914230@192.168.1.4
Cseq: 1 INVITE
Contact: <sip:24001@192.168.1.4:5061>
Max-Forwards: 70
Allow: INVITE, ACK, BYE, CANCEL
Content-Type: application/sdp
Content-Length: 138

Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): user1 77755765 7773687637 IN IP4 192.168.1.4
Session Name (s): -
Time Description, active time (t): 0 0
Connection Information (c): IN IP4 192.168.1.4
Media Description, name and address (m): audio 10000 RTP/AVP 8
Media Attribute (a): rtpmap:8 PCMA/8000
```

ESBC пересылает INVITE через транк TRUNK_PUBLIC_IP на встречное устройство.
В SDP, Via и Contact вместо адреса используемого SIP-транспорта (192.168.1.1) указывается адрес Public IP транка (10.25.0.1):

```
INVITE sip:23002@192.168.1.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.25.0.1:5060;rport;branch=z9hG4bKPj6e357f04-e13e-4ead-8386-2246d12450b4
Max-Forwards: 70
From: "24001" <sip:24001@192.168.1.1>;tag=76776c9a-022b-4ccf-9458-e83e2701f6c8
To: "23002" <sip:23002@192.168.1.3>
Contact: <sip:24001@10.25.0.1:5060;transport=udp>
Call-ID: 5fc229f6657d7706f2b6c81a44a5b10e
CSeq: 28491 INVITE
Allow: INVITE, ACK, BYE, CANCEL
Supported: 100rel, replaces, ice
Content-Type: application/sdp
Content-Length:   135

Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): user1 77755765 7773687637 IN IP4 10.25.0.1
Session Name (s): -
Time Description, active time (t): 0 0
Connection Information (c): IN IP4 10.25.0.1
Media Description, name and address (m): audio 10000 RTP/AVP 8
Media Attribute (a): rtpmap:8 PCMA/8000
```

9.14.3 Настройка STUN

STUN – сетевой протокол, позволяющий клиенту, находящемуся за NAT, определить свой публичный (внешний) IP-адрес и порт для обеспечения установления и корректной передачи медиатрафика через NAT. Применяется в механизмах обхода NAT (например, ICE) для установления прямых медиасоединений.

ESBC поддерживает STUN в двух режимах работы:

- взаимодействие с внешними STUN-серверами;
- функционирование в качестве STUN-сервера для клиентов за NAT.

Настройка внешнего STUN-сервера

ESBC поддерживает взаимодействие с внешними STUN-серверами для определения собственного публичного адреса (например, если ESBC находится в частной сети за NAT-устройством). ESBC отправляет STUN Binding Request на указанный сервер и использует полученный в ответе IP-адрес и порт в качестве публичного, подставляя его в сигнальные сообщения протокола SIP. Для получения портов для RTP/RTCP также будут отправляться STUN Binding Request в момент установления вызова.

По умолчанию ESBC взаимодействует с внешними STUN-серверами в режиме согласно RFC 3489 (Simple Traversal of UDP through NATs или "classic STUN") – использует в качестве публичного адреса и порта значения, полученные в атрибуте MAPPED-ADDRESS в Binding Response от STUN-сервера.
Доступна поддержка RFC 5389 (Session Traversal Utilities for NAT) – в этом случае ESBC в Binding Request на STUN-сервер будет отправлять Cookie, с помощью которого сервер зашифрует адрес и порт и передаст их в Binding Response в атрибуте XOR-MAPPED-ADDRESS, ESBC дешифрует полученные данные и использует в качестве публичного адреса и порта.

Доступна настройка интервала отправки запросов на внешний STUN-сервер.

Внешний STUN-сервер можно использовать в настройках абонентского интерфейса, транка и транковой группы.

> Если STUN-сервер настроен в транке и в транковой группе, в которую входит этот транк, то будет использоваться STUN-сервер из настроек транка.

Если на каком-либо направлении настроено получение публичного адреса через внешний STUN-сервер и этот сервер недоступен, то:

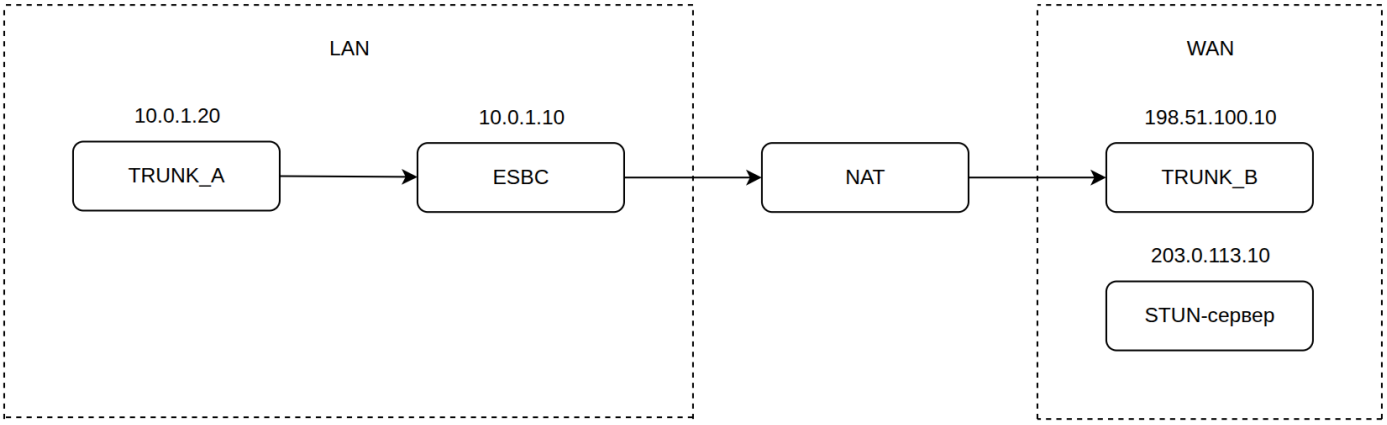
- при наличии настроенного Public IP в SIP-сообщениях будет использоваться адрес из Public-IP;
- если Public IP не настроен, то SIP запросы на это направление будут отклонены ошибкой 503.

Подробное описание команд настройки внешних STUN-серверов на ESBC представлено в разделе Настройки STUN Справочника команд CLI.

Пример настройки ESBC для взаимодействия со STUN-сервером:

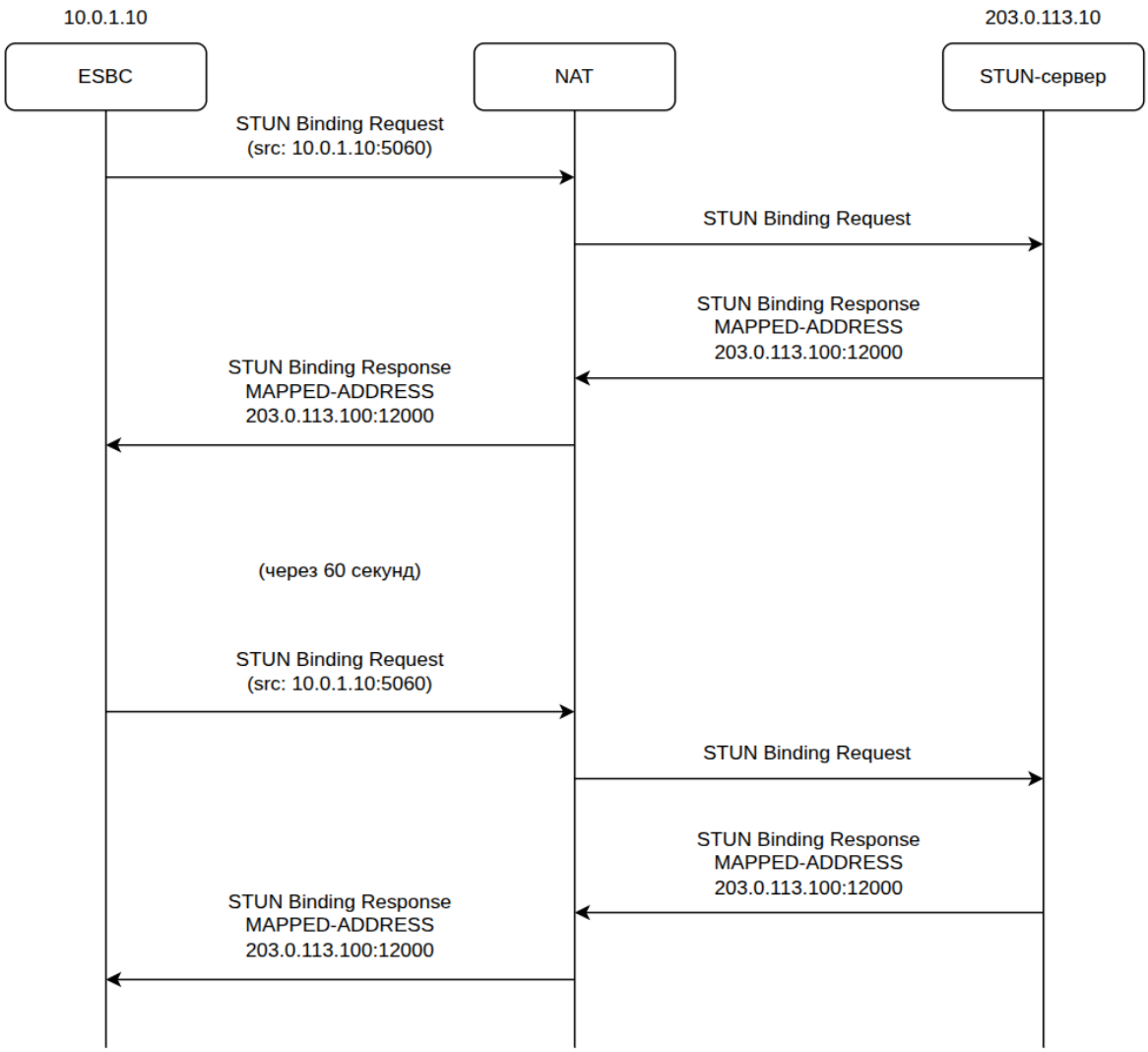
Схема:

ESBC находится в частной сети за NAT устройством, настроены 2 транка — TRUNK_A (в той же сети, что и ESBC) и TRUNK_B (в публичной сети), настроена маршрутизация между ними, адрес WAN-интерфейса NAT-устройства неизвестен. В публичной сети находится STUN-сервер. Необходимо настроить получение публичного адреса через STUN-сервер для вызовов, направленных в TRUNK_B.

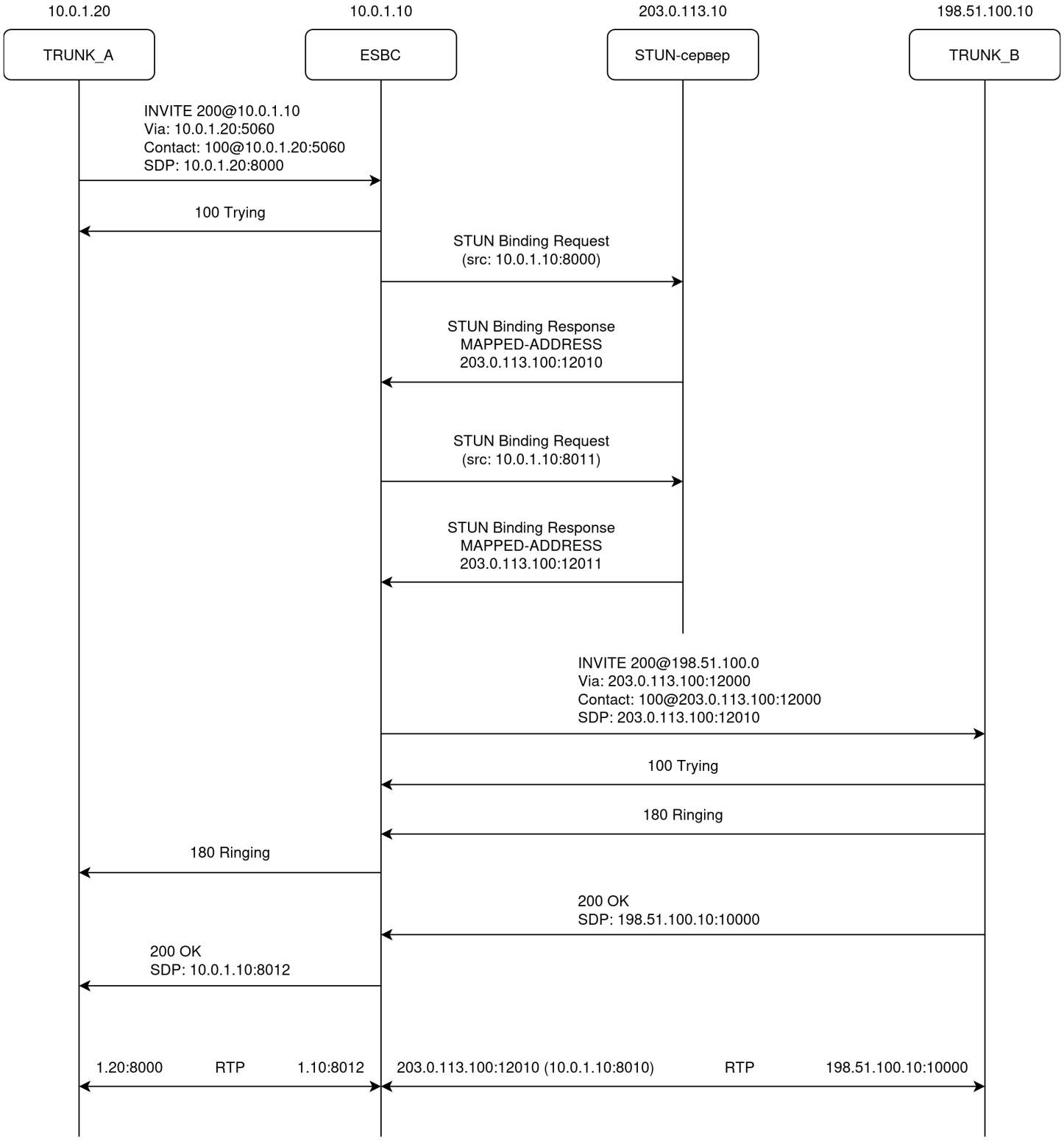



```
#Настройка внешнего STUN-сервера:
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# stun server external EXTERNAL_STUN
vesbc(config-esbc-stun-server-ext)# remote address 203.0.113.10

#Интервал отправки запросов на STUN-сервер = 60 секунд:
vesbc(config-esbc-stun-server-ext)# keepalive interval 60
vesbc(config-esbc-stun-server-ext)# exit

#Добавление STUN-сервера в настройки транка:
vesbc(config-esbc)# trunk sip TRUNK_B
vesbc(config-esbc-trunk-sip)# stun server EXTERNAL_STUN

#Применение и подтверждение изменений:
vesbc(config-esbc-stun-trunk-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-stun-trunk-sip)# do confirm
Configuration has been confirmed. Commit timer canceled
```

После успешного применения конфигурации ESBC начнёт каждые 60 секунд отправлять Binding Request на STUN-сервер с порта, который указан в привязанном к транку TRUNK_B SIP-транспорте, полученный в ответе адрес и порт будет использоваться в SIP-сообщениях на плече ESBC–TRUNK_B.



При исходящем вызове в транк TRUNK_B ESBC отправит запросы на STUN-сервер с RTP и RTCP-порта для согласования обмена медиатрафиком.



При входящем вызове из транка TRUNK_B ESBC также отправит запросы на STUN-сервер с RTP и RTCP-порта для согласования обмена медиатрафиком перед отправкой SDP-Answer инициатору.

Настройка локального STUN-сервера

ESBC может функционировать в качестве STUN-сервера для абонентов, находящихся за NAT. Для этого на ESBC настраиваются локальные STUN-серверы (до 64 шт.).

При получении STUN Binding Request, ESBC формирует Binding Success Response, содержащий IP-адрес и порт, с которого данный запрос был получен. Этот адрес называется Reflexive Address и может использоваться клиентом для корректного формирования SIP-сообщений, SDP и обеспечения прохождения медиатрафика (RTP) через NAT.

Поддерживается работа протокола STUN в режимах Classic STUN (RFC 3489) и современный STUN (RFC 5389+).

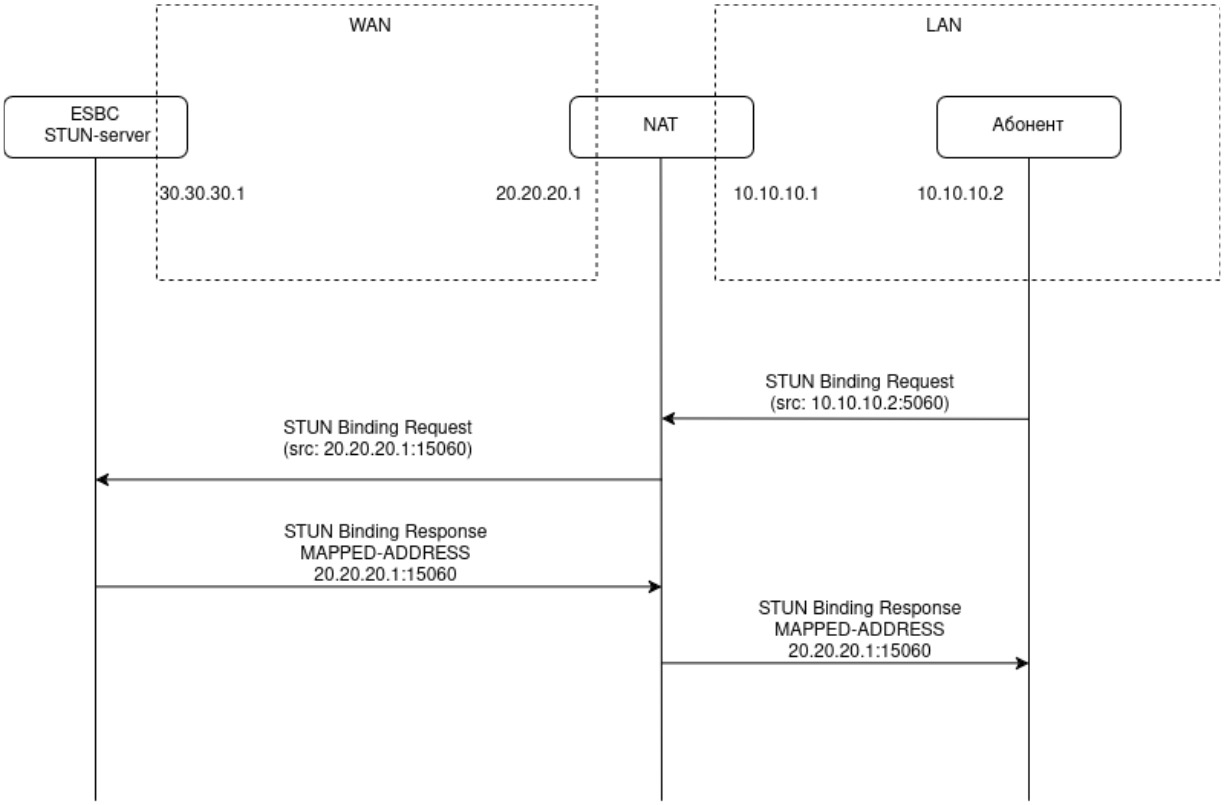
Подробное описание команд настройки локальных STUN-серверов на ESBC представлено в разделе Настройки STUN Справочника команд CLI.

Пример настройки локального STUN-сервера в конфигурации:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# stun server local LOCAL_STUN
vesbc(config-esbc-stun-server-loc)# ip address 30.30.30.1
vesbc(config-esbc-stun-server-loc)# port 13478
vesbc(config-esbc-stun-server-loc)#

#Применение и подтверждение изменений:
vesbc(config-esbc-stun-server-loc)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-stun-server-loc)# do confirm
Configuration has been confirmed. Commit timer canceled
```

После описанной выше настройки ESBC будет принимать сообщения STUN на IP-адрес 30.30.30.1 и порт UDP 13478.

> ⚠ При использовании в конфигурации ESBC локального STUN-сервера не требуется дополнительной настройки firewall, необходимые правила для пропуска входящих STUN-запросов будут настроены автоматически.



Когда абонент, находящийся за NAT, отправит STUN-запрос, который будет получен ESBC с адреса 20.20.20.1:15060 на адрес ESBC и порт локального STUN-сервера (30.30.30.1:13478), в ответе ESBC укажет 20.20.20.1:15060 в качестве Reflexive Address.

9.15 Настройка QoS

QoS (Quality of Service, рус. «качество обслуживания») — технология предоставления различным классам данных различных приоритетов в обслуживании.

Приоритет определяется значением DSCP (0-63) в поле IP-заголовка DS.

Установить необходимое значение DS можно отдельно для:

- аудиопакетов;
- видеопакетов;
- пакетов сигнализации SIP.

Параметры QoS настраиваются в конфигурации абонентского интерфейса, транка и транковой группы.

> Если QoS настроен для транка и для транковой группы, в которую входит этот транк, то будет использоваться QoS из настроек транка.

Пример настройки QoS для аудиотрафика в конфигурации абонентского интерфейса:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENT_QOS_50
vesbc(config-esbc-user-interface-sip)# sip transport ABONENT_TRANSPORT
vesbc(config-esbc-user-interface-sip)# media resource 0 ABONENT_MEDIA
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
vesbc(config-esbc-user-interface-sip)# dscp audio 50

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-user-interface-sip)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

В исходящем аудиотрафике абонентам, зарегистрированным через интерфейс ABONENT_QOS_50, поле DS в IP-пакете будет выглядеть следующим образом:

```
Differentiated Services Field: 0xc8 (DSCP: Unknown, ECN: Not-ECT)
    1100 10.. = Differentiated Services Codepoint: Unknown (50)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
```

9.16 Контроль трафика

9.16.1 Контроль входящего трафика

На ESBC реализован контроль интенсивности входящего трафика для:

- одновременных вызовов (max in calls);
- вызовов в секунду (max in cps);
- регистраций в секунду (max in regps);
- запросов в секунду (max in rps);
- SIP-сообщений в секунду в рамках одной сессии (max in session pps);
- подписок в секунду (max in subps);
- общего количества активных подписок (max in subscriptions);
- запросов в интервал времени с зарегистрированного абонента (max in user rpp);
- контактов на одного абонента (max in user contacts).

Описание всех команд для контроля входящего трафика приведено в разделах для каждого объекта в CLI:

1. general;
2. trunk;
3. trunk-group;
4. user-interface.

Реализация ограничения входящего трафика поддерживается для следующих объектов:

max in ...	Для всей системы (general)	Транков (trunk)	Транк-групп (trunk-group)	Абонентских интерфейсов (user-interface)
calls	✔	✔	✔	✔
cps	✔	✔	✔	✔
regps	✔	✘	✘	✔
rps	✔	✔	✔	✔
session pps	✔	✔	✔	✔
subps	✔	✔	✔	✔
subscriptions	✔	✔	✔	✔
user rpp	✔	✘	✘	✔
user contacts	✘	✘	✘	✔

Ограничения **max in user rpp** и **max in session pps**, указанные в разделе general, будут применяться для каждого объекта (зарегистрированного абонента, установленной сессии) системы **независимо**. Таким образом, например при настройке **max in user rpp 100 60,** ограничение в 100 запросов за 60 секунд будет применяться к каждому зарегистрированному абоненту.
Приоритет использования данных ограничений:
1) если в настройках абонентского интерфейса (UI) не указано ограничение max in user rpp/max in session pps, то будут использоваться значения, указанные в настройке системы (General);
2) значения max in user rpp/max in session pps в настройках абонентского интерфейса (UI) могут быть больше, чем значения, указанные в настройке системы (General). В таком случае для всех абонентов, зарегистрированных через данный абонентский интерфейс, будут использоваться ограничения, указанные в его настройках.
Все остальные ограничения являются **глобальными** и устанавливают ограничения на **всю** систему. Поэтому ограничения для отделных объектов системы (транков, транковых групп, абонентских интерфейсов) не могут превышать общесистемные.

Приоритет использования ограничений:

1. Ограничения для всей системы. Настройки ограничений **general** переопределяют настройки ограничений остальных объектов (trunk, trunk-group, user-interface) ESBC.
2. Ограничения для транк-группы. Настройки ограничений **trunk-group** переопределяют настройки ограничений транков, входящих в данную транк-группу, но переопределяются настройками **general.**
3. Ограничения для транков и абонентских интерфейсов. Настройки ограничений **trunk** переопределяются настройками ограничений транк-группы (если транк входит в состав какой-либо транк-группы) или настройками **general.** Настройки ограничений **user-interface** переопределяются настройками **general.**

> ⚠ Приоритет использования ограничений распространяется только на максимальные значения, т. е. например, при использовании max in cps 100 для всей системы (general) и использовании max in cps 10 для какого-либо транка, ограничение обработки CPS для этого транка будет 10, а не 100.

При одновременном использовании **max in cps** и **max in rps** в настройках транка, транковой группы или абонентского интерфейса, максимальным порогом для всех входящих запросов будет являться значение, настроенное в **max in rps**. При этом ограничение для входящих запросов INVITE можно уменьшить отдельно настройкой **max in cps**.

Пример:

1. В конфигурации настроено max in cps 10 и max in rps 5. В таком случае максимальное ограничение на входящие вызовы (INVITE) будет 5.
2. В конфигурации настроено max in cps 5 и max in rps 10. В таком случае максимальное ограничение на входящие вызовы будет 5, а на остальные запросы — 10.

Пример работы приоритета системных ограничений над абонентским интерфейсом:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Установка ограничения входящего максимального количества CPS для всей системы – 5:
vesbc(config-esbc-general)# max in cps 5
vesbc(config-esbc-general)# exit

#Установка ограничения максимального входящего количества CPS для отдельного абонентского
интерфейса (USER_IFACE) – 10:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface-sip)# max in cps 10

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface-sip)# do commit
vesbc(config-esbc-user-interface-sip)# do confirm
```

В данном случае не будет обрабатываться более 5 вызовов в секунду, приходящих на абонентский интерфейс USER_IFACE, так как у общесистемных ограничений приоритет выше.

Пример работы приоритета ограничений транк-группы над ограничениями транка, входящего в эту транк-группу:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки транка:
vesbc(config-esbc)# trunk sip TRUNK
vesbc(config-esbc-trunk-sip)#

#Установка ограничения входящего максимального количества CPS для транка – 50:
vesbc(config-esbc-trunk-sip)# max in cps 50
vesbc(config-esbc-trunk-sip)# exit

#Переход в настройки транк-группы:
vesbc(config-esbc)# trunk-group TRUNKGROUP

#Добавление транка TRUNK в состав транк-группы:
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK

#Установка ограничения входящего максимального количества CPS для транк-группы – 10:
vesbc(config-esbc-trunk-group)# max in cps 10

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-group)# do commit
vesbc(config-esbc-trunk-group)# do confirm
```

Т. к. транк TRUNK входит в состав транк-группы TRUNKGROUP, то будет обрабатываться не более 10 входящих вызовов в секунду, поступающих в транк, так как приоритет ограничения у транк-группы выше.

Пример работы ограничения количества одновременных вызовов на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Установка ограничения максимального количества одновременных вызовов для абонентского-
интерфейса – 10:
vesbc(config-esbc-user-interface-sip)# max in calls 10

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество одновременных вызовов, проходящих через абонентский интерфейс USER_IFACE, не может быть больше 10. Все запросы INVITE, поступающие после превышения лимита, будут проигнорированы.

Пример работы ограничения количества вызовов в секунду на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Установка ограничения входящего максимального количества CPS для абонентского-интерфейса — 10:
vesbc(config-esbc-user-interface-sip)# max in cps 10

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество вызовов, проходящих через абонентский интерфейс USER_IFACE в секунду, не может быть больше 10. Все запросы INVITE, поступающие после превышения лимита, будут проигнорированы.

Пример работы ограничения количества регистраций в секунду на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества регистраций в секунду:
vesbc(config-esbc-user-interface-sip)# max in regps 15

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество регистраций в секунду, проходящих через абонентский интерфейс USER_IFACE, не может быть больше 15. Все запросы REGISTER, поступающие после превышения лимита, будут проигнорированы.

Пример работы ограничения количества запросов в секунду на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества запросов в секунду:
vesbc(config-esbc-user-interface-sip)# max in rps 250

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество запросов, проходящих через абонентский интерфейс USER_IFACE, не может быть больше 250. Все запросы (INVITE, REGISTER и прочие), поступающие после превышения лимита, будут проигнорированы.

Пример работы ограничения количества SIP-сообщений в секунду в рамках одной сессии на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества SIP-сообщений в секунду в рамках одной сессии в секунду:
vesbc(config-esbc-user-interface-sip)# max in session pps 30

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество SIP-сообщений, проходящих через абонентский интерфейс USER_IFACE в секунду, в рамках одной сессии, не может быть больше 30. При достижении лимита сессия будет завершена.

Пример работы ограничения количества подписок в секунду на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества подписок в секунду:
vesbc(config-esbc-user-interface-sip)# max in subps 50

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit it
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество запросов SUBSCRIBE, проходящих через абонентский интерфейс USER_IFACE, не может быть больше 50. Все запросы SUBSCRIBE, поступающие после превышения лимита, будут проигнорированы.

Пример работы ограничения количества общего количества активных подписок на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества активных подписок:
vesbc(config-esbc-user-interface-sip)# max in subscriptions 1000


#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество активных подписок (у которых не истек Expires), подтвержденных через абонентский интерфейс USER_IFACE, не может превышать 1000. Все запросы SUBSCRIBE, поступающие после превышения лимита, будут проигнорированы, кроме запросов обновления подписки.

Пример работы ограничения количества запросов в интервал времени с зарегистрированного абонента на абонентском интерфейсе:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества запросов 50:
vesbc(config-esbc-user-interface-sip)# max in user rpp 50

#В интервал времени 240 секунд:
vesbc(config-esbc-user-interface-sip)# max in user rpp 50 240

#С блокировкой при достижении лимитов:
vesbc(config-esbc-user-interface-sip)# max in user rpp 50 240 block

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений количество запросов с зарегистрированного пользователя, проходящих через абонентский интерфейс USER_IFACE за 240 секунд, не может превышать 50, в случае превышения лимита абонент будет заблокирован. При получении новых запросов до истечения времени прощения ("Forgive time in minutes") оно будет установлено в прежнее значение.

Просмотр списка заблокированных абонентов при превышении лимитов max in user rpp:

```
vesbc# show esbc black-list sip-user

SIP user black-list:
--------------------------------------------------------------------------------
AOR                    Ban reason       User        Forgive      Time of
                                        error       time in      blocking
                                        count       minutes
--------------------   --------------   ---------   ----------   ----------
1142@192.168.113.177   TOOMANY USER     139         60           2025-11-07
                       REQUESTS                                  04:38:05

vesbc#
```

Пример работы ограничения на абонентском интерфейсе контактов на одном абоненте:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки абонентского интерфейса:
vesbc(config-esbc)# user-interface sip USER_IFACE
vesbc(config-esbc-user-interface)#

#Ограничение максимального количества контактов для одного AOR:
vesbc(config-esbc-user-interface-sip)# max in user contacts 2

#Применение и подтверждение изменений:
vesbc(config-esbc-user-interface)# do commit
vesbc(config-esbc-user-interface)# do confirm
```

После применения изменений, абонентам, регистрируемым через абонентский интерфейс USER_IFACE, будет возможно зарегистрировать только два разных контакта. При поступлении запроса REGISTER, содержащего дополнительный контакт, ESBC отклонит запрос кодом 403 Forbidden с "P-Eltex-Diagnostic: Too many registered contacts".

9.16.2 Контроль исходящего трафика

На ESBC реализован контроль исходящего трафика для:

- вызовов в секунду (max out cps);
- запросов в секунду (max out rps);
- контактов на направлении (max out contacts).

Описание всех команд для контроля исходящего трафика приведено в разделах для каждого объекта в CLI:

1. trunk;
2. trunk-group;
3. user-interface.

Реализация ограничения исходящего трафика поддержана на следующих объектах:

max out ...	Для всей системы (general)	Транков (trunk)	Транк-групп (trunk-group)	Абонентских интерфейсов (user-interface)
cps	✗	✓	✓	✓
rps	✗	✓	✓	✓
contacts	✓	✓	✓	✗

При использовании **max out contacts** будет ограничено возможное количество активных регистраций контактов на направлении.

> ⚠ Ограничение **max out contacts** является глобальным. Лимит ограничения из general действует на всю систему, а не используется в каждом объекте системы. Лимит этого ограничения, использующийся в general, нельзя превысить.

При одновременном использовании **max out cps** и **max out rps** в настройках транка, транковой группы или абонентского интерфейса, максимальным порогом для всех исходящих запросов будет являться значение, настроенное в **max out rps**. При этом ограничение для исходящих запросов INVITE можно уменьшить отдельно настройкой **max out cps**.

Пример:

1. В конфигурации настроено max out cps 10 и max out rps 5. В таком случае максимальное ограничение на вызовы (INVITE) будет 5.
2. В конфигурации настроено max out cps 5 и max out rps 10. В таком случае максимальное ограничение на вызовы будет 5, а на остальные запросы — 10.

> ⚠ При применении ограничений приоритет определяется уровнем: приоритет ограничений транк-групп выше, чем приоритет ограничений отдельных транков, входящих в эту транк-группу.

Пример приоритета ограничений транк-группы над ограничениями транков, входящих в транк-группу:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки транка TRUNK_1:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)#

#Ограничение максимальных исходящих CPS:
vesbc(config-esbc-trunk-sip)# max out cps 100
vesbc(config-esbc-trunk-sip)# exit

#Переход в настройки транка TRUNK_2:
vesbc(config-esbc)# trunk sip TRUNK_2
vesbc(config-esbc-trunk-sip)#

#Ограничение максимальных исходящих CPS:
vesbc(config-esbc-trunk-sip)# max out cps 100
vesbc(config-esbc-trunk-sip)# exit

#Переход в настройки транк-группы:
vesbc(config-esbc)# trunk-group TRUNKGROUP
vesbc(config-esbc-trunk-group)# trunk 0 TRUNK_1
vesbc(config-esbc-trunk-group)# trunk 1 TRUNK_2

#Ограничение максимальных исходящих CPS:
vesbc(config-esbc-trunk-group)# max out cps 50

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-group)# do commit
vesbc(config-esbc-trunk-group)# do confirm
```

Т. к. транки TRUNK_1 и TRUNK_2 входят в состав транк-группы TRUNKGROUP, то суммарное количество исходящих CPS с любого из этих транков не может превышать 50, так как приоритет ограничения у транк-группы выше.

Пример работы приоритета системных ограничений над транком:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Установка ограничения максимального количества контактов на направлении для всей системы – 50:
vesbc(config-esbc-general)# max out contacts 50
vesbc(config-esbc-general)# exit

#Переход в настройки транка TRUNK_1:
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)#

#Ограничение максимальных максимального количества контактов на транке – 100:
vesbc(config-esbc-trunk-sip)# max out contacts 100
vesbc(config-esbc-trunk-sip)# exit

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-group)# do commit
vesbc(config-esbc-trunk-group)# do confirm
```

В данном случае на системе будет не более 50 активных регистраций, так как у общесистемных ограничений приоритет выше, чем у транка.

Пример ограничения на транке исходящих вызовов в секунду:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки транка:
vesbc(config-esbc)# trunk sip TRUNK
vesbc(config-esbc-trunk-sip)#

#Ограничение максимального исходящего количества CPS:
vesbc(config-esbc-trunk-sip)# max out cps 10

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

После применения изменений количество исходящих вызовов в секунду с транка TRUNK не может превышать 10. При достижении лимита ESBC будет отвечать кодом 480 на каждый новый INVITE, который будет маршрутизирован на данный транк.

Пример ограничения на транке исходящих запросов в секунду:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки транка:
vesbc(config-esbc)# trunk sip TRUNK
vesbc(config-esbc-trunk-sip)#

#Ограничение максимального исходящего RPS:
vesbc(config-esbc-trunk-sip)# max out rps 250

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

После применения изменений количество исходящих запросов в секунду с транка TRUNK не может превышать 250. На все запросы (INVITE, REGISTER и другие), поступающие после превышения лимита, ESBC будет отвечать кодом 480.

Пример ограничения количества контактов на транке:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в настройки транка:
vesbc(config-esbc)# trunk sip TRUNK
vesbc(config-esbc-trunk-sip)#

#Ограничение максимального количества контактов:
vesbc(config-esbc-trunk-sip)# max out contacts 100

#Применение и подтверждение изменений:
vesbc(config-esbc-trunk-sip)# do commit
vesbc(config-esbc-trunk-sip)# do confirm
```

После применения изменений количество активных регистраций контактов на транке TRUNK не будет превышать 100. При достижении лимита ESBC будет маршрутизировать новые регистрации на следующее направление, если оно есть в конфигурации. Если в конфигурации нет подходящего направления, то новые регистрации будут отклоняться кодом 503.

Лицензионное ограничение обработки вызовов

Максимальное количество одновременных вызовов и максимальное количество вызовов в секунду ограничиваются лицензиями ESBC-LIMIT-MAX-CALLS и ESBC-LIMIT-MAX-CPS соответственно.
При этом в конфигурации можно задать ограничение, которое превышает лицензионное значение, но ESBC не будет обрабатывать больше, чем позволяет лицензия, пример:

```
#Просмотр активных лицензий:
vesbc# show licence
Feature                          Source     State         Value         Valid from  Expiries
-----------------------------   --------   -----------   -------------  ---------   --------
ESBC-LIMIT-MAX-CALLS               ELM      Active        5000            --         --
ESBC-LIMIT-MAX-CPS                 ELM      Active        100             --         --
ESBC-VIRTUAL-LIMIT-DEFAULT         ELM      Active        true            --         --
ESBC-VIRTUAL-LIMIT-NET             ELM      Active        10000000000     --         --
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Ограничение максимального CPS:
vesbc(config-esbc-general)# max cps
  COUNT  Possible max cps: 1-1000   #конфигурационное ограничение

vesbc(config-esbc-general)# max cps 1000
2025-04-22T09:10:17+00:00 %SYS-W-EVENT: WARNING!!! Configured max cps 1000 exceed licence limit
that is equal to 100   #предупреждение о том, что введённое значение превышает лицензионное

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
2025-04-22T08:44:46+00:00 snmpd restarted
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
2025-04-22T08:44:46+00:00 %CLI-I-CRIT: user admin from console  input: do commit
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.
vesbc(config-esbc-general)#
```

После применения изменений в конфигурации будет отображаться max cps 1000, но обрабатываться будет не более 100 вызовов в секунду.

9.17 Мониторинг

В ESBC доступен мониторинг следующих параметров:

- активные вызовы;
- расширенный мониторинг вызовов (call-flow);
- чёрный список;
- белый список;
- состояние транков;
- список зарегистрированных абонентов;
- статистика SIP;
- мониторинг параметров трафика (KPI).

9.17.1 Активные вызовы

Просмотр списка активных вызовов осуществляется командой show esbc active calls в CLI.

В выводе информации об активных сессиях присутствует:

- Total call sessions — общее количество активных сессий;
- Session id — id активной сессии;
- Duration (sec) — длительность активной сессии в секундах;
- CGPN unmodified — номер вызывающей стороны до модификаций;
- CDPN unmodified — номер вызываемой стороны до модификаций;
- Source — источник вызова;
- Destination — место назначения вызова;
- CGPN modified — номер вызывающей стороны после модификаций (если модификаций нет или они не применялись, то номер останется без изменений);
- CDPN modified — номер вызываемой стороны после модификаций (если модификаций нет или они не применялись, то номер останется без изменений).

Реализовано удаление активных сессий командой clear esbc active calls в CLI.

Поддерживается вывод до 50000 активных звонков.

9.17.2 Расширенный мониторинг вызовов (call-flow)

Реализован просмотр подробной информации sip call-flow сессий в CLI.

Для включения данного функционала, необходимо запустить сбор статистики командой esbc call-flow sip start в CLI.

При повторном включении ведения статистики уже собранные данные статистики очищаются.

Реализована возможность использовать фильтрацию при сборе статистики:

1. Фильтрация с соблюдением всех условий (esbc call-flow sip all);
2. Фильтрация с соблюдением любого условия (esbc call-flow sip any).

Доступные условия фильтрации (возможно использование более одного условия):

1. address — фильтрация по адресу;
2. cdpn — фильтрация по номеру вызываемого абонента;
3. cgpn — фильтрация по номеру вызывающего абонента;

4. contact — фильтрация по заголовку Contact;
5. transport — фильтрация по транспорту;
6. trunk — фильтрация по транку;
7. trunk-group — фильтрация по транк-группе;
8. user-agent — фильтрация по заголовку User-Agent;
9. user-interface — фильтрация по абонентскому интерфейсу.

Пример фильтрации с условиями:

```
#Сбор статистики с выполнением всех условий для вызываемого номера 114..., транспорта
 TRANSPORT_ABONENTS и транка TRUNK_SMG:
vesbc# esbc call-flow sip all cdpn 114.* transport TRANSPORT_ABONENTS trunk TRUNK_SMG
vesbc#
```

⚠ Если не использовать фильтрацию или условия в ней, то будет происходить сбор статистики всех sip call-flow сессий.

Вывод краткой информации по всем sip call-flow сессиям возможен с помощью команды show esbc call-flow sip list в CLI.

В выводе присутствует:

1. Общее количество call-flow сессий;
2. Session id — идентификатор сессии;
3. Start time — время начала установления соединения;
4. CGPN unmodified — немодифицированный номер вызывающего абонента;
5. CDPN unmodified — немодифицированный номер вызываемого абонента;
6. Source — источник (транк или абонентский интерфейс);
7. Destination — место назначения (транк или абонентский интерфейс);
8. CGPN modified — модифицированный номер вызывающего абонента;
9. CDPN modified — модифицированный номер вызываемого абонента.

Пример вывода статистики:

```
#Просмотр отфильтрованных call-flow сессий:
vesbc# show esbc call-flow sip list
 Total call-flow sessions:       2
ESBC active sessions:
-----------------------------------
Session id             Start time             CGPN         CDPN         Source                 Destination            CGPN         CDPN
                                              unmodified   unmodified                                                 modified     modified
--------------------   --------------------   ----------   ----------   --------------------   --------------------   ----------   ----------
50300000000000c        2025-10-08 09:50:25    1140         1142         ABONENTS (uiface)      TRUNK_SMG (trunk)      1140         1142
50300000000000d        2025-10-08 09:50:26    1140         1142         TRUNK_SMG (trunk)      ABONENTS (uiface)      1140         1142
vesbc#
```

Подробную информацию об отфильтрованной сессии можно посмотреть с помощью команды show esbc call-flow sip info <SESSION_ID> [detailed] в CLI.

При вводе *show esbc call-flow sip info <SESSION_ID>* будет доступна диаграмма отдельной сессии.

Пример вывода статистики, см. ниже.

```
#Просмотр диаграммы отфильтрованной call-flow сессий:
vesbc# show esbc call-flow sip info 50300000000000c
ESBC call flow 50300000000000c
Time              Leg A                                   Direction   Leg B
---------------   -------------------------------------   ---------   -------
2025-10-08        (UDP SDP) INVITE                           --->
09:50:25.976      sip:1142@192.168.113.177:5090 SIP/2.0
(+00:00:00.000)

2025-10-08        (UDP) SIP/2.0 100 Trying                   <---
09:50:25.976
(+00:00:00.000)

2025-10-08                                                   --->      (UDP SDP) INVITE
09:50:25.992
sip:1142@192.168.113.172:5070 SIP/2.0
(+00:00:00.016)

2025-10-08                                                   <---      (UDP) SIP/2.0 100
Trying
09:50:26.003
(+00:00:00.027)

2025-10-08                                                   <---      (UDP) SIP/2.0 180
Ringing
09:50:26.046
(+00:00:00.070)

2025-10-08        (UDP) SIP/2.0 180 Ringing                  <---
09:50:26.048
(+00:00:00.072)

2025-10-08                                                   <---      (UDP SDP) SIP/2.0 200
OK
09:50:27.544
(+00:00:01.568)

2025-10-08        (UDP SDP) SIP/2.0 200 OK                    <---
09:50:27.557
(+00:00:01.581)

2025-10-08        (UDP) ACK                                   --->
09:50:27.600      sip:1142@192.168.113.177:5090;transport=
(+00:00:01.624)   udp SIP/2.0

2025-10-08                                                    --->      (UDP) ACK
sip:1142@192.168.113.172:5070
09:50:27.603                                                            SIP/2.0
(+00:00:01.627)

2025-10-08                                                    <---      (UDP) BYE
09:50:44.673
sip:1140@192.168.113.177:5071;transport=
(+00:00:18.697)
udp;line=83b0f9ea4622375be3232f3c892b024
                                                                        0 SIP/2.0

2025-10-08        (UDP) BYE                                   <---
```

```
09:50:44.675      sip:1140@192.168.113.170;transport=udp
(+00:00:18.699)   SIP/2.0

2025-10-08        (UDP) SIP/2.0 200 Ok                        --->
09:50:44.700
(+00:00:18.724)

2025-10-08                                                    --->      (UDP) SIP/2.0 200 OK
09:50:44.703
(+00:00:18.727)
vesbc#
```

При вводе *show esbc call-flow sip info <SESSION_ID> detailed* будет доступна более подробная информация по выбранной сессии.

В зависимости от сценария вызова, в выводе статистики может присутствовать:

1. Call state — статус сессии;
2. Leg — плечо сессии;
3. Transport — используемый транспорт;
4. User interface — используемый абонентский интерфейс;
5. Remote Contact — display name и URI в заголовке Contact;
6. From — display name и URI в заголовке From;
7. To — display name и URI в заголовке To;
8. Remote User-Agent — полное содержимое заголовка User-Agent;
9. Call id — уникальный идентификатор плеча;
10. Local ip — локальный адрес;
11. Remote ip — адрес удаленной стороны;
12. Start time — время начала установления плеча сессии;
13. End time — время окончания плеча сессии.

Пример вывода статистики:

```
#Просмотр подробной информации отфильтрованной call-flow сессий:
vesbc# show esbc call-flow sip info 50300000000000c detailed
ESBC call flow detailed 50300000000000c
Call state: Destroyed
-----------------------------------
Leg: A
Transport: TRANSPORT_ABONENTS
User interface: ABONENTS
Remote Contact: "1140" <sip:1140@192.168.113.170;transport=udp>
From: "1140" <sip:1140@192.168.113.177>
To: <sip:1142@192.168.113.177>
Remote User-Agent: VP-17P/1.5.6-b46 sofia-sip/1.28
Call id: cb612700-9916-1200-4783-6813e209aef3
Local ip: 192.168.113.177:5090
Remote ip: 192.168.113.170:5060
Start time: 2025-10-08 09:50:25.976
End time: 2025-10-08 09:50:44.700
-----------------------------------
Leg: B
Transport: TRANSPORT_SMG
Trunk: TRUNK_SMG
Remote Contact: <sip:1142@192.168.113.172:5070>
From: "1140" <sip:1140@192.168.113.172>
To: <sip:1142@192.168.113.172>
Remote User-Agent: smg pa_sip 3.408.2.100
Call id: 5153401c612eebba1d5691886dbcbb92
Local ip: 192.168.113.177:5071
Remote ip: 192.168.113.172:5070
Start time: 2025-10-08 09:50:25.992
End time: 2025-10-08 09:50:44.703
vesbc#
```

Для сброса текущей статистики используется команда esbc call-flow sip clear в CLI.

Для отключения сбора статистики используется команда esbc call-flow sip stop в CLI. Статистика для новых сессий собираться не будет, но для активных будет актуализироваться до их завершения.

> ⚠ Сбор статистики прекращается, и уже собранные данные сбрасываются, если осталось менее 20% начального объема оперативной памяти (RAM).

9.17.3 Черный список

Реализован просмотр черного списка командой show esbc black-list (IP-адреса, AOR, User-Agent, SIP user) в CLI и на странице Мониторинг → Списки доступа → Чёрный список (IP-адреса) в WEB.

В выводе черного списка может присутствовать до 4 таблиц (по блокируемым объектам):

1. IP black-list:
   - IP address — заблокированный IP-адрес;
   - Ban reason — причина блокировки;
   - Address of Record (AOR) / User-Agent (UA);
   - Error count — количество ошибок AOR/UA;
   - Blocking timeout in minutes — оставшееся время блокировки в минутах;
   - Time of blocking — время блокировки.

2. AOR:
   - AOR;
   - Ban reason — причина блокировки;
   - AOR error count — количество ошибок AOR;
   - Forgive time in minutes — оставшееся время блокировки в минутах;
   - Time of blocking — время блокировки.
3. SIP user
   - AOR;
   - Ban reason — причина блокировки;
   - User error count — количество ошибок;
   - Forgive time in minutes — оставшееся время блокировки в минутах;
   - Time of blocking — время блокировки.
4. User-agent:
   - UA;
   - Ban reason — причина блокировки;
   - UA error count — количество ошибок UA;
   - Forgive time in minutes — оставшееся время блокировки в минутах;
   - Time of blocking — время блокировки.

Причины блокировок описаны в разделе Общий принцип работы модуля fail2ban.

Реализована очистка черного списка командой clear esbc black-list в CLI или кнопкой «Удалить» в WEB.

9.17.4 Белый список

Поддержана возможность просматривать белый список IP-адресов, AOR и User-Agent командой show esbc white-list в CLI и, для IP-адресов, на странице Мониторинг → Списки доступа → Белый список в WEB.

Реализовано добавление в белый список динамических адресов и доменов.

В белом списке также присутствуют два параметра:

- Is dynamic — объекты, которые не присутствуют в конфигурации ESBC, но были подтверждены иным способом (например, адрес абонента при регистрации заносится в белый список);
- Is configured — объекты, которые присутствуют в конфигурации ESBC.

9.17.5 Состояние транков

Реализован просмотр состояния транков командой show esbc trunks в CLI или на странице Мониторинг → Телефония → Транки в WEB.

В таблице выводится:

- Trunk — имя транка;
- Trunk type — тип транка;
- Status — статус транка (принимает значения Uncontrolled, Available или Not available, в зависимости от настройки SIP-профиля и реального состояния транка);
- Last change time — время изменения статуса транка;
- Registration type — режим регистрации транка (принимает значения None или Client);
- Registration count (active/all) — краткое отображение активных регистраций (active) и всех номеров (all), которые присутствуют в привязанном к транку профилю учётных данных. Отображается только при включении клиентской регистрации транка.

9.17.6 Список зарегистрированных абонентов

Реализован просмотр списка зарегистрированных абонентов командой show esbc users в CLI или на странице Мониторинг → Телефония → Абоненты в WEB.

Выводится общее количество AOR и Contact, а также базовый вывод информации о абонентах.

> Количество AOR и Contact может не совпадать, если абоненты имеют несколько Contact.

Также реализован просмотр подробной информации по конкретному абоненту, используя дополнительный параметр после основной команды (sip <AOR> detailed).

Выводится подробная информация по определенному AOR:

- User AOR;
- User type — тип абонента;
- IN User contact — входящий заголовок Contact;
- IP address of user — IP-адрес абонента;
- User interface name — user-interface, через который зарегистрировался абонент;
- Expires — время перерегистрации;
- Registration expires in — время до перерегистрации;
- Trunk name — trunk, на который отправлен запрос Register от абонента;
- IP address of registrar — IP-адрес сервера регистрации;
- OUT Trunk contact — исходящий заголовок Contact.

> Реализовано удаление активных регистраций командой clear esbc registration в CLI.

> Регулярное выражение для clear esbc registration pattern-aor применяется с логикой полного совпадения (exact match). Например, необходимо написать шаблон, который будет соответствовать номеру "24005". В данном случае шаблон "^24.*" будет корректным, а шаблон "24" — нет.

9.17.7 Статистика SIP

Реализован просмотр статистики для всей системы, всех транков, всех абонентских интерфейсов или по конкретному объекту командой show esbc statistics (вызовы, регистрации, подписки, RPS) в CLI или на странице Мониторинг → Телефония → Статистика в WEB.

> Ведение статистики по умолчанию включено.

При вызове команды для просмотра статистики отображаются таблицы с метриками, описание каждой метрики можно найти в разделе show esbc statistics справочника команд CLI.

> В CLI отображаются счётчики за последнюю секунду. В WEB реализован просмотр истории за последнюю минуту, час, 3 дня.

> Для отключения ведения статистики необходимо в меню general отключить ее командой statistics disable.

Пример:

Из TRUNK_IN в TRUNK_OUT через ESBC поступает 2 вызова каждую секунду длительностью 25 секунд.

```
#Просмотр полной статистики при активных вызовах:
vesbc# show esbc statistics all

ESBC global call counters:
-------------------------------------------------------------------------
Counter Name                Incoming                Outgoing
-------------------------   --------------------    --------------------
CALLS PER SECOND            2                       2
CALL LEGS                   52                      52
REQUESTS IN CALL            6                       6
RESPONSES IN CALL           8                       8
ANSWERED CALLS              2                       2
CALLS TO WRONG NUMBER       0                       0
BUSY CALLS                  0                       0
NO ANSWERED CALLS           0                       0
FORBIDDEN CALLS             0                       0
UNAUTHORIZED CALLS          0                       0
3XX CODES                   0                       0
4XX CODES                   0                       0
5XX CODES                   0                       0
6XX CODES                   0                       0
CALLS DURATION              743                     743
CALLS AVERAGE DURATION      13                      13
CALLS ACTIVE                50                      50

ESBC global register counters:
-------------------------------------------------------------------------
Counter Name                Incoming                Outgoing
-------------------------   --------------------    --------------------
REGISTERS PER SECOND        0                       0
REGISTER TRANSACTIONS       0                       0
REQUESTS IN REGISTER        0                       0
RESPONSES                   0                       0
SUCCESS REGISTERS           0                       0
REQUEST TIMEOUT             0                       0
FORBIDDEN REGISTERS         0                       0
UNAUTHORIZED REGISTERS      0                       0
INTERVAL TOO BRIEF          0                       0
3XX CODES                   0                       0
4XX CODES                   0                       0
5XX CODES                   0                       0
6XX CODES                   0                       0
REGISTERED USERS            0                       0
REGISTERED CONTACTS         0                       0

ESBC global subscribe counters:
-------------------------------------------------------------------------
Counter Name                Incoming                Outgoing
-------------------------   --------------------    --------------------
SUBSCRIBES PER SECOND       0                       0
ACTIVE SUBSCRIBES           0                       0
REQUESTS IN SUBSCRIBE       0                       0
RESPONSES IN SUBSCRIBE      0                       0
SUCCESS SUBSCRIBES          0                       0
REQUEST TIMEOUT             0                       0
FORBIDDEN SUBSCRIBES        0                       0
UNAUTHORIZED SUBSCRIBES     0                       0
INTERVAL TOO BRIEF          0                       0
```

```
3XX CODES                   0                       0
4XX CODES                   0                       0
5XX CODES                   0                       0
6XX CODES                   0                       0

ESBC global rps counters:
-------------------------------------------------------------------------
Counter Name                Incoming                Outgoing
-------------------------   --------------------    --------------------
REQUESTS PER SECOND         6                       6
INVITE PER SECOND           2                       2
ACK PER SECOND              2                       2
BYE PER SECOND              2                       2
CANCEL PER SECOND           0                       0
REFER PER SECOND            0                       0
PRACK PER SECOND            0                       0
SUBSCRIBE PER SECOND        0                       0
NOTIFY PER SECOND           0                       0
UPDATE PER SECOND           0                       0
OPTIONS PER SECOND          0                       0
INFO PER SECOND             0                       0
REGISTER PER SECOND         0                       0
MESSAGE PER SECOND          0                       0
```

9.17.8 Мониторинг параметров трафика (KPI)

Реализован просмотр истории статистики параметров трафика для всей системы или по конкретному объекту (транспорту/транку/абонентскому интерфейсу) командой show esbc history (активные вызовы, средняя длительность вызова (ACD), входящие попытки вызова, исходящие попытки вызова, зарегистрированные контакты, зарегистрированные пользователи) в CLI или, для активных вызовов и попыток вызова, на странице Мониторинг → Телефония → Статистика→ Вызовы в WEB. Описание каждой метрики можно найти в разделе show esbc history справочника команд CLI.

Данные выводятся в виде гистограммы. При вызове команды отображается 3 графика – за последнюю минуту, час и 3 дня. Опция interval позволяет отобразить график для конкретного интервала времени.

> ⚠ График отображает значения метрики от 10 и выше. Вертикальная ось графика автоматически подстраивается под диапазон отображаемых данных.

Пример 1.

Схема:

Из транка UAC в транк UAS через ESBC поступает 10 вызовов каждую секунду длительностью 25 секунд.

```
#Просмотр истории максимальной статистики активных вызовов:
vesbc# show esbc history active-calls max
CALLS ACTIVE
Active calls, quantity

300 |
270 |
240 |###########################
210 |##############################
180 |#################################
150 |####################################
120 |#######################################
 90 |##########################################
 60 |#############################################
 30 |################################################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                        Time, seconds

CALLS ACTIVE
Active calls, quantity

100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |
 10 |
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                        Time, minutes

CALLS ACTIVE
Active calls, quantity

100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |
 10 |
  0 |....|....|....|....|....|....|....|....|....|....|....|....|....|....|..
    0    5   10   15   20   25   30   35   40   45   50   55   60   65   70
                                                                   Time, hours

#Просмотр истории максимальной статистики входящих попыток вызова за последнюю минуту на транке
 UAC:
vesbc# show esbc history call-attempts-incoming max interval seconds trunk sip
UAC
INCOMING CALL ATTEMPS
```

```
Call attemps, quantity

100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |
 10 |##############################################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                        Time, seconds

#Просмотр истории максимальной статистики исходящих попыток вызова за последнюю минуту на
транке UAS:
vesbc# show esbc history call-attempts-outgoing max interval seconds trunk sip
UAS
OUTGOING CALL ATTEMPS
Call attemps, quantity

100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |
 10 |##############################################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                        Time, seconds

#Просмотр истории максимальной статистики средней длительности вызова (ACD) за последний час:
vesbc# show esbc history average-call-duration max interval minutes
AVERAGE CALLS AVERAGE DURATION
Avg duration, sec

100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |#####
 10 |######
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                        Time, minutes
```

Пример 2.

Через абонентский интерфейс USERS_1 каждую секунду регистрируются 10 абонентов. Всего зарегистрировалось 500 абонентов.

```
#Просмотр истории статистики зарегистрированных контактов за последнюю минуту на абонентском
интерфейсе:
vesbc# show esbc history contacts interval seconds user-interface sip USERS_1
REGISTERED CONTACTS
Contacts, quantity

500 |####################
450 |#######################
400 |##############################
350 |#################################
300 |#######################################
250 |##########################################
200 |###############################################
150 |#######################################################
100 |##########################################################
 50 |############################################################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                       Time, seconds

#Просмотр истории статистики зарегистрированных пользователей:
vesbc# show esbc history registrations
REGISTERED USERS
Users, quantity

500 |############################################################
450 |############################################################
400 |############################################################
350 |############################################################
300 |############################################################
250 |############################################################
200 |############################################################
150 |############################################################
100 |############################################################
 50 |############################################################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                       Time, seconds

REGISTERED USERS
Users, quantity

500 |####################
450 |####################
400 |####################
350 |#####################
300 |#####################
250 |#######################
200 |#######################
150 |#######################
100 |#######################
 50 |#######################
  0 |....|....|....|....|....|....|....|....|....|....|....|....|
    0    5   10   15   20   25   30   35   40   45   50   55   60
                                                       Time, minutes

REGISTERED USERS
Users, quantity
```

```
100 |
 90 |
 80 |
 70 |
 60 |
 50 |
 40 |
 30 |
 20 |
 10 |
  0 |....|....|....|....|....|....|....|....|....|....|....|....|....|....|..
    0    5   10   15   20   25   30   35   40   45   50   55   60   65   70
                                                                   Time, hours
```

9.17.9 Утилита Sngrep

Утилита **sngrep** — инструмент интерактивного мониторинга SIP-сигнализации и медиапотоков RTP в реальном времени.

Возможности утилиты:

- отображение SIP-сообщений (INVITE, ACK, BYE и др.) в реальном времени с кодировкой цветом по методам и кодам ответов;
- отслеживание потоков вызовов (call-flow) между точками; в режиме *call flow* отображается последовательность SIP-транзакций по Call-ID.

Варианты запуска утилиты **sngrep:**

- команда **sngrep** – отображение всех SIP-сообщений (INVITE, ACK, BYE, REGISTER, SUBSCRIBE и др.);
- команда **sngrep calls** – отображение SIP-сообщений только в рамках вызова (INVITE, ACK, BYE и др.). В данном режиме не отображаются SIP-сообщения вне диалога.

Для выхода из утилиты используется комбинация клавиш **<Ctrl> + <C>** или клавиша **<Esc>**.

⚠ Не рекомендуется запускать утилиту **sngrep** в сессии **serial-подключения (последовательный порт)**. Это может приводить к **искажению графического отображения.**

⚠ Утилита Sngrep позволяет оперативно получить информацию о прохождении SIP-сигнализации. Для получения полной диагностической информации, анализа RTP-трафика, рекомендуется использовать запись дампа сетевого трафика в файл командой monitor.

Пример

Посмотреть прохождение вызова через ESBC.

Запуск утилиты:

```
vesbc# sngrep
```

Выбор двух call-leg вызова для просмотра детальной информации (для выбора используется клавиша **<Space>**):

```
                                   sngrep - SIP messages flow viewer
Current Mode: Online [any]                Dialogs: 2
Match Expression:                         BPF Filter:
Display Filter:
    ^Idx Method     SIP From                  SIP To                    Msgs  Source                  Destination             Call State
[*] 1    INVITE     24001@192.168.113.200:507 23002@192.168.113.207:507 7     192.168.113.200:5070    192.168.113.207:5070    COMPLETED
[*] 2    INVITE     24001@192.168.113.207     23002@192.168.113.200     7     192.168.113.207:5080    192.168.113.200:5080    COMPLETED
```

Просмотр детальной информации:

```
                                   Call flow for 2 dialogs (Color by Request/Response)
          192.168.113.200:5070          192.168.113.207:5070          192.168.113.207:5080          192.168.113.200:508   INVITE sip:23002@192.168.113.207:5070 SIP/2.0
                                                                                                                          Via: SIP/2.0/UDP 192.168.113.200:5070;rport;branch=z9hG4bK-1198652-1-1
                                                                                                                          From: sipp <sip:24001@192.168.113.200:5070>;tag=1
11:49:47.068040                INVITE (SDP)                                                                               To: sut <sip:23002@192.168.113.207:5070>
       +0.024798               ------------------------------>                                                          Call-ID: 1-1198652@192.168.113.200
11:49:47.092838                100 Trying                                                                                 Cseq: 1 INVITE
       +0.022866               <------------------------------                                                          Contact: <sip:24001@192.168.113.200:5070>
11:49:47.115704                                                                                                           Max-Forwards: 70
       +0.002416                                                             INVITE (SDP)                               Subject: Performance Test
11:49:47.118120                                                              ------------------------------>            Allow: INVITE, ACK, BYE, CANCEL, PRACK, REGISTER, INFO, REFER, NOTIFY, OPTIONS, S
       +0.052144                                                             100 Trying                                 UBSCRIBE, MESSAGE, UPDATE, PUBLISH
11:49:47.170264                                                              <------------------------------            Supported: timer, replaces, path
       +0.004391                                                             180 Ringing                                Content-Type: application/sdp
11:49:47.174655                180 Ringing                                   <------------------------------            Content-Length:   325
       +0.999750               <------------------------------
11:49:48.174405                                                              200 OK (SDP)                               v=0
       +0.016068                                                             <------------------------------            o=user1 77755765 7773687637 IN IP4 192.168.113.200
11:49:48.190473                200 OK (SDP)                                                                               s=-
       +0.002282               <------------------------------                                                          t=0 0
11:49:48.192755                ACK                                                                                        c=IN IP4 192.168.113.200
       +0.006367               ------------------------------>                                                          m=audio 10000 RTP/AVP 8 0 3 97 9 102 101
11:49:48.199122                                                                                                           a=rtpmap:8 PCMA/8000
      +10.005682                                                             ACK                                        a=rtpmap:0 PCMU/8000
11:49:58.204804                BYE                                           ------------------------------>            a=rtpmap:3 GSM/8000
       +0.008707               ------------------------------>                                                          a=rtpmap:97 G726-32/8000
11:49:58.213511                                                                                                           a=rtpmap:9 G722/8000
       +0.002279                                                             BYE                                        a=rtpmap:102 CN/16000
11:49:58.215790                                                              ------------------------------>            a=rtpmap:101 telephone-event/8000
       +0.006353                                                             200 OK                                     a=fmtp:101 0-16
11:49:58.222143                200 OK                                        <------------------------------
                               <------------------------------

Esc Calls List   Enter Raw   Space Compare   F1 Help   F2 SDP   F3 RTP   F4 Extended   s Compressed   F6 Raw   c Colour by   9 Increase Raw
```

9.18 Аварии

Включение журналирования аварий происходит с помощью команды alarm enable journal в CLI.

Данная команда без указания параметров включает весь набор аварий:

- cdr-send-error — ошибки отправки CDR;
- cdr-write-error — ошибки записи CDR;
- general-max-calls-limit — превышение общего лимита вызовов;
- general-max-cps-limit — превышение общего лимита cps;
- general-max-rps-limit — превышение общего лимита rps;
- media-resources — отсутствие свободных медиаресурсов (портов для RTP);
- module-connection — перезапуск модулей ESBC;
- trunk-group-max-calls-limit — превышение лимита вызовов на транк-группе;
- trunk-group-max-cps-limit — превышение лимита cps на транк-группе;
- trunk-group-max-rps-limit — превышение общего лимита rps на транк-группе;
- trunk-max-calls-limit — превышение лимита вызовов на транке;
- trunk-max-cps-limit — превышение лимита cps на транке;
- trunk-max-rps-limit — превышение общего лимита rps на транке;
- trunk-unavailable — недоступность транка;
- user-interface-max-calls-limit — превышение лимита вызовов на user-interface;
- user-interface-max-cps-limit — превышение лимита cps на user-interface;
- user-interface-max-rps-limit — превышение общего лимита rps на user-interface.

История аварийных событий выводится командой show alarms brief в CLI. Данная команда выводит историю аварий, включая уже нормализированные аварии.

Для отображения только активных аварий используется команда show alarms brief active.

Пример:

```
#Просмотр истории аварийных событий:
vesbc# show alarms brief

   History Alarms
   --------------
Severity   Group   Set time             Clear time             Description
--------   ------  -------------------  --------------------   --------------------------------
major      esbc    2026-03-20 09:55:32  2026-03-20 10:01:32    Trunk(TRUNK_SSW_1) is unavailable
major      esbc    2026-03-20 15:51:17  -                      Trunk(TRUNK_SSW_2) is unavailable

#Просмотр активных аварийных событий:
vesbc# show alarms brief active

   History Alarms
   --------------
Severity   Group    Set time               Clear time           Description
--------   -------  --------------------   -------------------  --------------------------------
major      esbc     2026-03-20 15:51:17    -                    Trunk(TRUNK_SSW_2) is unavailable
```

Для удаления истории аварийных событий используется команда clear alarms в CLI.

> ⚠ Аварии сохраняются локально. Для отправки уведомлений об авариях на удаленный хост необходимо настроить отправку аварийных SNMP-трапов.

Текст аварий и причины их нормализации представлены в таблице ниже.

Авария	Текст аварии	Причины нормализации
cdr-send-error	CDR alarm: failed to send to <main | reserve> ftp server	успешная отправка CDR на FTP-сервер
cdr-write-error	CDR alarm: failed to write	успешная запись CDR
general-max-calls-limit	Host <host_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита
general-max-cps-limit	Host <host_name> max cps limit reached	через 10 секунд после последней аварии
general-max-rps-limit	Host <host_name> max rps limit reached	через 10 секунд после последней аварии
media-resources	Session<session_id>: <Trunk/User interface><trunk_name/ui_name> media resources out	через 15 секунд после последней аварии ИЛИ при освобождении портов для RTP
module-connection	Module <module_type> host <host_id> is down	при успешном добавлении модуля в диспетчер ESBC
trunk-group-max-calls-limit	Trunk-Group <trunk_group_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита

Авария	Текст аварии	Причины нормализации
trunk-group-max-cps-limit	Trunk-Group <trunk_group_name> max cps limit reached	через 10 секунд после последней аварии
trunk-group-max-rps-limit	Trunk-Group <trunk_group_name> max rps limit reached	через 10 секунд после последней аварии
trunk-max-calls-limit	Trunk <trunk_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита
trunk-max-cps-limit	Trunk <trunk_name> max cps limit reached	через 10 секунд после последней аварии
trunk-max-rps-limit	Trunk <trunk_name> max rps limit reached	через 10 секунд после последней аварии
trunk-unavailable	Trunk <trunk_name> is unavailable	при обновлении статуса транка на "Available"
user-interface-max-calls-limit	User interface <ui_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита
user-interface-max-cps-limit	User interface <ui_name> max cps limit reached	через 10 секунд после последней аварии
user-interface-max-rps-limit	User interface <ui_name> max rps limit reached	через 10 секунд после последней аварии

9.18.1 Отправка аварийных SNMP-трапов

Отправка уведомлений об аварийных событиях на удаленный хост с помощью SNMP-трапов настраивается с помощью команды snmp-server enable traps esbc в CLI.

Данная команда без указания параметров включает весь набор SNMP-трапов:

- cdr-send-error — ошибки отправки CDR;
- cdr-write-error — ошибки записи CDR;
- general-max-calls-limit — превышение общего лимита вызовов;
- general-max-cps-limit — превышение общего лимита cps;
- general-max-rps-limit — превышение общего лимита rps;
- media-resources — отсутствие свободных медиаресурсов (портов для RTP);
- module-connection — перезапуск модулей ESBC;
- trunk-group-max-calls-limit — превышение лимита вызовов на транк-группе;
- trunk-group-max-cps-limit — превышение лимита cps на транк-группе;
- trunk-group-max-rps-limit — превышение общего лимита rps на транк-группе;
- trunk-max-calls-limit — превышение лимита вызовов на транке;
- trunk-max-cps-limit — превышение лимита cps на транке;
- trunk-max-rps-limit — превышение общего лимита rps на транке;
- trunk-unavailable — недоступность транка;
- user-interface-max-calls-limit — превышение лимита вызовов на user-interface;
- user-interface-max-cps-limit — превышение лимита cps на user-interface;

- user-interface-max-rps-limit — превышение общего лимита rps на user-interface;
- voip-block-aor — блокировки по AOR;
- voip-block-ip — блокировки по IP-адресу;
- voip-block-user-agent — блокировки по User-Agent;
- voip-block-sip-user — блокировки по SIP-User.

⚠ При включении отправки SNMP-трапов командой *snmp-server enable traps esbc*, трапы будут отправляться вне зависимости от включения журналирования аварий локально.

Текст SNMP-трапов и причины их нормализации представлены в таблице ниже.

SNMP-трап	Текст SNMP-трапа	Причины нормализации
cdr-send-error	CDR alarm: failed to send to <main | reserve> ftp server	успешная отправка CDR на FTP-сервер
cdr-write-error	CDR alarm: failed to write	успешная запись CDR
general-max-calls-limit	Host <host_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80% от лимита
general-max-cps-limit	Host <host_name> max cps limit reached	через 10 секунд после последней аварии
general-max-rps-limit	Host <host_name> max rps limit reached	через 10 секунд после последней аварии
media-resources	Session<session_id>: <Trunk/User interface><trunk_name/ui_name> media resources out	через 15 секунд после последней аварии ИЛИ при освобождении портов для RTP
module-connection	Module <module_type> host <host_id> is down	при успешном добавлении модуля в диспетчер ESBC
trunk-group-max-calls-limit	Trunk-Group <trunk_group_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита
trunk-group-max-cps-limit	Trunk-Group <trunk_group_name> max cps limit reached	через 10 секунд после последней аварии
trunk-group-max-rps-limit	Trunk-Group <trunk_group_name> max rps limit reached	через 10 секунд после последней аварии
trunk-max-calls-limit	Trunk <trunk_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80 % от лимита
trunk-max-cps-limit	Trunk <trunk_name> max cps limit reached	через 10 секунд после последней аварии

SNMP-трап	Текст SNMP-трапа	Причины нормализации
trunk-max-rps-limit	Trunk <trunk_name> max rps limit reached	через 10 секунд после последней аварии
trunk-unavailable	Trunk <trunk_name> is unavailable	при обновлении статуса транка на "Available"
user-interface-max-calls-limit	User interface <ui_name> max calls limit reached	через 10 секунд после последней аварии ИЛИ при снижении счетчика на 80% от лимита
user-interface-max-cps-limit	User interface <ui_name> max rps limit reached	через 10 секунд после последней аварии
user-interface-max-rps-limit	User interface <ui_name> max cps limit reached	через 10 секунд после последней аварии
voip-block-aor	AOR <aor> has been banned	не нормализуются
voip-block-ip	IP address <ip> has been banned	не нормализуются
voip-block-user-agent	User-Agent <user-agent> has been banned	не нормализуются
voip-block-sip-user	SIP user <aor> has been banned	не нормализуются

В отличие от аварий, SNMP-трапы передают уведомления о заблокированных объектах:
- voip-block-aor;
- voip-block-ip;
- voip-block-user-agent;
- voip-block-sip-user.

Для просмотра заблокированных объектов в CLI используется команда show esbc black-list.

Конфигурирование удаленного хоста для отправки SNMP-уведомлений выполняется с помощью команды snmp-server host в CLI.

9.19 Настройка CDR

CDR (Call Detail Record, рус. «запись сведений о звонках») — это запись, содержащая подробную информацию о совершённых вызовах.

В файл CDR записываются следующие данные:

- Заголовок файла (опционален) (<hostname> CDR. File started at "YYYYMMDDhhmmss", где "YYYYMMDDhhmmss" – время начала сохранения записей в файл);
- Отличительный признак (опционален);
- Время поступления вызова;
- Время ответа на вызов;
- Входящий номер вызывающего абонента;
- Исходящий номер вызывающего абонента;
- Входящий номер вызываемого абонента;
- Исходящий номер вызываемого абонента;
- Имя trunk/user-interface вызывающего абонента;
- Имя trunk/user-interface вызываемого абонента;
- Длительность вызова;
- Причина разъединения (согласно ITU-T Q.850);
- Индикатор успешного вызова (1 — успешный, 0 — неуспешный);
- Сторона-инициатор разъединения (1 — вызывающая сторона, 2 — вызываемая сторона, 3 — ESBC);
- Call-ID входящего вызова;
- Call-ID исходящего вызова;
- Номер вызываемого абонента при переадресации;
- IP-адрес шлюза вызывающего абонента;
- IP-адрес шлюза вызываемого абонента;
- Список IP-адресов из заголовка Record-Route при установлении соединения в направлении от вызывающего абонента;
- Список IP-адресов из заголовка Via при установлении соединения в направлении от вызывающего абонента;
- IP-адрес из заголовка Contact вызывающего абонента;
- IP-адрес из заголовка Contact вызываемого абонента.

Значения параметров в файле CDR записываются в указанном выше порядке и разделяются символом ";".

> Запись "Номер вызываемого абонента при переадресации" создается только при включенной локальной обработке 3xx ответа в настройках sip profile.

Хранение записей CDR осуществляется в локальном хранилище ESBC или на внешнем USB-накопителе.

Отправка на внешний сервер осуществляется по протоколу FTP, SFTP и SCP. Поддерживается отправка на два FTP-сервера, два SFTP-сервера и два SCP-сервера одновременно.

Дополнительно поддерживается сохранение и отправка CDR в SYSLOG. Для отправки в SYSLOG требуется дополнительная конфигурация syslog.

Описание всех команд для настройки CDR приведено в разделе Настройки CDR.

Пример настройки записи CDR с опциональными полями, локальным хранением и отправкой на сервер FTP с резервированием в случае неудачной отправки приведен ниже.

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# cdr
vesbc(config-esbc-cdr)# enable

#Добавление заголовка в CDR-запись:
vesbc(config-esbc-cdr)# add-header

#Запись неудачных вызовов:
vesbc(config-esbc-cdr)# collect unsuccess

#Запись пустых CDR:
vesbc(config-esbc-cdr)# collect empty-files

#Режим создания записей:
vesbc(config-esbc-cdr)# create-mode periodically
vesbc(config-esbc-cdr)# per days 1
vesbc(config-esbc-cdr)# period hours 12
vesbc(config-esbc-cdr)# per minutes 30

#Включение отправки логов:
vesbc(config-esbc-cdr)# syslog enable

#Добавление отличительного признака:
vesbc(config-esbc-cdr)# signature otlichitelnyi_priznak

#Настройка локального хранения:
vesbc(config-esbc-cdr)# local
vesbc(config-esbc-cdr-local)# create-directories by-date
vesbc(config-esbc-cdr-local)# keep days 30
vesbc(config-esbc-cdr-local)# keep hours 12
vesbc(config-esbc-cdr-local)# keep minutes 30
vesbc(config-esbc-cdr-local)# path flash:cdr/cdr_record
vesbc(config-esbc-cdr-local)# save
vesbc(config-esbc-cdr-local)# exit

#Настройка основного FTP-сервера:
vesbc(config-esbc-cdr)# ftp
vesbc(config-esbc-cdr-ftp)# login main_ftp_server
vesbc(config-esbc-cdr-ftp)# password password_m_ftp
vesbc(config-esbc-cdr-ftp)# path /main_ftp/cdr_record
vesbc(config-esbc-cdr-ftp)# remote address 192.168.23.100
vesbc(config-esbc-cdr-ftp)# save
vesbc(config-esbc-cdr-ftp)# exit

#Настройка резервного FTP-сервера:
vesbc(config-esbc-cdr)# reserved-ftp
vesbc(config-esbc-cdr-res-ftp)# as-reserved
vesbc(config-esbc-cdr-res-ftp)# login reserve_ftp_server
vesbc(config-esbc-cdr-res-ftp)# password password_r_ftp
vesbc(config-esbc-cdr-res-ftp)# path /reserve_ftp/cdr_record
vesbc(config-esbc-cdr-res-ftp)# remote address 192.168.23.200
vesbc(config-esbc-cdr-res-ftp)# save

#Применение и подтверждение изменений:
vesbc(config-esbc-cdr-res-ftp)# do commit
```

```
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc-cdr-res-ftp)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Если отправка записи CDR на основной FTP-сервер (192.168.23.100) по какой-либо причине не произойдет, то она попытается отправиться на резервный FTP-сервер (192.168.23.200), в случае неудачи и на резервном, запись сохранится только локально.

Пример записи файла CDR

На ESBC настроена следующая конфигурация CDR:

```
cdr
  enable
  add-header
  signature test_signature
  period minutes 1
  local
    save
    path flash:cdr/
    create-directories by-date
    keep days 1
  exit
exit
```

После совершения успешного вызова длительностью 5 секунд, с номера 241 на номер 231 будет сформирован файл CDR (через 1 минуту) вида:

```
vesbc CDR. File started at "20260320050817"
test_signature
2026-03-20 05:07:45;2026-03-20 05:07:53;241;241;231;231;UAC;UAS;000005;016;1;1;974433e3-9ebd-12
3f-7cb4-ecb1e029e6ba;93a6cc0da4e18d8ad308b277e3434d53;;192.168.1.6;192.168.1.2;;192.168.1.6;192
.168.1.6;192.168.12.2;
```

9.20 Работа с логами

Логирование ESBC осуществляется с помощью syslog. Более подробно настройки syslog описаны в разделе Управление SYSLOG справочника команд CLI.

По умолчанию логирование модулей ESBC выключено.

> Включение логирования всех модулей при большой вызывной нагрузке может повлиять на производительность системы. Наибольшее влияние на производительность оказывает вывод логов в консоль (syslog console).

Для получения наиболее подробной информации, при диагностике неисправностей, рекомендуется использовать уровень логирования debug.

Пример настройки логирования на внешний syslog-сервер с уровнем debug:

```
vesbc# configure
vesbc(config)# syslog host SYSLOG_SERVER
vesbc(config-syslog-host)# remote-address 192.168.1.1
vesbc(config-syslog-host)# severity debug
vesbc(config-syslog-host)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-syslog-host)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

Модули, входящие в состав ESBC

Название	Описание	Назначение
esbc_core	модуль основной логики	обработка вызовов, отвечает за маршрутизацию вызовов, обеспечивает взаимодействие остальных модулей
esbc_sip_balancer	модуль управления подсистемой SIP	получение сообщений SIP (на открытый сокет) и передача их в модуль **esbc_sip_worker**
esbc_sip_worker	модуль расширения подсистемы SIP	адаптер протокола SIP, обрабатывает сообщения и передает данные модулю **esbc_core**
esbc_media_balancer	модуль управления подсистемой media	управление ресурсами в подсистеме media, выделяет RTP-порты и передает их в модуль **esbc_media_worker**
esbc_media_worker	модуль расширения подсистемы media	обработка медиапотоков (RTP)
esbc_config_manager	адаптер базы данных конфигурации	хранение конфигурации системы
esbc_access_mediator	модуль внешнего доступа	обработка внешних взаимодействий с системой CLI
esbc_ipc	брокер сообщений	обеспечение связи всех модулей в системе
esbc_dispatcher	модуль контроля состояния модулей	контроль модулей, индикация об изменении состояний модулей
esbc_sm	модуль управления абонентскими записями	добавление/удаление записей о регистрации абонентов, добавление/удаление/изменение контактов регистрации, хранение и восстановление записей из базы, предоставление информации о записях и контактах абонентов другим модулям системы

esbc_voip_guard	модуль fail2ban	отслеживает попытки обращения к сервису телефонии, при обнаружении постоянно повторяющихся неудачных попыток обращения с одного и того же IP-адреса или хоста, модуль блокирует попытки с этого IP-адреса/хоста
esbc_sysio	модуль взаимодействия с ОС	служит прослойкой между ESBC и ОС, на которой он разворачивается, предоставляет единый интерфейс взаимодействия с системой и реализует мониторинг различных системных событий
esbc_mon	модуль мониторинга	обеспечение функции мониторинга и сбора статистики
esbc_aaa	модуль aaa	аутентификация, хранение информации о вызовах
stun_server	модуль stun-server	обработка запросов STUN в режиме сервера

Включение логирования модулей ESBC производится в разделе debug:

```
vesbc#

#Переход в раздел debug:
vesbc# debug
vesbc(debug)#

#Включение логирования модуля esbc_dispatcher:
vesbc(debug)# debug esbc disp

#Включение логирования модуля esbc_config_manager:
vesbc(debug)# debug esbc cfgmgr

#Включение логирования модуля esbc_access_mediator:
vesbc(debug)# debug esbc accmed

#Включение логирования модуля esbc_mon:
vesbc(debug)# debug esbc mon

#Включение логирования модуля esbc_aaa:
vesbc(debug)# debug esbc aaa

#Включение логирования модуля esbc_core:
vesbc(debug)# debug esbc core

#Включение логирования модуля esbc_sip_balancer:
vesbc(debug)# debug esbc sipbl

#Включение логирования модуля esbc_sip_worker:
vesbc(debug)# debug esbc sipwrk

#Включение логирования модуля esbc_media_balancer:
vesbc(debug)# debug esbc mediabl

#Включение логирования модуля esbc_media_worker:
vesbc(debug)# debug esbc mediawrk

#Включение логирования модуля esbc_sysio:
vesbc(debug)# debug esbc sysio

#Включение логирования модуля esbc_sm:
vesbc(debug)# debug esbc submngr

#Включение логирования модуля esbc_voip_guard:
vesbc(debug)# debug esbc voip-guard

#Включение логирования модуля stun_server:
vesbc(debug)# debug esbc stun-server
```

Для отключения логирования модулей ESBC используется команда, аналогичная включению, с приставкой **no**:

```
#Выключение логирования модуля esbc_voip_guard:
vesbc(debug)# no debug esbc voip-guard
```

Для установки параметров логирования по умолчанию используется команда *no debug all*. Данная команда отключает логирование всех модулей ESBC.

> ⚠ С целью исключения повышенной нагрузки на устройство, рекомендуется отключать логирование модулей ESBC (*no debug all*) сразу после получения логов, необходимых для анализа неисправности.

9.21 Изменение количества модулей

При обработке сигнального SIP-трафика и медиапотоков RTP, ресурсы CPU используются разными модулями ESBC. Соответственно для оптимизации нагрузки на CPU предусмотрена возможность управлять количеством модулей.

При высокой нагрузке сигнальным SIP-трафиком наибольшую нагрузку на ядро CPU производит модуль sip worker, а при большом количестве одновременных вызовов (особенно в режиме транскодирования медиа) — media worker.

Поэтому для установления баланса производительности, для многоядерных систем следует использовать оптимальное количество каждого из модулей, т. к. каждый дополнительный экземпляр модуля будет использовать ресурс дополнительного ядра CPU системы.

По умолчанию в системе используется по одному экземпляру каждого модуля.

Список модулей, количество которых можно изменить:

- core
- sip worker
- sip balancer
- media worker
- media balancer
- stun server

Максимальное количество модулей определяется динамически в зависимости от количества ядер CPU.

> ⊗ После изменения количества модулей для стабильной работы необходим перезапуск ПО ESBC.

> ⚠ Заданное в конфигурации количество модулей не изменяется при увеличении/уменьшении количества ядер CPU системы.

Описание всех команд для настройки количества модулей приведено в разделе Общие настройки ESBC.

Пример:

```
vesbc#
vesbc# config
vesbc(config)# esbc

#Переход в общие настройки:
vesbc(config-esbc)# general
vesbc(config-esbc-general)#

#Увеличение количества медиа-воркеров до 2:
vesbc(config-esbc-general)# count media worker 2
vesbc(config-esbc-general)#

#Применение и подтверждение изменений:
vesbc(config-esbc-general)# do commit
2024-09-09T05:26:55+00:00 %SYS-W-EVENT: WARNING!!! After changing ESBC modules count, the
system may work unstable. Please restart software.
2024-09-09T05:26:57+00:00 snmpd restarted
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
2024-09-09T05:26:58+00:00 %CLI-I-CRIT: user admin from console  input: do commit
vesbc(config-esbc-general)# do confirm
Configuration has been confirmed. Commit timer canceled.
2024-09-09T05:27:01+00:00 %CLI-I-CRIT: user admin from console  input: do confirm
vesbc(config-esbc-general)#

#Перезапуск ПО ESBC для корректного перераспределения модулей:
vesbc(config-esbc-general)# do reload esbc force
Do you really want to reload esbc now? (y/N): y
```

⚠ Для вывода предупреждения о необходимости перезапуска нужно, чтобы уровень syslog severity был не ниже warning.

9.22 Настройка VPN (PPTP и L2TP over IPSec)

Для организации абонентских и транковых подключений, с целью передачи сигнального (SIP) и медиа (RTP) трафика через VPN-соединение, ESBC поддерживает работу в следующих режимах:

- PPTP-сервер
- PPTP-клиент
- L2TP-сервер
- L2TP-клиент

Подробное описание настройки каждого из режимов приведено в разделе Управление удаленным доступом документации ESR.

Описание всех команд приведено в разделе Управление VPN. Настройки удаленного доступа справочника команд CLI.

Общий порядок настройки VPN:

1. Базовая настройка сети (конфигурация IP-адресов интерфейсов, настройка маршрутизации и т. д.).
2. Настройка туннелирования (конфигурация PPTP/L2TP-сервера, и/или настройка PPTP/L2TP-туннелей).
3. Настройка firewall для прохождения протоколов туннелирования (разрешить TCP-порт 1723, протокол GRE(47) для PPTP и UDP-порты 500, 1701, 4500 и протоколы ESP (50) и GRE (47) для L2TP).
4. Настройка транков и/или абонентских интерфейсов ESBC для работы через VPN.

⚠ Ниже приведены примеры настройки ESBC в качестве PPTP-сервера и в качестве PPTP-клиента. Настройки для L2TP-сервера и L2TP-клиента осуществляются аналогично.

9.22.1 Пример настройки PPTP-сервера для подключения SIP-транков

Задача

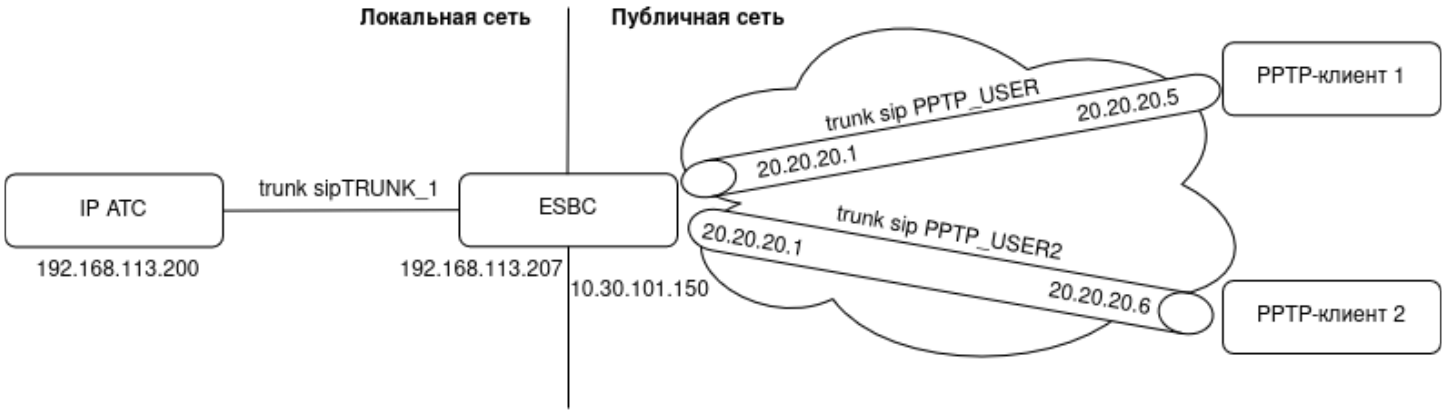
Организовать подключение двух PPTP-клиентов к ESBC и использовать эти подключения в качестве SIP-транков.

IP-адрес сервера PPTP — 20.20.20.1.

Учетная запись для подключения клиента 1 – логин: pptp_user, пароль: simplepass, IP-адрес: 20.20.20.5

Учетная запись для подключения клиента 2 – логин: pptp_user2, пароль: simplepass2, IP-адрес: 20.20.20.6

Схема



Решение

Выполнить настройку сетевых интерфейсов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description LAN
vesbc(config-if-gi)# ip address 192.168.113.207/20
vesbc(config-if-gi)# exit
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description WAN
vesbc(config-if-gi)# ip address 10.30.101.150/24
vesbc(config-if-gi)# exit
```

Выполнить настройку firewall для разрешения прохождения протокола PPTP на интерфейс gigabitethernet 1/0/2:

```
vesbc(config)# security zone WAN
vesbc(config-security-zone)# exit
vesbc(config)# security zone-pair WAN self
vesbc(config-security-zone-pair)# rule 10
vesbc(config-security-zone-pair-rule)# action permit
vesbc(config-security-zone-pair-rule)# match protocol gre
vesbc(config-security-zone-pair-rule)# enable
vesbc(config-security-zone-pair-rule)# exit
vesbc(config-security-zone-pair)# rule 20
vesbc(config-security-zone-pair-rule)# action permit
vesbc(config-security-zone-pair-rule)# match protocol tcp
vesbc(config-security-zone-pair-rule)# match destination-port port-range 1723
vesbc(config-security-zone-pair-rule)# enable
vesbc(config-security-zone-pair-rule)# exit
vesbc(config-security-zone-pair)# exit
vesbc(config)#
```

Поместить интерфейс gigabitethernet 1/0/2 в зону безопасности WAN:

```
vesbc(config)#
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# security-zone WAN
vesbc(config-if-gi)# exit
vesbc(config)#
```

Создать зону безопасности, к которой будут относиться сессии пользователей:

```
vesbc(config)#
vesbc(config)# security zone VPN_USERS
vesbc(config-security-zone)# exit
vesbc(config)#
```

Выполнить настройку PPTP-сервера:

```
vesbc(config)# remote-access pptp PPTP_SERVER
vesbc(config-pptp-server)# authentication mode local

#IP-адрес PPTP-сервера:
vesbc(config-pptp-server)# local-address ip-address 20.20.20.1

#Диапазон IP-адресов, которые будут выдаваться PPTP-клиентам:
vesbc(config-pptp-server)# remote-address address-range 20.20.20.5-20.20.20.6

#IP-адрес ESBC для организации подключения по протоколу PPTP:
vesbc(config-pptp-server)# outside-address ip-address 10.30.101.150
vesbc(config-pptp-server)# security-zone VPN_USERS

#Создание и настройка учетной записи клиента pptp_user:
vesbc(config-pptp-server)# username pptp_user
vesbc(config-ppp-user)# password ascii-text simplepass

#Указать, какой именно IP-адрес из диапазона будет выдан клиенту pptp_user:
vesbc(config-ppp-user)# remote address 20.20.20.5
vesbc(config-ppp-user)# enable
vesbc(config-ppp-user)# exit

#Создание и настройка учетной записи клиента pptp_user2:
vesbc(config-pptp-server)# username pptp_user2
vesbc(config-ppp-user)# password ascii-text simplepass2

#Указать, какой именно IP-адрес из диапазона будет выдан клиенту pptp_user2:
vesbc(config-ppp-user)# remote address 20.20.20.6
vesbc(config-ppp-user)# enable
vesbc(config-ppp-user)# exit
vesbc(config-pptp-server)# enable
vesbc(config-pptp-server)# exit
vesbc(config)#
```

Для использования подключенных PPTP-клиентов в качестве SIP-транков следует указывать IP-адрес, который будет выдаваться сервером PPTP каждому клиенту в явном виде (команда *remote address* в конфигурации **ppp-user**). Иначе IP-адреса, выдаваемые PPTP-клиентам, будут выдаваться в случайном порядке из диапазона, указанного в настройках PPTP-сервера. В таком случае невозможно будет настроить SIP-транк до конкретного клиента, т. к. для работы в режиме транка требуется указание адреса встречной стороны.
Для использования подключенных PPTP-клиентов в качестве SIP-абонентов, указывать IP-адрес в явном виде не требуется.

Выполнить настройку транков ESBC для работы с клиентами 1 и 2:

```
esbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_PPTP

#Указать IP-адрес сервера PPTP:
vesbc(config-esbc-media-resource)# ip address 20.20.20.1
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# sip transport PPTP_1

#Указать IP-адрес сервера PPTP:
vesbc(config-esbc-sip-transport)# ip address 20.20.20.1
vesbc(config-esbc-sip-transport)# port 5070
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# sip transport PPTP_2

#Указать IP-адрес сервера PPTP:
vesbc(config-esbc-sip-transport)# ip address 20.20.20.1
vesbc(config-esbc-sip-transport)# port 5071
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# trunk sip PPTP_USER
vesbc(config-esbc-trunk-sip)# sip transport PPTP_1
vesbc(config-esbc-trunk-sip)# media resource 1 MEDIA_PPTP

#Указать IP-адрес PPTP-клиента pptp_user:
vesbc(config-esbc-trunk-sip)# remote address 20.20.20.5
vesbc(config-esbc-trunk-sip)# remote port 5080
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# trunk sip PPTP_USER2
vesbc(config-esbc-trunk-sip)# sip transport PPTP_2
vesbc(config-esbc-trunk-sip)# media resource 1 MEDIA_PPTP

#Указать IP-адрес PPTP-клиента pptp_user2:
vesbc(config-esbc-trunk-sip)# remote address 20.20.20.6
vesbc(config-esbc-trunk-sip)# remote port 5081
vesbc(config-esbc-trunk-sip)# exit
```

Выполнить настройку ESBC для транка TRUNK_1:

```
vesbc(config-esbc)#
vesbc(config-esbc)# media resource MEDIA_TRUNK_1
vesbc(config-esbc-media-resource)# ip address 192.168.113.207
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-sip-transport)# ip address 192.168.113.207
vesbc(config-esbc-sip-transport)# port 5090
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# route-table TO_PPTP_USER
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk PPTP_USER
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-trunk-sip)# route-table TO_PPTP_USER
vesbc(config-esbc-trunk-sip)# media resource 1 MEDIA_TRUNK_1
vesbc(config-esbc-trunk-sip)# remote address 192.168.113.200
vesbc(config-esbc-trunk-sip)# remote port 5091
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# exit
vesbc(config)#
```

Настроить маршрутизацию вызовов из транка PPTP_USER в транк TRUNK_1:

```
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TRUNK_1
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
vesbc(config-esbc)# trunk sip PPTP_USER
vesbc(config-esbc-trunk-sip)# route-table TO_TRUNK_1
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# exit
```

Настройка маршрутизации для транка PPTP_USER2 выполняется аналогично.

9.22.2 Пример настройки PPTP-клиента для подключения SIP-транков

Задача

Организовать подключение в режиме PPTP-клиента к вышестоящему серверу и использовать это подключение в качестве SIP-транка.

IP-адрес сервера PPTP — 10.30.101.150

Учетная запись для подключения клиента — логин: pptp_user, пароль: simplepass.

Схема

Решение

Выполнить настройку сетевых интерфейсов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description LAN
vesbc(config-if-gi)# ip address 192.168.113.208/20
vesbc(config-if-gi)# exit
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description WAN
vesbc(config-if-gi)# ip address 10.30.101.151/24
vesbc(config-if-gi)# exit
vesbc(config)#
```

Создать зону безопасности для PPTP-туннеля:

```
vesbc(config)# security zone PPTP_TENNEL
vesbc(config-security-zone)# exit
vesbc(config)#
```

Создать и настроить PPTP-туннель:

```
vesbc(config)#
vesbc(config)# tunnel pptp 1
vesbc(config-pptp)# security-zone PPTP_TENNEL

#Указать учетные данные пользователя pptp_user:
vesbc(config-pptp)# username pptp_user password ascii-text simplepass

#Указать IP-адрес PPTP-сервера:
vesbc(config-pptp)# remote address 10.30.101.150
vesbc(config-pptp)# ignore default-route
vesbc(config-pptp)# enable
vesbc(config-pptp)# exit
vesbc(config)#
```

Выполнить настройку транка ESBC для работы через PPTP-туннель:

```
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_PPTP

#Указать туннель pptp 1:
vesbc(config-esbc-media-resource)# ip address interface pptp 1
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# sip transport PPTP

#Указать туннель pptp 1:
vesbc(config-esbc-sip-transport)# ip address interface pptp 1
vesbc(config-esbc-sip-transport)# port 5080
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# trunk sip PPTP
vesbc(config-esbc-trunk-sip)# sip transport PPTP
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_PPTP

#Указать Remote IP PPTP-туннеля:
vesbc(config-esbc-trunk-sip)# remote address 20.20.20.1
vesbc(config-esbc-trunk-sip)# remote port 5070
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)#
```

IP-адреса (Local и Remote) PPTP-туннеля можно узнать командой *show tunnels status*.

```
vesbc# show tunnels status
Tunnel       Admin    Link     MTU      Local IP     Remote IP     Last change
             State    State                                         (d,h:m:s)
---------    -----    -----    ------   -----------  -----------   -------------
pptp 1       Up       Up       1500     20.20.20.5   20.20.20.1    00,00:13:35
```

Выполнить настройку ESBC для транка TRUNK_1:

```
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_TRUNK_1
vesbc(config-esbc-media-resource)# ip address 192.168.113.208
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-sip-transport)# ip address 192.168.113.208
vesbc(config-esbc-sip-transport)# port 5091
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# route-table TO_PPTP
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk PPTP
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
vesbc(config-esbc)# trunk sip TRUNK_1
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_1
vesbc(config-esbc-trunk-sip)# route-table TO_PPTP
vesbc(config-esbc-trunk-sip)# media resource 1 MEDIA_TRUNK_1
vesbc(config-esbc-trunk-sip)# remote address 192.168.113.200
vesbc(config-esbc-trunk-sip)# remote port 5092
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# exit
vesbc(config)#
```

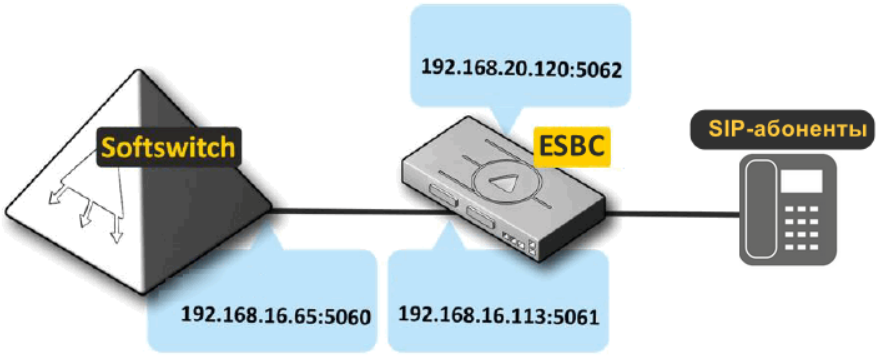
Настроить маршрутизацию вызовов из транка PPTP в транк TRUNK_1:

```
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TRUNK_1
vesbc(config-esbc-route-table)# rule 1
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_1
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
vesbc(config-esbc)# trunk sip PPTP
vesbc(config-esbc-trunk-sip)# route-table TO_TRUNK_1
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# exit
```

9.23 Примеры настройки ESBC

9.23.1 Настройка для SIP-абонентов

Схема применения:



Описание:

SIP-абоненты (IP-телефон/VoIP шлюз/Мобильный SIP-клиент и т. д.) отправляют сообщение на IP-адрес 192.168.20.120 порт 5062, ESBC пересылает данный трафик с IP-адреса 192.168.16.113 порт 5061 на адрес Softswitch (IP АТС/SIP-proxy и т. д) 192.168.16.65 порт 5060.

Для реализации данной схемы общий алгоритм настройки следующий:

1. Настроить сетевые интерфейсы на ESBC в сторону SSW и в сторону SIP-абонентов.
2. Создать SIP-транспорт в сторону SSW и SIP-абонентов.
3. Создать медиаресурсы для обоих направлений, назначить им диапазон портов для передачи голоса.
4. Создать абонентский интерфейс и SIP-транк.
5. Создать правила, по которым будут маршрутизироваться вызовы от абонентов до SSW.

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.16.113/24
vesbc(config-if-gi)# ip firewall disable
```

2. Настроить IP-адрес на внешнем интерфейсе в сторону абонентов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "ABONENTS"
vesbc(config-if-gi)# ip address 192.168.20.120/24
```

3. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5061
```

4. Создать SIP-транспорт в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# port 5062
```

5. Создать медиаресурсы для согласования и передачи голоса на плече SSW --- ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC --- Абонентский шлюз/SIP-абоненты:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_ABONENTS
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать SIP-транк в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.65
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

8. Создать абонентский интерфейс в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_ABONENTS

#Если абоненты находятся за NAT выполнить команду:
vesbc(config-esbc-user-interface-sip)# nat comedia-mode on
```

9. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с абонентов, будут маршрутизироваться на SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW
```

10. Привязать созданную таблицу маршрутизации к абонентскому интерфейсу:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
```

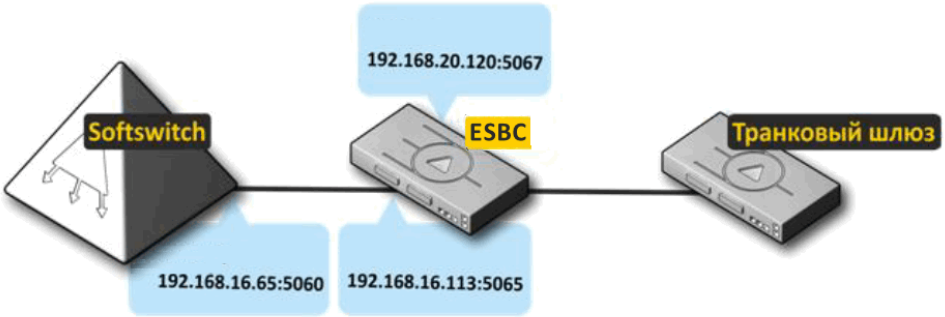
11. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

⚠ В приведённой схеме описаны базовые настройки.

9.23.2 Настройка для SIP-транков

Схема применения:



Описание:

Транковый шлюз (IP АТС/ SIP-proxy/Удаленный SSW и др.) отправляет сообщения с IP-адреса 192.168.20.99 порта 5060 на IP-адрес 192.168.20.120 порт 5067, ESBC пересылает данный трафик с IP-адреса 192.168.16.113 порта 5065 на адрес Softswitch 192.168.16.65 порт 5060. И в обратную сторону SSW отправляет сообщения с IP-адреса 192.168.16.65 порта 5060 на IP-адрес 192.168.16.113 порт 5065, ESBC пересылает данный трафик с IP-адреса 192.168.20.120 порта 5067 на адрес транкового шлюза 192.168.20.99 порт 5060.

Для реализации данной схемы общий алгоритм настройки следующий:

1. Настроить сетевые интерфейсы на ESBC в сторону SSW и в сторону транкового шлюза.
2. Создать SIP-транспорт в сторону SSW и транкового шлюза.
3. Создать медиаресурсы для обоих направлений, назначить им диапазон портов для передачи голоса.
4. Создать 2 SIP-транка в сторону SSW и в сторону транкового шлюза.
5. Создать правила, по которым будут маршрутизироваться вызовы от транкового шлюза до SSW и наоборот от SSW до транкового шлюза.

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.16.113/24
```

2. Настроить IP-адрес на интерфейсе в сторону транкового шлюза:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "TRUNK_GATEWAY"
vesbc(config-if-gi)# ip address 192.168.20.120/24
```

3. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# port 5065
```

4. Создать SIP-транспорт в сторону транкового шлюза:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_TRUNK_GATEWAY
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# port 5067
```

5. Создать медиаресурсы для согласования и передачи голоса на плече SSW --- ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC --- Транковый шлюз:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_TRUNK_GATEWAY
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать SIP-транк в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.65
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

8. Создать SIP-транк в сторону транкового шлюза:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_GATEWAY
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_TRUNK_GATEWAY
vesbc(config-esbc-trunk-sip)# remote address 192.168.20.99
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_TRUNK_GATEWAY
```

9. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с транкового шлюза, будут маршрутизироваться на SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW
```

10. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с SSW, будут маршрутизироваться на транковый шлюз:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_TRUNK_GATEWAY
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_GATEWAY
```

11. Привязать созданные таблицы маршрутизации к транкам:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# route-table TO_TRUNK_GATEWAY
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# trunk sip TRUNK_GATEWAY
vesbc(config-esbc-trunk-sip)# route-table TO_SSW
```

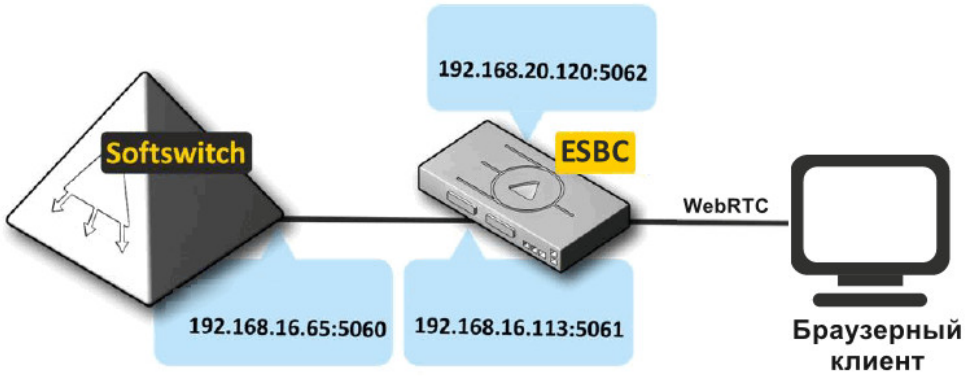
12. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

⚠ В приведённой схеме описаны базовые настройки.

9.23.3 Настройка для SIP-абонентов, использующих WebRTC

Схема применения:



Описание:

SIP-абоненты (WEB, Desktop-клиенты) отправляют сообщения на IP-адрес 192.168.20.120 порт 5062 с помощью WebSocket Secure, ESBC отправляет по TCP данный трафик с IP-адреса 192.168.16.113 порт 5061 на адрес Softswitch (IP ATC/SIP-proxy и т. д) 192.168.16.65 порт 5060.

Для реализации данной схемы общий алгоритм настройки следующий:

1. Настроить сетевые интерфейсы на ESBC в сторону SSW и в сторону SIP-абонентов.
2. Создать SIP-транспорт в режиме TCP (only/prefer) в сторону SSW и SIP-транспорт в режиме WSS для SIP-абонентов.
3. Создать медиаресурсы для обоих направлений, назначить им диапазон портов для передачи голоса.
4. Создать медиапрофиль для SIP-абонентов и включить на нём шифрование DTLS-SRTP.
5. Создать абонентский интерфейс и SIP-транк.
6. Создать правила, по которым будут маршрутизироваться вызовы от абонентов до SSW.

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на интерфейсе в сторону SSW:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# description "SSW"
vesbc(config-if-gi)# ip address 192.168.16.113/24
vesbc(config-if-gi)# ip firewall disable
```

2. Настроить IP-адрес на внешнем интерфейсе в сторону абонентов:

```
vesbc# configure
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# description "ABONENTS"
vesbc(config-if-gi)# ip address 192.168.20.120/24
```

3. Создать SIP-транспорт в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_SSW
vesbc(config-esbc-sip-transport)# ip address 192.168.16.113
vesbc(config-esbc-sip-transport)# mode tcp-prefer
vesbc(config-esbc-sip-transport)# port 5061
```

4. Создать SIP-транспорт в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-sip-transport)# ip address 192.168.20.120
vesbc(config-esbc-sip-transport)# mode wss
vesbc(config-esbc-sip-transport)# port 5062
```

⚠ Если абоненты используют WebSocket, а не WebSocket Secure, то необходимо выбрать **mode ws** в настройках SIP-транспорта для абонентов.

5. Создать медиаресурсы для согласования и передачи голоса на плече SSW — ESBC:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_SSW
vesbc(config-esbc-media-resource)# ip address 192.168.16.113

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
vesbc(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC — Абонентский шлюз/SIP-абоненты:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media resource MEDIA_ABONENTS
vesbc(config-esbc-media-resource)# ip address 192.168.20.120
```

7. Создать медиапрофиль с шифрованием DTLS-SRTP для SIP-абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# media profile MEDIA_PROFILE_ABONENTS
vesbc(config-esbc-media-profile)# srtp mode mandatory
vesbc(config-esbc-media-profile)# srtp keying dtls-srtp
```

8. Создать SIP-транк в сторону SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# trunk sip TRUNK_SSW
vesbc(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
vesbc(config-esbc-trunk-sip)# remote address 192.168.16.65
vesbc(config-esbc-trunk-sip)# remote port 5060
vesbc(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
```

9. Создать абонентский интерфейс в сторону абонентов:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# sip transport TRANSPORT_ABONENTS
vesbc(config-esbc-user-interface-sip)# media resource 0 MEDIA_ABONENTS
vesbc(config-esbc-user-interface-sip)# media profile MEDIA_PROFILE_ABONENTS

#Если абоненты находятся за NAT, выполнить команду:
vesbc(config-esbc-user-interface-sip)# nat comedia-mode on
```

10. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с абонентов, будут маршрутизироваться на SSW:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# route-table TO_SSW
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW
```

11. Привязать созданную таблицу маршрутизации к абонентскому интерфейсу:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# user-interface sip ABONENTS
vesbc(config-esbc-user-interface-sip)# route-table TO_SSW
```

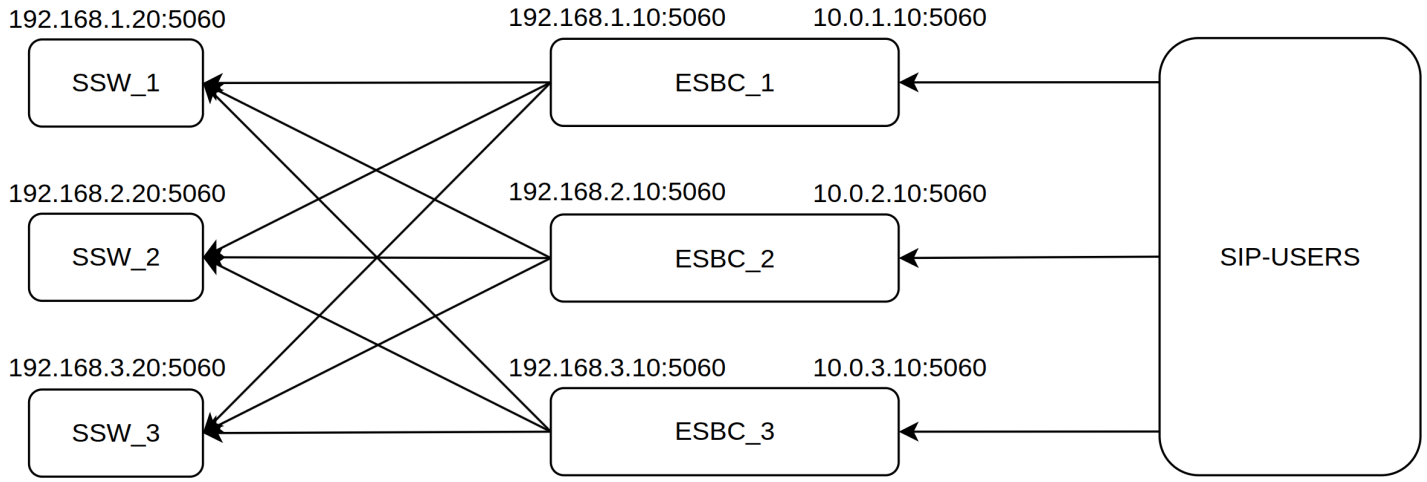
12. Применить конфигурацию и подтвердить изменения:

```
vesbc# commit
vesbc# confirm
```

⚠ В приведённой схеме описаны базовые настройки.

9.23.4 Пример настройки балансировки сессий для резервирования направлений

Схема подключения:



Описание:
SIP-абоненты регистрируются на SSW через ESBC. Каждый ESBC должен направлять запросы регистрации на SSW исходя из номеров абонентов: на SSW_1 должны регистрироваться абоненты с номерами, начинающимися с цифры «1», на SSW_2 — с номерами, начинающимися с цифры «2», на SSW_3 — с номерами, начинающимися с цифры «3». Если одно из направлений отказывает, то ESBC должен перенаправлять регистрации на оставшиеся SSW. При восстановлении доступа к отказавшему SSW на него нужно отправлять регистрации, которые были распределены по другим серверам. Доступность направлений будет контролироваться исходящими OPTIONS.

Для реализации данной схемы общий алгоритм настройки следующий:

1. Настроить сетевые интерфейсы на каждом ESBC в сторону SSW и в сторону SIP-абонентов.
2. Создать SIP-транспорт в сторону SSW и SIP-транспорт в режиме для SIP-абонентов.
3. Создать медиаресурсы для обоих направлений, назначить им диапазон портов для передачи голоса.
4. Создать абонентский интерфейс и SIP-транки.
5. Создать правила, по которым будут маршрутизироваться и балансироваться вызовы от абонентов до SSW.

Порядок конфигурирования ESBC:

1. Настроить IP-адрес на интерфейсе в сторону трёх SSW на каждом ESBC:

Для ESBC_1:

```
ESBC-1# configure
ESBC-1(config)# interface gigabitethernet 1/0/1
ESBC-1(config-if-gi)# description "SSW"
ESBC-1(config-if-gi)# ip address 192.168.1.10/24
ESBC-1(config-if-gi)# ip firewall disable
```

Для ESBC_2:

```
ESBC-2# configure
ESBC-2(config)# interface gigabitethernet 1/0/1
ESBC-2(config-if-gi)# description "SSW"
ESBC-2(config-if-gi)# ip address 192.168.2.10/24
ESBC-2(config-if-gi)# ip firewall disable
```

Для ESBC_3:

```
ESBC-3# configure
ESBC-3(config)# interface gigabitethernet 1/0/1
ESBC-3(config-if-gi)# description "SSW"
ESBC-3(config-if-gi)# ip address 192.168.3.10/24
ESBC-3(config-if-gi)# ip firewall disable
```

2. Настроить IP-адрес на внешнем интерфейсе в сторону абонентов на каждом ESBC:

Для ESBC_1:

```
ESBC-1# configure
ESBC-1(config)# interface gigabitethernet 1/0/2
ESBC-1(config-if-gi)# description "ABONENTS"
ESBC-1(config-if-gi)# ip address 10.0.1.10/24
```

Для ESBC_2:

```
ESBC-2# configure
ESBC-2(config)# interface gigabitethernet 1/0/2
ESBC-2(config-if-gi)# description "ABONENTS"
ESBC-2(config-if-gi)# ip address 10.0.2.10/24
```

Для ESBC_3:

```
ESBC-3# configure
ESBC-3(config)# interface gigabitethernet 1/0/2
ESBC-3(config-if-gi)# description "ABONENTS"
ESBC-3(config-if-gi)# ip address 10.0.3.10/24
```

3. Создать SIP-транспорт в сторону SSW на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# sip transport TRANSPORT_SSW
ESBC-1(config-esbc-sip-transport)# ip address 192.168.1.10
ESBC-1(config-esbc-sip-transport)# port 5060
```

Для ESBC_2:

```
ESBC-2#
ESBC-2# configure
ESBC-2(config)# esbc
ESBC-2(config-esbc)# sip transport TRANSPORT_SSW
ESBC-2(config-esbc-sip-transport)# ip address 192.168.2.10
ESBC-2(config-esbc-sip-transport)# port 5060
```

Для ESBC_3:

```
ESBC-3#
ESBC-3# configure
ESBC-3(config)# esbc
ESBC-3(config-esbc)# sip transport TRANSPORT_SSW
ESBC-3(config-esbc-sip-transport)# ip address 192.168.3.10
ESBC-3(config-esbc-sip-transport)# port 5060
```

4. Создать SIP-транспорт в сторону абонентов на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# sip transport TRANSPORT_ABONENTS
ESBC-1(config-esbc-sip-transport)# ip address 10.0.1.10
ESBC-1(config-esbc-sip-transport)# port 5060
```

Для ESBC_2:

```
ESBC-2#
ESBC-2# configure
ESBC-2(config)# esbc
ESBC-2(config-esbc)# sip transport TRANSPORT_ABONENTS
ESBC-2(config-esbc-sip-transport)# ip address 10.0.2.10
ESBC-2(config-esbc-sip-transport)# port 5060
```

Для ESBC_3:

```
ESBC-3#
ESBC-3# configure
ESBC-3(config)# esbc
ESBC-3(config-esbc)# sip transport TRANSPORT_ABONENTS
ESBC-3(config-esbc-sip-transport)# ip address 10.0.3.10
ESBC-3(config-esbc-sip-transport)# port 5060
```

5. Создать медиаресурсы для согласования и передачи голоса на плече SSW — ESBC на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# media resource MEDIA_SSW
ESBC-1(config-esbc-media-resource)# ip address 192.168.1.10

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
ESBC-1(config-esbc-media-resource)# port-range 1024-65535
```

Для ESBC_2:

```
ESBC-2#
ESBC-2# configure
ESBC-2(config)# esbc
ESBC-2(config-esbc)# media resource MEDIA_SSW
ESBC-2(config-esbc-media-resource)# ip address 192.168.2.10

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
ESBC-2(config-esbc-media-resource)# port-range 1024-65535
```

Для ESBC_3:

```
ESBC-3#
ESBC-3# configure
ESBC-3(config)# esbc
ESBC-3(config-esbc)# media resource MEDIA_SSW
ESBC-3(config-esbc-media-resource)# ip address 192.168.3.10

#Указать диапазон портов, который будет выделяться на ESBC для передачи голоса. Данная команда
необязательная. Если ее не указывать, будет использоваться диапазон портов 8000-65535.
ESBC-3(config-esbc-media-resource)# port-range 1024-65535
```

6. Создать медиаресурсы для согласования и передачи голоса на плече ESBC --- Абонентский шлюз/SIP-абоненты на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# media resource MEDIA_ABONENTS
ESBC-1(config-esbc-media-resource)# ip address 10.0.1.10
```

Для ESBC_2:

```
ESBC-2#
ESBC-2# configure
ESBC-2(config)# esbc
ESBC-2(config-esbc)# media resource MEDIA_ABONENTS
ESBC-2(config-esbc-media-resource)# ip address 10.0.2.10
```

Для ESBC_3:

```
ESBC-3#
ESBC-3# configure
ESBC-3(config)# esbc
ESBC-3(config-esbc)# media resource MEDIA_ABONENTS
ESBC-3(config-esbc-media-resource)# ip address 10.0.3.10
```

7. Создать SIP-профиль для включения контроля направлений по OPTIONS на каждом ESBC:
Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# sip profile SIP_PROFILE
ESBC-1(config-esbc-sip-profile)# keepalive enable
ESBC-1(config-esbc-sip-profile)# keepalive success-interval 10
ESBC-1(config-esbc-sip-profile)# keepalive failed-interval 10
ESBC-1(config-esbc-sip-profile)# exit
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

8. Создать SIP-транки в сторону SSW_1, SSW_2 и SSW_3 на каждом ESBC и добавить в них настроенный SIP-профиль:
Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc

#Создаём транк до SSW_1:
ESBC-1(config-esbc)# trunk sip TRUNK_SSW_1
ESBC-1(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
ESBC-1(config-esbc-trunk-sip)# remote address 192.168.1.20
ESBC-1(config-esbc-trunk-sip)# remote port 5060
ESBC-1(config-esbc-trunk-sip)# sip profile SIP_PROFILE
ESBC-1(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
ESBC-1(config-esbc-trunk-sip)# exit

#Создаём транк до SSW_2:
ESBC-1(config-esbc)# trunk sip TRUNK_SSW_2
ESBC-1(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
ESBC-1(config-esbc-trunk-sip)# remote address 192.168.2.20
ESBC-1(config-esbc-trunk-sip)# remote port 5060
ESBC-1(config-esbc-trunk-sip)# sip profile SIP_PROFILE
ESBC-1(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
ESBC-1(config-esbc-trunk-sip)# exit

#Создаём транк до SSW_3:
ESBC-1(config-esbc)# trunk sip TRUNK_SSW_3
ESBC-1(config-esbc-trunk-sip)# sip transport TRANSPORT_SSW
ESBC-1(config-esbc-trunk-sip)# remote address 192.168.3.20
ESBC-1(config-esbc-trunk-sip)# remote port 5060
ESBC-1(config-esbc-trunk-sip)# sip profile SIP_PROFILE
ESBC-1(config-esbc-trunk-sip)# media resource 0 MEDIA_SSW
ESBC-1(config-esbc-trunk-sip)# exit
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

9. На каждом ESBC создать транковые группы с балансировкой round robin и включить в неё все созданные транки:
Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# trunk-group TRUNK_GROUP
ESBC-1(config-esbc-trunk-group)# balancing round-robin
ESBC-1(config-esbc-trunk-group)# trunk 1 TRUNK_SSW_1
ESBC-1(config-esbc-trunk-group)# trunk 2 TRUNK_SSW_2
ESBC-1(config-esbc-trunk-group)# trunk 3 TRUNK_SSW_3
ESBC-1(config-esbc-trunk-group)# exit
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

10. Создать абонентский интерфейс в сторону абонентов на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# user-interface sip ABONENTS
ESBC-1(config-esbc-user-interface-sip)# sip transport TRANSPORT_ABONENTS
ESBC-1(config-esbc-user-interface-sip)# media resource 0 MEDIA_ABONENTS

#Если абоненты находятся за NAT, выполнить команду:
ESBC-1(config-esbc-user-interface-sip)# nat comedia-mode on
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

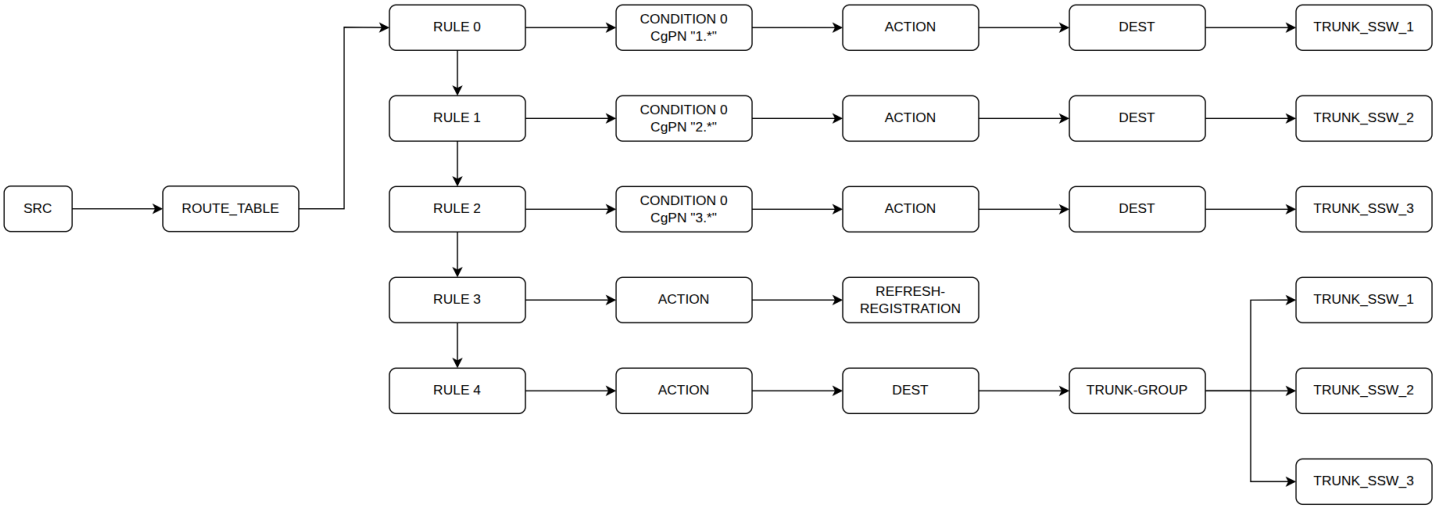
11. Создать таблицу маршрутизации и добавить в нее правила, по которым вызовы, приходящие с абонентов, будут балансироваться между SSW_1, SSW_2 и SSW_3 на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# route-table ROUTE_TABLE

#Создаём правило, направляющее запросы на транк TRUNK_SSW_1 для абонентов с номером, который
начинается с 1:
ESBC-1(config-esbc-route-table)# rule 0
ESBC-1(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW_1
ESBC-1(config-esbc-route-table-rule)# condition 0 cgpn pattern "1.*"
ESBC-1(config-esbc-route-table-rule)# exit

#Создаём правило, направляющее запросы на транк TRUNK_SSW_2 для абонентов с номером, который
начинается с 2:
ESBC-1(config-esbc-route-table)# rule 1
ESBC-1(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW_2
ESBC-1(config-esbc-route-table-rule)# condition 0 cgpn pattern "2.*"
ESBC-1(config-esbc-route-table-rule)# exit

#Создаём правило, направляющее запросы на транк TRUNK_SSW_3 для абонентов с номером, который
начинается с 3:
ESBC-1(config-esbc-route-table)# rule 2
ESBC-1(config-esbc-route-table-rule)# action direct-to-trunk TRUNK_SSW_3
ESBC-1(config-esbc-route-table-rule)# condition 0 cgpn pattern "3.*"
ESBC-1(config-esbc-route-table-rule)# exit

#Создаём правило для закрепления регистраций на направлении. Правило нужно, чтобы при
обновлении регистрации абонент не регистрировался на другом сервере, а обновлял регистрацию
там, где она уже есть. В случае если основной сервер станет недоступен и абонент
зарегистрируется на другом SSW:
ESBC-1(config-esbc-route-table)# rule 3
ESBC-1(config-esbc-route-table-rule)# action refresh-registration
ESBC-1(config-esbc-route-table-rule)# exit

#Создаём правило, направляющее запросы на транковую группу TRUNK_GROUP для перераспределения
абонентов в случае отказа основного SSW:
ESBC-1(config-esbc-route-table)# rule 4
ESBC-1(config-esbc-route-table-rule)# action direct-to-trunk-group TRUNK_GROUP
ESBC-1(config-esbc-route-table-rule)# exit
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

Схема таблицы маршрутизации:



12. Привязать созданную таблицу маршрутизации к абонентскому интерфейсу на каждом ESBC:

Для ESBC_1:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# user-interface sip ABONENTS
ESBC-1(config-esbc-user-interface-sip)# route-table ROUTE_TABLE
```

Для ESBC_2 и ESBC_3 аналогичные настройки.

13. Применить конфигурацию и подтвердить изменения:

```
ESBC-1# commit
ESBC-1# confirm
```

⚠ В приведённой схеме описаны базовые настройки.

Теперь регистрации будут распределяться по серверам исходя из номера абонента. При выходе из строя одного из SSW абоненты с него будут регистрироваться на других серверах. В случае восстановления доступа к отказавшему SSW на него будут маршрутизироваться регистрации, которые были распределены по другим серверам. Также, если регистрация придёт на другой ESBC либо он выйдет из строя, другие ESBC смогут отправить регистрацию на нужный SSW.

10 Управление интерфейсами

Алгоритм и примеры настройки функций управления интерфейсами см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

11 Управление туннелированием

Алгоритм и примеры настройки функций управления туннелированием см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

12 Управление функциями второго уровня (L2)

Алгоритм и примеры настройки управления функциями второго уровня (L2) см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

13 Управление QoS

Настройка классификации DSCP (Layer 3) для сигнализации SIP, а также для аудио/видеотрафика (RTP) приведена в разделе Управление ESBC. Настройка QoS.

Настройка прочих параметров QoS приведена в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

14 Управление маршрутизацией

Алгоритм и примеры настройки функций управления маршрутизацией см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

15 Управление технологией MPLS

Управление технологией MPLS описано в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

16 Управление безопасностью

- Настройка AAA
  - Алгоритм настройки локальной аутентификации
  - Алгоритм настройки AAA по протоколу RADIUS
  - Алгоритм настройки AAA по протоколу TACACS
  - Алгоритм настройки AAA по протоколу LDAP
  - Пример настройки аутентификации по Telnet через RADIUS-сервер
- Настройка привилегий команд
  - Алгоритм настройки
  - Пример настройки привилегий команд
- Настройка логирования и защиты от ARP-spoofing атак
  - Алгоритм настройки
  - Пример настройки защиты от ARP-spoofing атак
- Настройка логирования и защиты от сетевых атак
  - Алгоритм настройки
  - Описание механизмов защиты от атак
  - Пример настройки логирования и защиты от сетевых атак
- Конфигурирование Firewall
  - Алгоритм настройки
  - Пример настройки Firewall
  - Пример настройки Firewall по доменным именам
  - Пример настройки фильтрации приложений (DPI)
  - Пример настройки Firewall по расписанию
- Настройка списков доступа (IP ACL)
  - Алгоритм настройки
  - Пример настройки списка доступа
- Настройка списков доступа (MAC ACL)
  - Алгоритм настройки
  - Пример настройки списка доступа
- Проксирование HTTP/HTTPS-трафика
  - Алгоритм настройки группы вышестоящих серверов
  - Алгоритм настройки таблицы модификации
  - Алгоритм настройки HTTP/HTTPS-прокси сервера
  - Пример настройки HTTP-прокси
- Настройка IPS/IDS
  - Алгоритм базовой настройки
  - Алгоритм настройки автообновления правил IPS/IDS из внешних источников
  - Рекомендуемые открытые источники обновления правил
    - SSL Blacklist
    - Feodo Tracker
    - Travis Green
    - Etnetera Core
  - Пример настройки IPS/IDS с автообновлением правил
  - Алгоритм настройки базовых пользовательских правил
  - Пример настройки базовых пользовательских правил
  - Алгоритм настройки расширенных пользовательских правил
  - Пример настройки расширенных пользовательских правил

16.1 Настройка AAA

AAA (Authentication, Authorization, Accounting) – используется для описания процесса предоставления доступа и контроля над ним.

- Authentication (аутентификация) – сопоставление персоны (запроса) существующей учётной записи в системе безопасности. Осуществляется по логину, паролю.
- Authorization (авторизация, проверка полномочий, проверка уровня доступа) – сопоставление учётной записи в системе и определённых полномочий.
- Accounting (учёт) – слежение за подключением пользователя или внесенным им изменениям.

16.1.1 Алгоритм настройки локальной аутентификации

Шаг	Описание	Команда	Ключи
1	Задать список методов аутентификации по умолчанию (default)/с именем <NAME> и указать local.	**esr(config)# aaa authentication login { default | <NAME> } <METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: • local – аутентификация с помощью локальной базы пользователей; • tacacs – аутентификация по списку TACACS-серверов; • radius – аутентификация по списку RADIUS-серверов; • ldap – аутентификация по списку LDAP-серверов.
2	Задать список методов аутентификации повышения привилегий пользователей по умолчанию (default)/с именем <NAME> и указать enable.	**esr(config)# aaa authentication enable <NAME><METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: • local – аутентификация с помощью локальной базы пользователей; • tacacs – аутентификация по списку TACACS-серверов; • radius – аутентификация по списку RADIUS-серверов; • ldap – аутентификация по списку LDAP-серверов.

Шаг	Описание	Команда	Ключи
3	Указать способ перебора методов аутентификации в случае отказа (необязательно).	**esr(config)# aaa authentication mode <MODE>**	<MODE> – способы перебора методов: • chain – если сервер вернул FAIL, перейти к следующему в цепочке методу аутентификации; • break – если сервер вернул FAIL, прекратить попытки аутентификации. Если сервер недоступен, продолжить попытки аутентификации следующими в цепочке методами. Значение по умолчанию: chain.
4	Указать количество неудачных попыток аутентификации для блокировки логина пользователя и время блокировки (необязательно).	**esr(config)# aaa authentication attempts max-fail <COUNT> <TIME>**	<COUNT> – количество неудачных попыток аутентификации, после которых произойдет блокировка пользователя, принимает значения [1..65535]; <TIME> – интервал времени в минутах, на который будет заблокирован пользователь, принимает значения [1..65535]. Значение по умолчанию: <COUNT> – 5; <TIME> – 300.
5	Включить запрос на смену пароля по умолчанию для пользователя admin (необязательно).	**esr(config)# security passwords default-expired**	
6	Включить режим запрета на использование ранее установленных паролей локальных пользователей (необязательно).	**esr(config)# security passwords history <COUNT>**	<COUNT> – количество паролей, сохраняемых в памяти маршрутизатора. Принимает значение в диапазоне [1..15]. Значение по умолчанию: 0.

Шаг	Описание	Команда	Ключи
7	Установить время действия пароля локального пользователя (необязательно).	**esr(config)# security passwords lifetime <TIME>**	<TIME> – интервал времени действия пароля в днях. Принимает значение в диапазоне [1..365]. По умолчанию: время действия пароля локального пользователя не ограничено.
8	Установить ограничение на минимальную длину пароля локального пользователя и ENABLE-пароля (необязательно).	**esr(config)# security passwords min-length <NUM>**	<NUM> – минимальное количество символов в пароле. Принимает значение в диапазоне [8..128]. Значение по умолчанию: 0.
9	Установить ограничение на максимальную длину пароля локального пользователя и ENABLE-пароля (необязательно).	**esr(config)# security passwords max-length <NUM>**	<NUM> – максимальное количество символов в пароле. Принимает значение в диапазоне [8..128]. Значение по умолчанию: не ограничено.
10	Установить минимальное количество типов символов, которые должны присутствовать в пароле локального пользователя и ENABLE-пароле (необязательно).	**esr(config)# security passwords symbol-types <COUNT>**	<COUNT> – минимальное количество типов символов в пароле. Принимает значение в диапазоне [1..4]. Значение по умолчанию: 1.
11	Установить минимальное количество строчных букв в пароле локального пользователя и ENABLE-пароле (необязательно).	**esr(config)# security passwords lower-case <COUNT>**	<COUNT> – минимальное количество строчных букв в пароле локального пользователя и ENABLE-пароле. Принимает значение в диапазоне [0..128]. Значение по умолчанию: 0.
12	Установить минимальное количество прописных (заглавных) букв в пароле локального пользователя и ENABLE-пароле (необязательно).	**esr(config)# security passwords upper-case <COUNT>**	<COUNT> – минимальное количество прописных (заглавных) букв в пароле. Принимает значение в диапазоне [0..128]. Значение по умолчанию: 0.
13	Установить минимальное количество цифр в пароле локального пользователя и ENABLE-пароле (необязательно).	**esr(config)# security passwords numeric-count <COUNT>**	<COUNT> – минимальное количество цифр в пароле. Принимает значение в диапазоне [0..128]. Значение по умолчанию: 0.

Шаг	Описание	Команда	Ключи
14	Установить минимальное количество специальных символов в пароле локального пользователя и ENABLE-пароле (необязательно).	**esr(config)# security passwords special-case <COUNT>**	<COUNT> – минимальное количество специальных символов в пароле. Принимает значение в диапазоне [0..128]. Значение по умолчанию: 0.
15	Добавить пользователя в локальную базу и перейти в режим настройки параметров пользователя.	**esr(config)# username <NAME>**	<NAME> – имя пользователя, задаётся строкой до 31 символа.
16	Установить пароль пользователя.	**esr(config-user)# password { <CLEAR-TEXT> | encrypted <HASH_SHA512> }**	<CLEAR-TEXT> – пароль, задаётся строкой [8 .. 32] символов, принимает значения [0-9a-fA-F]; <HASH_SHA512> – хеш пароля по алгоритму sha512, задаётся строкой из 110 символов.
17	Установить уровень привилегий пользователя.	**esr(config-user)# privilege <PRIV>**	<PRIV> – необходимый уровень привилегий. Принимает значение [1..15].

Шаг	Описание	Команда	Ключи
18	Установить режим работы учетной записи пользователя (необязательно).	**esr(config-user)# mode <MODE>**	<MODE> – режим работы учетной записи пользователя. Может принимать значения: • cli – режим работы по умолчанию, пользователь получает доступ к интерфейсу командной строки, предназначенному для управления, просмотра состояния и мониторинга устройства; • techsupport – пользователь получает доступ к командной оболочке, в которой выполняется процедура отладки устройства совместно с специалистами технической поддержки; • sftp – пользователь используется для организации доступа к встроенному SFTP-серверу, возможность работы в какой-либо командой оболочке при этом у пользователя отсутствует.
19	Указать метод аутентификации SSH-сессий для пользователя (необязательно).	**esr(config-user)# ssh authentication method <METHOD>**	<METHOD> – метод аутентификации SSH-сессий. Может принимать значения: • password – аутентификация пользователя при открытии SSH-сессий может быть произведена только по паролю; • pubkey – аутентификация пользователя при открытии SSH-сессий может быть произведена только по публичному ключу; • both – аутентификация пользователя при открытии SSH-сессий может быть произведена как по паролю, так и по публичному ключу.

Шаг	Описание	Команда	Ключи
20	Указать имя файла публичного ключа, который будет использован при аутентификации SSH-сессии пользователя (необязательно).	**esr(config-user)# ssh pubkey <NAME>**	<NAME> – имя файла публичного ключа, расположенного в разделе crypto:public-key, задаётся строкой до 31 символа.
21	Отключить авторизацию для предустановленного пользователя admin (необязательно).	**esr(config)# no admin login enable**	
22	Перейти в режим конфигурирования соответствующего терминала.	**esr(config)# line <TYPE>**	<TYPE> – тип консоли: • console – локальная консоль; • telnet – удаленная консоль; • ssh – защищенная удаленная консоль.
23	Активировать список аутентификации входа пользователей в систему.	**esr(config-line-console)# login authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 1.
24	Активировать список аутентификации повышения привилегий пользователей.	**esr(config-line-console)# enable authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 2.
25	Задать интервал, по истечении которого будет разрываться бездействующая сессия.	**esr(config-line-console)# exec-timeout <SEC>**	<SEC> – период времени в минутах, принимает значения [1..65535].

16.1.2 Алгоритм настройки AAA по протоколу RADIUS

Шаг	Описание	Команда	Ключи
1	Задать глобальное значение кода DSCP для использования в IP-заголовках исходящих пакетов RADIUS-сервера (необязательно).	**esr(config)# radius-server dscp <DSCP>**	<DSCP> – значение кода DSCP, принимает значения в диапазоне [0..63]. Значение по умолчанию: 63.
2	Задать глобальное значение количества перезапросов к последнему активному RADIUS-серверу (необязательно).	**esr(config)# radius-server retransmit <COUNT>**	<COUNT> – количество перезапросов к RADIUS-серверу, принимает значения [1..10]. Значение по умолчанию: 1.

Шаг	Описание	Команда	Ключи
3	Задать глобальное значение интервала, по истечении которого маршрутизатор считает, что RADIUS-сервер недоступен (необязательно).	**esr(config)# radius-server timeout <SEC>**	<SEC> – период времени в секундах, принимает значения [1..30]. Значение по умолчанию: 3 секунды.
4	Добавить RADIUS-сервер в список используемых серверов и перейти в режим его конфигурирования.	**esr(config)# radius-server host { <IP-ADDR> | <IPV6-ADDR> } [vrf <VRF>]**	<IP-ADDR> – IP-адрес RADIUS-сервера, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес RADIUS-сервера, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF] <VRF> – имя экземпляра VRF, задается строкой до 31 символа.
5	Задать описание конфигурируемого RADIUS-сервера (необязательно).	**esr(config-radius-server)# description <description>**	<description> – описание RADIUS-сервера, задается строкой до 255 символов.
6	Указать количество неудачных попыток аутентификации для блокировки логина пользователя и времени блокировки (необязательно).	**esr(config-radius-server)# aaa authentication attempts max-fail <COUNT> <TIME>**	<COUNT> – количество неудачных попыток аутентификации, после которых произойдет блокировка пользователя, принимает значения [1..65535]; <TIME> – интервал времени в секундах, на который будет заблокирован пользователь, принимает значения [1..65535]. Значение по умолчанию: <COUNT> – 5; <TIME> – 300.
7	Задать пароль для аутентификации на удаленном RADIUS-сервере.	**esr(config-radius-server)# key ascii-text { <TEXT> | encrypted <ENCRYPTED-TEXT> }**	<TEXT> – строка [8..16] ASCII-символов; <ENCRYPTED-TEXT> – зашифрованный пароль, размером [8..16] байт, задаётся строкой [16..32] символов.

Шаг	Описание	Команда	Ключи
8	Задать приоритет использования удаленного RADIUS-сервера (необязательно).	**esr(config-radius-server)# priority <PRIORITY>**	<PRIORITY> – приоритет использования удаленного сервера, принимает значения [1..65535]. Чем ниже значение, тем приоритетнее сервер. Значение по умолчанию: 1.
9	Задать интервал, по истечении которого маршрутизатор считает, что данный RADIUS-сервер недоступен (необязательно).	**esr(config-radius-server)# timeout <SEC>**	<SEC> – период времени в секундах, принимает значения [1..30]. Значение по умолчанию: используется значение глобального таймера.
10	Задать IPv4/IPv6-адрес, который будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых RADIUS-пакетах.	**esr(config-radius-server)# source-address { <ADDR> | object-group <NETWORK_OBJ_GROUP_NAME> }**	<ADDR> – IP-адрес источника, задается в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <NETWORK_OBJ_GROUP_NAME> – список адресов, которые будут использоваться в качестве source address.
11	Задать интерфейс или туннель маршрутизатора, IPv4/IPv6-адрес которого будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых RADIUS-пакетах.	**esr(config-radius-server)# source-interface { <IF> | <TUN> }**	<IF> – имя интерфейса устройства, задаётся в виде, описанном в разделе Типы и порядок именования интерфейсов маршрутизатора; <TUN> – имя туннеля устройства, задаётся в виде, описанном в разделе Типы и порядок именования туннелей маршрутизатора.

Шаг	Описание	Команда	Ключи
12	Задать список методов аутентификации по умолчанию (default)/с именем <NAME> и указать radius.	**esr(config)# aaa authentication login { default | <NAME> } <METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: • local – аутентификация с помощью локальной базы пользователей; • tacacs – аутентификация по списку TACACS-серверов; • radius – аутентификация по списку RADIUS-серверов; • ldap – аутентификация по списку LDAP-серверов.
13	Задать список методов аутентификации повышения привилегий пользователей по умолчанию (default)/с именем <NAME> и указать radius.	**esr(config)# aaa authentication enable <NAME><METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка строка до 31 символа; • default – имя списка по умолчанию. <METHOD> – способы аутентификации: • enable – аутентификация с помощью enable-паролей; • tacacs – аутентификация по протоколу TACACS; • radius – аутентификация по протоколу RADIUS; • ldap – аутентификация по протоколу LDAP.
14	Указать способ перебора методов аутентификации в случае отказа (необязательно).	**esr(config)# aaa authentication mode <MODE>**	<MODE> – способы перебора методов: • chain – если сервер вернул FAIL, переход к следующему в цепочке методу аутентификации; • break – если сервер вернул FAIL, прекратить попытки аутентификации. Если сервер недоступен, продолжить попытки аутентификации следующими в цепочке методами. Значение по умолчанию: chain.

Шаг	Описание	Команда	Ключи
15	Сконфигурировать RADIUS в списке способов учета сессий пользователей (необязательно).	**esr(config)# aaa accounting login start-stop <METHOD 1> [<METHOD 2>]**	<METHOD> – способы учета: • tacacs – учет сессий по протоколу TACACS; • radius – учет сессий по протоколу RADIUS.
16	Перейти в режим конфигурирования соответствующего терминала.	**esr(config)# line <TYPE>**	<TYPE> – тип консоли: • console – локальная консоль; • telnet – удаленная консоль; • ssh – защищенная удаленная консоль.
17	Активировать список аутентификации входа пользователей в систему.	**esr(config-line-console)# login authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 12.
18	Активировать список аутентификации повышения привилегий пользователей.	**esr(config-line-console)# enable authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 13.

16.1.3 Алгоритм настройки AAA по протоколу TACACS

Шаг	Описание	Команда	Ключи
1	Задать глобальное значение кода DSCP для использования в IP-заголовках исходящих пакетов TACACS-сервера (необязательно).	**esr(config)# tacacs-server dscp <DSCP>**	<DSCP> – значение кода DSCP, принимает значения в диапазоне [0..63]. Значение по умолчанию: 63.
2	Задать глобальное значение интервала, по истечении которого маршрутизатор считает, что TACACS-сервер недоступен (необязательно).	**esr(config)# tacacs-server timeout <SEC>**	<SEC> – период времени в секундах, принимает значения [1..30]. Значение по умолчанию: 3 секунды.

Шаг	Описание	Команда	Ключи
3	Добавить TACACS-сервер в список используемых серверов и перейти в режим его конфигурирования.	**esr(config)# tacacs -server host { <IP-ADDR> | <IPV6-ADDR> } [vrf <VRF>]**	<IP-ADDR> – IP-адрес TACACS-сервера, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255] <IPV6-ADDR> – IPv6-адрес TACACS -сервера, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF] <VRF> – имя экземпляра VRF, задается строкой до 31 символа.
4	Задать описание конфигурируемого TACACS-сервера (необязательно).	**esr(config-tacacs-server)# description <description>**	<description> – описание TACACS-сервера, задается строкой до 255 символов.
5	Указать количество неудачных попыток аутентификации для блокировки логина пользователя и время блокировки (необязательно)	**esr(config-tacacs-server)# aaa authentication attempts max-fail <COUNT> <TIME>**	<COUNT> – количество неудачных попыток аутентификации, после которых произойдет блокировка пользователя, принимает значения [1..65535]; <TIME> – интервал времени в минутах, на который будет заблокирован пользователь, принимает значения [1..65535]. Значение по умолчанию: <COUNT> – 5; <TIME> – 300.
6	Задать пароль для аутентификации на удаленном TACACS-сервере	**esr(config-tacacs-server)# key ascii-text { <TEXT> | encrypted <ENCRYPTED-TEXT> }**	<TEXT> – строка [8..16] ASCII-символов; <ENCRYPTED-TEXT> – зашифрованный пароль, размером [8..16] байт, задаётся строкой [16..32] символов.

Шаг	Описание	Команда	Ключи
7	Задать номер порта для обмена данными с удаленным TACACS-сервером (необязательно).	**esr(config-tacacs-server)# port <PORT>**	<PORT> – номер TCP-порта для обмена данными с удаленным сервером, принимает значения [1..65535]. Значение по умолчанию: 49 для TACACS-сервера.
8	Задать приоритет использования удаленного TACACS сервера (необязательно).	**esr(config-tacacs-server)# priority <PRIORITY>**	<PRIORITY> – приоритет использования удаленного сервера, принимает значения [1..65535]. Чем ниже значение, тем приоритетнее сервер. Значение по умолчанию: 1.
9	Задать IPv4/IPv6-адрес, который будет использоваться в качестве IP/IPv6-адреса источника в отправляемых TACACS-пакетах.	**esr(config-tacacs-server)# source-address { <ADDR> | object-group <NETWORK_OBJ_GROUP_NAME> }**	<ADDR> – IP-адрес источника, задается в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <NETWORK_OBJ_GROUP_NAME> – список адресов, которые будут использоваться в качестве source address.
10	Задать интерфейс или туннель маршрутизатора, IPv4/IPv6-адрес которого будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых TACACS-пакетах.	**esr(config-tacacs-server)# source-interface { <IF> | <TUN> }**	<IF> – имя интерфейса устройства, задаётся в виде, описанном в разделе Типы и порядок именования интерфейсов маршрутизатора; <TUN> – имя туннеля устройства, задаётся в виде, описанном в разделе Типы и порядок именования туннелей маршрутизатора.

Шаг	Описание	Команда	Ключи
11	Задать список методов аутентификации по умолчанию (default)/с именем <NAME> и указать tacacs.	**esr(config)# aaa authentication login { default | <NAME> } <METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: • local – аутентификация с помощью локальной базы пользователей; • tacacs – аутентификация по списку TACACS-серверов; • radius – аутентификация по списку RADIUS-серверов; • ldap – аутентификация по списку LDAP-серверов.
12	Задать список методов аутентификации повышения привилегий пользователей по умолчанию (default)/с именем <NAME> и указать tacacs.	**esr(config)# aaa authentication enable <NAME><METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка строка до 31 символа; • default – имя списка по умолчанию. <METHOD> – способы аутентификации: • enable – аутентификация с помощью enable-паролей; • tacacs – аутентификация по протоколу TACACS; • radius – аутентификация по протоколу RADIUS; • ldap – аутентификация по протоколу LDAP.
13	Задать список методов авторизации команд, вводимых пользователем в систему по умолчанию (default)/с именем <NAME> и указать tacacs.	**esr(config)# aaa authorization commands { default | <NAME> } <METHOD 1>[<METHOD 2>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: local – авторизация с помощью локальной базы пользователей; tacacs – авторизация по списку TACACS-серверов;

Шаг	Описание	Команда	Ключи
14	Указать способ перебора методов аутентификации в случае отказа (необязательно).	**esr(config)# aaa authentication mode <MODE>**	<MODE> – способы перебора методов: • chain – если сервер вернул FAIL, переход к следующему в цепочке методу аутентификации; • break – если сервер вернул FAIL, прекратить попытки аутентификации. Если сервер недоступен, продолжить попытки аутентификации следующими в цепочке методами. Значение по умолчанию: chain.
15	Сконфигурировать список способов учета команд, введённых в CLI (необязательно).	**esr(config)# aaa accounting commands stop-only <METHOD>**	<METHOD> – способы учета: tacacs – учет введенных команд по протоколу TACACS.
16	Сконфигурировать tacacs в списке способов учета сессий пользователей (необязательно).	**esr(config)# aaa accounting login start-stop <METHOD 1> [<METHOD 2>]**	<METHOD> – способы учета: • tacacs – учет сессий по протоколу TACACS; • radius – учет сессий по протоколу RADIUS.
17	Перейти в режим конфигурирования соответствующего терминала.	**esr(config)# line <TYPE>**	<TYPE> – тип консоли: • console – локальная консоль; • telnet – удаленная консоль; • ssh – защищенная удаленная консоль.
18	Активировать список аутентификации входа пользователей в систему.	**esr(config-line-console)# login authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 11.
19	Активировать список аутентификации повышения привилегий пользователей.	**esr(config-line-console)# enable authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 12.
20	Активировать список авторизации команд вводимых пользователем в систему.	**esr(config-line-console)# commands authorization <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 13.

16.1.4 Алгоритм настройки AAA по протоколу LDAP

Шаг	Описание	Команда	Ключи
1	Задать базовый DN (Distinguished name), который будет использоваться при поиске пользователей.	**esr(config)# ldap-server base-dn <NAME>**	<NAME> – базовый DN, задается строкой до 255 символов.
2	Задать интервал, по истечении которого устройство считает, что LDAP-сервер недоступен (необязательно).	**esr(config)# ldap-server bind timeout <SEC>**	<SEC> – период времени в секундах, принимает значения [1..30]. Значение по умолчанию: 3 секунды.
3	Задать DN (Distinguished name) пользователя с правами администратора, под которым будет происходить авторизация на LDAP-сервере при поиске пользователей.	**esr(config)# ldap-server bind authenticate root-dn <NAME>**	<NAME> – DN пользователя с правами администратора, задается строкой до 255 символов.
4	Задать пароль пользователя с правами администратора, под которым будет происходить авторизация на LDAP-сервере при поиске пользователей.	**esr(config)# ldap-server bind authenticate root-password ascii-text { <TEXT> | encrypted <ENCRYPTED-TEXT> }**	<TEXT> – строка [8..16] ASCII-символов; <ENCRYPTED-TEXT> – зашифрованный пароль, размером [8..16] байт, задаётся строкой [16..32] символов.
5	Задать имя класса объектов, среди которых необходимо выполнять поиск пользователей на LDAP-сервере (необязательно).	**esr(config)# ldap-server search filter user-object-class <NAME>**	<NAME> – имя класса объектов, задаётся строкой до 127 символов. Значение по умолчанию: posixAccount.
6	Задать область поиска пользователей в дереве LDAP-сервера (необязательно).	**esr(config)# ldap-server search scope <SCOPE>**	<SCOPE> – область поиска пользователей на LDAP-сервере, принимает следующие значения: • onelevel – выполнять поиск в объектах на следующем уровне после базового DN в дереве LDAP-сервера; • subtree – выполнять поиск во всех объектах поддерева базового DN в дереве LDAP-сервера. Значение по умолчанию: subtree.

Шаг	Описание	Команда	Ключи
7	Задать интервал, по истечении которого устройство считает, что LDAP-сервер не нашел записей пользователей, подходящих под условие поиска (необязательно).	**esr(config)# ldap-server search timeout <SEC>**	<SEC> – период времени в секундах, принимает значения [0..30] Значение по умолчанию: 0 – устройство ожидает завершения поиска и получения ответа от LDAP-сервера.
8	Задать имя атрибута объекта, со значением которого идет сравнение имени искомого пользователя на LDAP-сервере (необязательно).	**esr(config)# ldap-server naming-attribute <NAME>**	<NAME> – имя атрибута объекта, задаётся строкой до 127 символов. Значение по умолчанию: uid.
9	Задать имя атрибута объекта, значение которого будет определять начальные привилегии пользователя на устройстве (необязательно).	**esr(config)# ldap-server privilege-level-attribute <NAME>**	<NAME> – имя атрибута объекта, задаётся строкой до 127 символов. Значение по умолчанию: priv-lvl.
10	Задать глобальное значение кода DSCP для использования в IP-заголовках исходящих пакетов LDAP-сервера (необязательно).	**esr(config)# ldap-server dscp <DSCP>**	<DSCP> – значение кода DSCP, принимает значения в диапазоне [0..63]. Значение по умолчанию: 63.
11	Добавить LDAP-сервер в список используемых серверов и перейти в режим его конфигурирования.	**esr(config)# ldap-server host { <IP-ADDR> | <IPV6-ADDR> } [vrf <VRF>]**	<IP-ADDR> – IP-адрес LDAP-сервера, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255] <IPV6-ADDR> – IPv6-адрес LDAP-сервера, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF] <VRF> – имя экземпляра VRF, задается строкой до 31 символа.
12	Задать описание конфигурируемого LDAP-сервера (необязательно).	**esr(config-ldap-server)# description <description>**	<description> – описание LDAP-сервера, задается строкой до 255 символов.

Шаг	Описание	Команда	Ключи
13	Указать количество неудачных попыток аутентификации для блокировки логина пользователя и время блокировки (необязательно).	**esr(config-ldap-server)# aaa authentication attempts max-fail <COUNT> <TIME>**	<COUNT> – количество неудачных попыток аутентификации, после которых произойдет блокировка пользователя, принимает значения [1..65535]; <TIME> – интервал времени в минутах, на который будет заблокирован пользователь, принимает значения [1..65535]. Значение по умолчанию: <COUNT> – 5; <TIME> – 300.
14	Задать номер порта для обмена данными с удаленным LDAP-сервером (необязательно).	**esr(config-ldap-server)# port <PORT>**	<PORT> – номер TCP-порта для обмена данными с удаленным сервером, принимает значения [1..65535]. Значение по умолчанию: 389 для LDAP-сервера.
15	Задать приоритет использования удаленного LDAP-сервера (необязательно).	**esr(config-ldap-server)# priority <PRIORITY>**	<PRIORITY> – приоритет использования удаленного сервера, принимает значения [1..65535]. Чем ниже значение, тем приоритетнее сервер. Значение по умолчанию: 1.
16	Задать IPv4/IPv6-адрес, который будет использоваться в качестве IP/IPv6-адреса источника в отправляемых LDAP-пакетах.	**esr(config-ldap-server)# source-address { <ADDR> | object-group <NETWORK_OBJ_GROUP_NAME> }**	<ADDR> – IP-адрес источника, задается в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <NETWORK_OBJ_GROUP_NAME> – список адресов, которые будут использоваться в качестве source address.

Шаг	Описание	Команда	Ключи
17	Задать интерфейс или туннель маршрутизатора, IPv4/IPv6-адрес которого будет использоваться в качестве IPv4/IPv6-адреса источника в отправляемых LDAP-пакетах.	**esr(config-ldap-server)# source-interface { <IF> | <TUN> }**	<IF> – имя интерфейса устройства, задаётся в виде, описанном в разделе Типы и порядок именования интерфейсов маршрутизатора; <TUN> – имя туннеля устройства, задаётся в виде, описанном в разделе Типы и порядок именования туннелей маршрутизатора.
18	Задать список методов аутентификации по умолчанию (default)/с именем <NAME> и указать ldap.	**esr(config)# aaa authentication login { default | <NAME> } <METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка, задаётся строкой до 31 символа. Способы аутентификации: • local – аутентификация с помощью локальной базы пользователей; • tacacs – аутентификация по списку TACACS-серверов; • radius – аутентификация по списку RADIUS-серверов; • ldap – аутентификация по списку LDAP-серверов.
19	Задать список методов аутентификации повышения привилегий пользователей по умолчанию (default)/с именем <NAME> и указать ldap.	**esr(config)# aaa authentication enable <NAME> <METHOD 1> [<METHOD 2>] [<METHOD 3>] [<METHOD 4>]**	<NAME> – имя списка строка до 31 символа; • default – имя списка по умолчанию. <METHOD> – способы аутентификации: • enable – аутентификация с помощью enable-паролей; • tacacs – аутентификация по протоколу TACACS; • radius – аутентификация по протоколу RADIUS; • ldap – аутентификация по протоколу LDAP.

Шаг	Описание	Команда	Ключи
20	Указать способ перебора методов аутентификации в случае отказа.	**esr(config)# aaa authentication mode <MODE>**	<MODE> – способы перебора методов: • chain – если сервер вернул FAIL, переход к следующему в цепочке методу аутентификации; • break – если сервер вернул FAIL, прекратить попытки аутентификации. Если сервер недоступен, продолжить попытки аутентификации следующими в цепочке методами. Значение по умолчанию: chain.
21	Перейти в режим конфигурирования соответствующего терминала.	**esr(config)# line <TYPE>**	<TYPE> – тип консоли: • console – локальная консоль; • telnet – удаленная консоль; • ssh – защищенная удаленная консоль.
22	Активировать список аутентификации входа пользователей в систему.	**esr(config-line-console)# login authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 17.
23	Активировать список аутентификации повышения привилегий пользователей.	**esr(config-line-console)# enable authentication <NAME>**	<NAME> – имя списка, задаётся строкой до 31 символа. Создан на шаге 18.

16.1.5 Пример настройки аутентификации по Telnet через RADIUS-сервер

Задача:

Настроить аутентификацию пользователей, подключающихся по Telnet, через RADIUS (192.168.16.1/24).

Решение:

Настроим подключение к RADIUS-серверу и укажем ключ (password):

```
esr# configure
esr(config)# radius-server host 192.168.16.1
esr(config-radius-server)# key ascii-text encrypted 8CB5107EA7005AFF
esr(config-radius-server)# exit
```

Создадим профиль аутентификации:

```
esr(config)# aaa authentication login log radius
```

Укажем режим аутентификации, используемый при подключении по Telnet-протоколу:

```
esr(config)# line telnet
esr(config-line-telnet)# login authentication log
esr(config-line-telnet)# exit
esr(config)# exit
```

Просмотреть информацию по настройкам подключения к RADIUS-серверу можно командой:

```
esr# show aaa radius-servers
```

Посмотреть профили аутентификации можно командой:

```
esr# show aaa authentication
```

16.2 Настройка привилегий команд

Настройка привилегий команд является гибким инструментом, который позволяет назначить набору команд минимально необходимый уровень пользовательских привилегий (1-15). В дальнейшем при создании пользователя можно задать уровень привилегий, определяя ему доступный набор команд.

- *1-9 уровни* – позволяют использовать все команды мониторинга (show …);
- *10-14 уровни* – позволяют использовать все команды кроме команд перезагрузки устройства, управления пользователями и ряда других;
- *15 уровень* – позволяет использовать все команды.

16.2.1 Алгоритм настройки

Для изменения минимального уровня привилегий необходимого для выполнения команды CLI используется команда:

```
esr(config)# privilege <COMMAND-MODE> level <PRIV><COMMAND>
```

<COMMAND-MODE> – командный режим;

<PRIV> – необходимый уровень привилегий поддерева команд, принимает значение [1..15];

<COMMAND> – поддерево команд, задается строкой до 255 символов.

16.2.2 Пример настройки привилегий команд

Задача:

Перевести все команды просмотра информации об интерфейсах на уровень привилегий 10, кроме команды **show interfaces bridges**. Команду **show interfaces bridges** перевести на уровень привилегий 3.

Решение:

В режиме конфигурирования определим команды, разрешенные на использование с уровнем привилегий 10 и уровнем привилегий 3:

```
esr(config)# privilege root level 3 "show interfaces bridge"
esr(config)# privilege root level 10 "show interfaces"
```

16.3 Настройка логирования и защиты от ARP-spoofing атак

16.3.1 Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Включить защиту от ARP-spoofing атак на сетевом интерфейсе/сетевом мосту.	**esr(config-if-gi)# ip arp inspection**	
2	Включить логирование отброшенных ARP-ответов (необязательно).	**esr(config)# logging firewall arp-inspection**	

16.3.2 Пример настройки защиты от ARP-spoofing атак

Задача:

Необходимо защитить LAN-сеть и маршрутизатор ESR от ARP-spoofing атак и настроить логирование об атаках на удаленный syslog-сервер.

Решение:

Настраиваем интерфейсы и включем на них защиту от ARP-spoofing атак:

```
esr(config)# interface gigabitethernet 1/0/2
esr(config-if-gi)# ip address 10.0.0.1/24
esr(config-if-gi)# exit
esr(config)# interface gigabitethernet 1/0/1
esr(config-if-gi)# ip address 192.168.0.1/24
esr(config-if-gi)# ip arp inspection
esr(config-if-gi)# exit
```

Настраиваем параметры удаленного syslog-сервера:

```
esr(config)# syslog host SYSLOG_SERVER
esr(config-syslog-host)# remote-address 192.168.0.10
esr(config-syslog-host)# severity info
esr(config-syslog-host)# exit
```

Включаем логирование об ARP-spoofing атаках:

```
esr(config)# logging firewall arp-inspection
```

Просмотреть счетчики заблокированных ARP-ответов:

```
esr# show ip firewall arp-inspections
Interface              Count
--------------------   --------------------
gi1/0/1                9
```

Сбросить счетчики заблокированных ARP-ответов:

```
esr# clear ip firewall arp-inspections counters
```

Уведомление об ARP-spoofing атаке на удаленном syslog-сервере:

```
2025-11-24T12:29:01+07:00 esr firewalld %FIREWALL-W-EVENT_WARN: arp-inspection
a8:f9:4b:ac:4d:3e(192.168.0.12) -> 50:75:bd:04:fd:00(192.168.0.1) denied (gi1/0/1
 a8:f9:4b:ac:4d:3e)
```

16.4 Настройка логирования и защиты от сетевых атак

16.4.1 Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Включить защиту от ICMP flood-атак.	**esr(config)# ip firewall screen dos-defense icmp-threshold { <NUM> }**	<NUM> – количество ICMP-пакетов в секунду, задается в диапазоне [1..10000].

Шаг	Описание	Команда	Ключи
2	Включить защиту от land-атак.	**esr(config)# firewall screen dos-defense land**	
3	Включить ограничение числа пакетов, отправляемых за одну секунду на один адрес назначения	**esr(config)# ip firewall screen dos-defense limit-session-destination { <NUM> }**	<NUM> – ограничение числа IP-пакетов в секунду, задается в диапазоне [1..10000].
4	Включить ограничение числа пакетов, отправляемых за одну секунду с единого адреса источника	**esr(config)# ip firewall screen dos-defense limit-session-source { <NUM> }**	<NUM> – ограничение числа IP-пакетов в секунду, задается в диапазоне [1..10000].
5	Включить защиту от SYN flood-атак.	**esr(config)# ip firewall screen dos-defense syn-flood { <NUM> } [src-dsr]**	<NUM> – максимальное количество TCP-пакетов с установленным флагом SYN в секунду, задается в диапазоне [1..10000]. src-dst – ограничение количества TCP-пакетов с установленным флагом SYN на основании адреса источника и адреса назначения.
6	Включить защиту от UDP flood-атак.	**esr(config)# ip firewall screen dos-defense udp-threshold { <NUM> }**	<NUM> – максимальное количество UDP-пакетов в секунду, задается в диапазоне [1..10000].
7	Включить защиту от winnuke-атак.	**esr(config)# ip firewall screen dos-defense winnuke**	
8	Включить защиту от zero-port-атак	**esr(config)# ip firewall screen dos-defense zero-port**	
9	Включить блокировку TCP-пакетов с установленным флагом FIN и не установленным флагом ACK.	**esr(config)# ip firewall screen spy-blocking fin-no-ack**	
10	Включить блокировку ICMP-пакетов различных типов.	**esr(config)# ip firewall screen spy-blocking icmp-type**	<TYPE> – тип ICMP, может принимать значения: • destination-unreachable • echo-request • reserved • source-quench • time-exceeded

Шаг	Описание	Команда	Ключи
11	Включить защиту от IP sweep-атак.	**esr(config)# ip firewall screen spy-blocking ip-sweep { <NUM> }**	<NUM> – интервал выявления ip sweep атаки, задается в миллисекундах [1..1000000].
12	Включить защиту от port scan-атак.	**esr(config)# ip firewall screen spy-blocking port-scan { <threshold> } [<TIME>]**	<threshold> – интервал в секундах, в течение которого будет фиксироваться port scan-атака [1..10000]. <TIME> – время блокировки в миллисекундах [1..1000000].
13	Включить защиту от IP spoofing-атак.	**esr(config)# ip firewall screen spy-blocking spoofing**	
14	Исключить из защиты от IP-spoofing атак указанную Object Group.	**esr(config)# ip firewall screen spy-blocking spoofing exclude <object-group>**	<object-group> – список разрешённых для spoofing подсетей.
15	Включить блокировку TCP-пакетов, с установленными флагами SYN и FIN.	**esr(config)# ip firewall screen spy-blocking syn-fin**	
16	Включить блокировку TCP-пакетов, со всеми флагами или с набором флагов: FIN, PSH, URG. Данной командой обеспечивается защита от атаки XMAS.	**esr(config)# ip firewall screen spy-blocking tcp-all-flag**	
17	Включить блокировку TCP-пакетов, с нулевым полем flags.	**esr(config)# ip firewall screen spy-blocking tcp-no-flag**	
18	Включить блокировку фрагментированных ICMP-пакетов.	**esr(config)# ip firewall screen suspicious-packets icmp-fragment**	
19	Включить блокировку фрагментированных IP-пакетов.	**esr(config)# ip firewall screen suspicious-packets ip-fragment**	
20	Включить блокировку ICMP-пакетов длиной более 1024 байт.	**esr(config)# ip firewall screen suspicious-packets icmp-fragment**	
21	Включить блокировку фрагментированных TCP-пакетов, с флагом SYN.	**esr(config)# ip firewall screen suspicious-packets syn-fragment**	
22	Включить блокировку фрагментированных UDP-пакетов.	**esr(config)# ip firewall screen suspicious-packets udp-fragment**	

Шаг	Описание	Команда	Ключи
23	Включить блокировку пакетов, с ID протокола в заголовке IP равном 137 и более.	**esr(config)# ip firewall screen suspicious-packets unknown-protocols**	
24	Установить частоту оповещения (по SNMP, syslog и в CLI) об обнаруженных и отраженных сетевых атаках.	**esr(config)# ip firewall logging interval <NUM>**	<NUM> – интервал времени в секундах [30 .. 2147483647].
25	Включить механизм обнаружения и логирования DoS-атак через CLI, Syslog и по SNMP.	**esr(config)# logging firewall screen dos-defense <ATACK_TYPE>**	<ATACK_TYPE> – тип DoS-атаки, принимает значения: icmp-threshold, land, limit-session-destination, limit-session-source, syn-flood, udp-threshold, winnuke.
26	Включить механизм обнаружения и логирования шпионской активности через CLI, Syslog и по SNMP.	**esr(config)# logging firewall screen spy-blocking { <ATACK_TYPE> | icmp-type <ICMP_TYPE> }**	<ATACK_TYPE> – тип шпионской активности, принимает значения: fin-no-ack, ip-sweep, port-scan, spoofing, syn-fin, tcp-all-flag, tcp-no-flag. <ICMP_TYPE> – тип ICMP, принимает значения: destination-unreachable, echo-request, reserved, source-quench, time-exceeded.
27	Включить механизм обнаружения нестандартных пакетов и логирования через CLI, Syslog и по SNMP.	**esr(config)# logging firewall screen suspicious-packets <PACKET_TYPE>**	<PACKET_TYPE> – тип нестандартных пакетов, принимает значения: icmp-fragment, ip-fragment, large-icmp, syn-fragment, udp-fragment, unknown-protocols.

16.4.2 Описание механизмов защиты от атак

Команда	Описание
ip firewall screen dos-defense icmp-threshold	Данная команда включает защиту от ICMP flood-атак. При включенной защите ограничивается количество ICMP-пакетов всех типов в секунду для одного source-адреса. Атака приводит к перегрузке хоста и выводу его из строя из-за необходимости обрабатывать каждый запрос и отвечать на него.
firewall screen dos-defense land	Данная команда включает защиту от land-атак. При включенной защите блокируются пакеты с одинаковыми source и destination IP-адресами и флагом SYN в заголовке TCP. Атака приводит к перегрузке хоста и выводу его из строя из-за необходимости обрабатывать каждый TCP SYN пакет и попыток хоста установить TCP-сессию с самим собой.

Команда	Описание
ip firewall screen dos-defense limit-session-destination	Когда таблица IP-сессий хоста переполняется, он больше не в состоянии организовывать новые сессии и отбрасывает запросы (такое может происходить при различных DoS-атаках: SYN flood, UDP flood, ICMP flood, и т.д.). Команда включает ограничение числа пакетов, передаваемых за секунду на один адреса назначения, которое смягчает DoS-атаки.
ip firewall screen dos-defense limit-session-source	Когда таблица IP-сессий хоста переполняется, он больше не в состоянии организовывать новые сессии и отбрасывает запросы (такое может происходить при различных DoS-атаках: SYN flood, UDP flood, ICMP flood, и т.д.). Команда включает ограничение числа пакетов, передаваемых за секунду с одного адреса источника, которое смягчает DoS-атаки.
ip firewall screen dos-defense syn-flood	Данная команда включает защиту от SYN flood-атак. При включенной защите ограничивается количество TCP-пакетов с установленным флагом SYN в секунду для одного адреса назначения. Атака приводит к перегрузке хоста и выводу его из строя из-за необходимости обрабатывать каждый TCP SYN пакет и попыток установить TCP-сессии.
ip firewall screen dos-defense udp-threshold	Данная команда включает защиту от UDP flood-атак. При включенной защите ограничивается количество UDP-пакетов в секунду для одного адреса назначения. Атака приводит к перегрузке хоста и выводу его из строя из-за массивного UDP-трафика.
ip firewall screen dos-defense winnuke	Данная команда включает защиту от winnuke-атак. При включенной защите блокируются TCP-пакеты с установленным флагом URG и 139 портом назначения. Атака приводит к выходу из строя старых версий Windows (до 95 версии).
ip firewall screen dos-defense zero-port	Данная команда включает защиту от zero-port-атак. При включенной защите блокируются TCP/UDP-пакеты с установленным портом источника/назначения равным нулю.
ip firewall screen spy-blocking fin-no-ack	Данная команда включает блокировку TCP-пакетов с установленным флагом FIN и не установленным флагом ACK. Такие пакеты являются нестандартными, и по ответу можно определить операционную систему жертвы.
ip firewall screen spy-blocking icmp-type destination-unreachable	Данная команда включает блокировку всех ICMP-пакетов 3 типа (destination-unreachable), включая пакеты, сгенерированные самим маршрутизатором. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.
ip firewall screen spy-blocking icmp-type echo-request	Данная команда включает блокировку всех ICMP-пакетов 8 типа (echo-request), включая пакеты, сгенерированные самим маршрутизатором. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.
ip firewall screen spy-blocking icmp-type reserved	Данная команда включает блокировку всех ICMP-пакетов 2 и 7 типов (reserved), включая пакеты, сгенерированные самим маршрутизатором. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.

Команда	Описание
ip firewall screen spy-blocking icmp-type source-quench	Данная команда включает блокировку всех ICMP-пакетов 4 типа (source quench), включая пакеты, сгенерированные самим маршрутизатором. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.
ip firewall screen spy-blocking icmp-type time-exceeded	Данная команда включает блокировку всех ICMP-пакетов 11 типа (time exceeded), включая пакеты, сгенерированные самим маршрутизатором. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.
ip firewall screen spy-blocking ip-sweep	Данная команда включает защиту от IP sweep-атак. При включенной защите, если в течение заданного в параметрах интервала приходит более 10 ICMP-запросов от одного источника, первые 10 запросов пропускаются маршрутизатором, а 11 и последующие отбрасываются на оставшееся время интервала. Защита не дает злоумышленнику узнать о топологии сети и доступности хостов.
ip firewall screen spy-blocking port-scan	Данная команда включает защиту от port scan-атак. Если в течение первого заданного интервала времени (<threshold>) на один источник приходит более 10 TCP-пакетов с флагом SYN на разные TCP-порты или более 10 UDP-пакетов на разные UDP-порты, то такое поведение фиксируется как port scan-атака, и все последующие пакеты такого рода от источника блокируются на второй заданный интервал времени (<TIME>). Злоумышленник не сможет быстро просканировать открытые порты на устройстве.
ip firewall screen spy-blocking spoofing	Данная команда включает защиту от ip spoofing-атак. При включенной защите маршрутизатор проверяет пакеты на соответствие адреса источника и записей в таблице маршрутизации, и в случае несоответствия пакет отбрасывается. Например, если пакет с адресом источника 10.0.0.1/24 приходит на интерфейс Gi1/0/1, а в таблице маршрутизации данная подсеть располагается за интерфейсом Gi1/0/2, то считается, что адрес источника был подменен. Защищает от вторжений в сеть с подмененными source IP-адресами.
ip firewall screen spy-blocking spoofing exclude <object-group>	Данная команда исключает из защиты от IP-spoofing атак указанную Object Group. В Object Group помещается список из допустимых адресов, которые будут игнорироваться. Команда используется вместе с основной ip firewall screen spy-blocking spoofing, которая включает защиту, иначе она не будет иметь эффекта. В случае, если на маршрутизатор приходит spoofing от разрешённых подсетей (например, частый опрос устройств в сети), пакеты пропускаются.
ip firewall screen spy-blocking syn-fin	Данная команда включает блокировку TCP-пакетов с установленными флагами SYN и FIN. Такие пакеты являются нестандартными, и по ответу можно определить операционную систему жертвы.
ip firewall screen spy-blocking tcp-all-flag	Данная команда включает блокировку TCP-пакетов со всеми флагами или с набором флагов: FIN, PSH, URG. Обеспечивается защита от атаки XMAS.

Команда	Описание
ip firewall screen spy-blocking tcp-no-flag	Данная команда включает блокировку TCP-пакетов с нулевым полем flags. Такие пакеты являются нестандартными, и по ответу можно определить операционную систему жертвы.
ip firewall screen suspicious-packets icmp-fragment	Данная команда включает блокировку фрагментированных ICMP-пакетов. ICMP-пакеты обычно небольшого размера и необходимости в их фрагментировании нет.
ip firewall screen suspicious-packets ip-fragment	Данная команда включает блокировку фрагментированных пакетов.
ip firewall screen suspicious-packets large-icmp	Данная команда включает блокировку ICMP-пакетов длиной более 1024 байт.
ip firewall screen suspicious-packets syn-fragment	Данная команда включает блокировку фрагментированных TCP-пакетов с флагом SYN. TCP-пакеты с SYN-флагом обычно небольшого размера и необходимости в их фрагментировании нет. Защита предотвращает накопление фрагментированных пакетов в буфере.
ip firewall screen suspicious-packets udp-fragment	Данная команда включает блокировку фрагментированных UDP-пакетов.
ip firewall screen suspicious-packets unknown-protocols	Данная команда включает блокировку пакетов, с ID протокола в заголовке IP равном 137 и более.

16.4.3 Пример настройки логирования и защиты от сетевых атак

Задача:

Необходимо защитить LAN-сеть и маршрутизатор ESR от сетевых атак land, syn-flood, ICMP flood и настроить оповещение об атаках по SNMP на SNMP-сервер 192.168.0.10.

Решение:

Предварительно необходимо настроить интерфейсы и firewall (настройка firewall или ее отсутствие не повлияют на работу защиты от сетевых атак):

```
esr(config)# security zone LAN
esr(config-security-zone)# exit
esr(config)# security zone WAN
esr(config-security-zone)# exit
esr(config)# security zone-pair LAN WAN
esr(config-security-zone-pair)# rule 100
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
esr(config)# security zone-pair WAN LAN
esr(config-security-zone-pair)# rule 100
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
esr(config)# exit
esr(config)# interface gigabitethernet 1/0/1
esr(config-if-gi)# security-zone LAN
esr(config-if-gi)# ip address 192.168.0.1/24
esr(config-if-gi)# exit
esr(config)# interface gigabitethernet 1/0/2
esr(config-if-gi)# security-zone WAN
esr(config-if-gi)# ip address 10.0.0.1/24
esr(config-if-gi)# exit
```

Настроим защиту от land, syn-flood, ICMP flood-атак:

```
esr(config)# ip firewall screen dos-defense land
esr(config)# ip firewall screen dos-defense syn-flood 100 src-dst
esr(config)# ip firewall screen dos-defense icmp-threshold 100
```

Настроим логирование обнаруженных атак:

```
esr(config)# logging firewall screen dos-defense land
esr(config)# logging firewall screen dos-defense syn-flood
esr(config)# logging firewall screen dos-defense icmp-threshold
```

Настроим SNMP-сервер, на который будут отправляться трапы:

```
esr(config)# snmp-server
esr(config)# snmp-server host 192.168.0.10
esr(config)# snmp-server enable traps screen land
esr(config)# snmp-server enable traps screen syn-flood
esr(config)# snmp-server enable traps screen icmp-threshold
```

Посмотреть статистику по зафиксированным сетевым атакам можно командой:

```
esr# show ip firewall screen counters
```

Очистить статистику по зафиксированным сетевым атакам можно командой:

```
esr# clear ip firewall screen counters
```

16.5 Конфигурирование Firewall

Firewall – комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

16.5.1 Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Создать зоны безопасности.	**esr(config)# security zone <zone-name1>** **esr(config)# security zone <zone-name2>**	<zone-name> – до 12 символов. Имена all, any и self зарезервированы.
2	Задать описание зоны безопасности.	**esr(config-security-zone)# description <description>**	<description> – до 255 символов.
3	Указать экземпляр VRF, в котором будет работать данная зона безопасности (необязательно).	**esr(config- security-zone)# ip vrf forwarding <VRF>**	<VRF> – имя VRF, задается строкой до 31 символа.
4	Включить счетчики сессий для NAT и Firewall (необязательно, снижает производительность).	**esr(config)# ip firewall sessions counters**	
5	Отключить фильтрацию пакетов, для которых не удалось определить принадлежность к какому-либо известному соединению и которые не являются началом нового соединения (необязательно, снижает производительность).	**esr(config)# ip firewall sessions unknown <ACTION>**	<ACTION> – правило обработки неизвестных сессий для межсетевого экрана: permit – разрешить неизвестные сессии; deny – отбрасывать неизвестные сессии; reject – отбрасывать неизвестные сессии и отправлять обратно пакет с ошибкой.

Шаг	Описание	Команда	Ключи
6	Выбрать режим работы межсетевого экрана (необязательно). В режиме stateful проверяется только первый пакет сессии, и если «прямой» трафик разрешён, «ответный» трафик разрешается автоматически. В режиме stateless происходит проверка каждого пакета. «Прямой» и «ответный» трафики требуется разрешать в соответствующих zone-pair (см. шаг 29). Работа межсетевого экрана по списку приложений возможна только в режиме stateless.	**esr(config)# ip firewall mode <MODE>**	<MODE> – режим работы межсетевого экрана, может принимать значения: stateful, stateless. Значение по умолчанию: stateful.
7	Определить время жизни сессии для неподдерживаемых протоколов (необязательно).	**esr(config)# ip firewall sessions generic-timeout <TIME>**	<TIME> – время жизни сессии для неподдерживаемых протоколов, принимает значения в секундах [1..8553600]. По умолчанию: 60 секунд.
8	Определить время жизни ICMP-сессии, по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions icmp-timeout <TIME>**	<TIME> – время жизни ICMP-сессии, принимает значения в секундах [1..8553600]. По умолчанию: 30 секунд.
9	Определить время жизни ICMPv6-сессии, по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions icmpv6-timeout <TIME>**	<TIME> – время жизни ICMP-сессии, принимает значения в секундах [1..8553600]. По умолчанию: 30 секунд.
10	Определить размер таблицы сессий, ожидающих обработки (необязательно).	**esr(config)# ip firewall sessions max-expect <COUNT>**	<COUNT> – размер таблицы, принимает значения [1..8553600]. По умолчанию: 256.
11	Определить размер таблицы отслеживаемых сессий (необязательно).	**esr(config)# ip firewall sessions max-tracking <COUNT>**	<COUNT> – размер таблицы, принимает значения [1..8553600]. По умолчанию: 512000.

Шаг	Описание	Команда	Ключи
12	Определить время жизни TCP-сессии в состоянии «соединение устанавливается», по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions tcp-connect-timeout <TIME>**	<TIME> – время жизни TCP-сессии в состоянии «соединение устанавливается», принимает значения в секундах [1..8553600]. По умолчанию: 60 секунд.
13	Определить время жизни TCP-сессии в состоянии «соединение закрывается», по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions tcp-disconnect-timeout <TIME>**	<TIME> – время жизни TCP-сессии в состоянии «соединение закрывается» принимает значения в секундах [1..8553600]. По умолчанию: 30 секунд.
14	Определить время жизни TCP-сессии в состоянии «соединение установлено», по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions tcp-established-timeout <TIME>**	<TIME> – время жизни TCP-сессии в состоянии «соединение установлено», принимает значения в секундах [1..8553600]. По умолчанию: 120 секунд.
15	Определить время ожидания, по истечении которого происходит фактическое удаление закрытой TCP-сессии из таблицы отслеживаемых сессий (необязательно).	**esr(config)# ip firewall sessions tcp-latecome-timeout <TIME>**	<TIME> – время ожидания, принимает значения в секундах [1..8553600]. По умолчанию: 120 секунд.

Шаг	Описание	Команда	Ключи
16	Включить функцию отслеживания сессий уровня приложений для отдельных протоколов (необязательно).	**esr(config)# ip firewall sessions tracking**	<PROTOCOL> – протокол уровня приложений [ftp, h323, pptp, netbios-ns, tftp], сессии которого должны отслеживаться. <OBJECT-GROUP-SERVICE> – имя профиля TCP/UDP-портов sip-сессии, задаётся строкой до 31 символа. Если группа не указана, то отслеживание сессий sip будет осуществляться для порта 5060. Вместо имени отдельного протокола можно использовать ключ «all», который включает функцию отслеживания сессий уровня приложений для всех доступных протоколов. По умолчанию – отключено для всех протоколов.
17	Определить время жизни UDP-сессии в состоянии «соединение подтверждено», по истечении которого она считается устаревшей (необязательно).	**esr(config)# ip firewall sessions udp-assured-timeout <TIME>**	<TIME> – время жизни UDP-сессии в состоянии «соединение подтверждено», принимает значения в секундах [1..8553600]. По умолчанию: 180 секунд.
18	Определить время жизни UDP-сессии в состоянии «соединение не подтверждено», по истечении которого она считается устаревшей.	**esr(config)# ip firewall sessions udp-wait-timeout <TIME>**	<TIME> – время жизни UDP-сессии в состоянии «соединение не подтверждено», принимает значения в секундах [1..8553600]. По умолчанию: 30 секунд.
19	Создать списки MAC-адресов, которые будут использоваться при фильтрации.	**esr(config)# object-group mac <obj-group-name>**	<obj-group-name> – до 31 символа.
20	Задать описание списка MAC-адресов (необязательно).	**esr(config-object-group-mac)# description <description>**	<description> – описание профиля, задается строкой до 255 символов.

Шаг	Описание	Команда	Ключи
21	Внести необходимые MAC-адреса в список.	**esr(config-object-group-mac)# mac address <ADDR> <WILDCARD>**	<ADDR> – MAC-адрес, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]. <WILDCARD> – маска MAC-адреса, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]. Биты маски, установленные в 0, задают биты MAC-адреса, исключаемые из сравнения при поиске.
22	Создать списки IP-адресов, которые будут использоваться при фильтрации.	**esr(config)# object-group network <obj-group-name>**	<obj-group-name> – до 31 символа.
23	Задать описание списка IP-адресов (необязательно).	**esr(config-object-group-network)# description <description>**	<description> – описание профиля, задается строкой до 255 символов.
24	Внести необходимые IPv4/IPv6-адреса в список.	**esr(config-object-group-network)# ip prefix <ADDR/LEN> [unit <ID>]**	<ADDR/LEN> – подсеть, задаётся в виде AAA.BBB.CCC.DDD/EE, где каждая часть AAA – DDD принимает значения [0..255] и EE принимает значения [1..32]. <ID> – номер юнита, принимает значения [1..4].
		esr(config-object-group-network)# ip address-range <FROM-ADDR>-<TO-ADDR> [unit <ID>]	<FROM-ADDR> – начальный IP-адрес диапазона адресов; <TO-ADDR> – конечный IP-адрес диапазона адресов, опциональный параметр. Если параметр не указан, то командой задаётся одиночный IP-адрес. Адреса задаются в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]. <ID> – номер юнита, принимает значения [1..4].

Шаг	Описание	Команда	Ключи
		esr(config-object-group-network)# ipv6 prefix <IPV6-ADDR/LEN> [unit <ID>]	<IPV6-ADDR/LEN> – IP-адрес и маска подсети, задаётся в виде X:X:X:X::X/EE, где каждая часть X принимает значения в шестнадцатеричном формате [0..FFFF] и EE принимает значения [1..128]. <ID> – номер юнита, принимает значения [1..4].
		esr(config-object-group-network)# ipv6 address-range <FROM-ADDR>-<TO-ADDR> [unit <ID>]	<FROM-ADDR> – начальный IPv6-адрес диапазона адресов; <TO-ADDR> – конечный IPv6-адрес диапазона адресов, опциональный параметр. Если параметр не указан, то командой задаётся одиночный IPv6-адрес. Адреса задаются в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF]. <ID> – номер юнита, принимает значения [1..4].
25	Создать списки сервисов, которые будут использоваться при фильтрации.	**esr(config)# object-group service <obj-group-name>**	<obj-group-name> – имя профиля сервисов, задается строкой до 31 символа.
26	Задать описание списка сервисов (необязательно).	**esr(config-object-group-service)# description <description>**	<description> – описание профиля, задается строкой до 255 символов.
27	Внести необходимые сервисы (tcp/udp-порты) в список.	**esr(config-object-group-service)# port-range <port>**	<port> – принимает значение [1..65535]. Можно указать несколько портов перечислением через запятую «,» либо указать диапазон портов через «-».
28	Создать списки приложений, которые будут использоваться в механизме DPI.	**esr(config)# object-group application <NAME>**	<NAME> – имя профиля приложений, задается строкой до 31 символа.
29	Задать описание списка приложений (необязательно).	**esr(config-object-group-application)# description <description>**	<description> – описание профиля, задается строкой до 255 символов.

Шаг	Описание	Команда	Ключи
30	Внести необходимые приложения в списки.	**esr(config-object-group-application)# application < APPLICATION >**	<APPLICATION> – указывает приложение, попадающее под действие данного профиля.
31	Создать список доменных имен, которые будут использоваться при фильтрации.	**esr(config)# object-group domain-name <NAME>**	<NAME> – имя профиля доменных имен, задается строкой до 31 символа.
32	Задать описание списка доменных имен (необязательно).	**esr(config-object-group-domain-name)# description <description>**	<description> – описание профиля, задается строкой до 255 символов.
33	Внести необходимые доменные имена в списки.	**esr(config-object-group-domain-name)# domain <DOMAIN>**	<DOMAIN> – доменное имя, строка длинной от 1 до 253 символов.
34	Включить интерфейсы (физические, логические, E1/Multilink и подключаемые), сервер удаленного доступа (l2tp, openvpn, pptp) или туннели (gre, ip4ip4, l2tp, lt, pppoe, pptp) в зоны безопасности (если необходимо).	**esr(config-if-gi)# security-zone <zone-name>**	<zone-name> – до 12 символов.
	Отключить функции Firewall на сетевом интерфейсе (физические, логические, E1/Multilink и подключаемые), сервере удаленного доступа (l2tp, openvpn, pptp) или туннели (gre, ip4ip4, l2tp, lt, pppoe, pptp) (если необходимо).	**esr(config-if-gi)# ip firewall disable**	
	Отключить функции Firewall глобально на всех сетевых сущностях (если необходимо).	**esr(config)# ip firewall disable**	
35	Создать набор правил межзонового взаимодействия. На маршрутизаторе всегда существует зона безопасности с именем «self». Если в качестве получателя трафика выступает сам маршрутизатор, то есть трафик не является транзитным, то в качестве параметра указывается зона «self». Очерёдность обработки трафика для разных zone-pair описана в примечании.	**esr(config)# security zone-pair <src-zone-name1> <dst-zone-name2>**	<src-zone-name> – до 12 символов. <dst-zone-name> – до 12 символов.

Шаг	Описание	Команда	Ключи
36	Создать правило межзонового взаимодействия.	**esr(config-security-zone-pair)# rule <rule-number>**	<rule-number> – 1..10000.
37	Задать описание правила (необязательно).	**esr(config-security-zone-pair)# description <description>**	<description> – до 255 символов.
38	Указать действие данного правила.	**esr(config-security-zone-pair-rule)# action <action> [log]**	<action> – permit/deny/reject/ netflow-sample/sflow-sample/ rate-limit/session-limit Ключ session-limit применим только на моделях ESR-30/ESR-31/ ESR-3100/ESR-3150/ ESR-3200/ESR-3200L/ ESR-3250/ ESR-3300/ ESR-3350. log – ключ для активации логирования сессий, устанавливающимися согласно данному правилу.
39	Установить имя или номер IP-протокола, для которого должно срабатывать правило (необязательно).	**esr(config-security-zone-pair-rule)# match [not] protocol <protocol-type>**	<protocol-type> – тип протокола, принимает значения: esp, icmp, ah, eigrp, ospf, igmp, ipip, tcp, pim, udp, vrrp, rdp, l2tp, gre. При указании значения «any» правило будет срабатывать для любых протоколов.
		esr(config-security-zone-pair-rule)# match [not] protocol-id <protocol-id>	<protocol-id> – идентификационный номер IP-протокола, принимает значения [0x00-0xFF].

Шаг	Описание	Команда	Ключи
40	Установить IP-адрес отправителя, для которых должно срабатывать правило (необязательно).	**esr(config-security-zone-pair-rule)# match [not] source-address { address-range { <ADDR>[-<ADDR>] | <IPV6-ADDR>[-<IPV6-ADDR>] } | prefix { <ADDR/LEN> | <IPv6-ADDR/LEN> } | object-group { network <OBJ-GROUP-NETWORK-NAME> | domain-name <OBJ-GROUP-DOMAIN-NAME> } | any }**	**address-range** <ADDR>[-<ADDR>] – диапазон IP-адресов для правил firewall. Если не указывать IP-адрес конца диапазона, то в качестве IP-адреса для срабатывания правила используется только IP-адрес начала диапазона. Параметр задаётся в виде A.B.C.D, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF]; **prefix** <ADDR/LEN> – IP-подсеть, используемая для срабатывания правила фильтрации firewall. Параметр задаётся в виде A.B.C.D/E, где каждая часть A – D принимает значения [0..255] и E принимает значения [1..32]; <IPv6-ADDR/LEN> – IPv6-адрес, задаётся в виде X:X:X:X::X/E, где каждая часть X принимает значения в шестнадцатеричном формате [0..FFFF] и E принимает значения [1..128]; **object-group network** <OBJ-GROUP-NETWORK-NAME> – имя профиля IP-адресов, задаётся строкой до 31 символа; **object-group domain-name** <OBJ-GROUP-DOMAIN-NAME> – имя профиля доменных имен, задаётся строкой до 31 символа. При указании значения **any** правило будет срабатывать для любого IP-адреса получателя.
41	Установить IP-адрес получателя, для которых должно срабатывать правило (необязательно).	**esr(config-security-zone-pair-rule)# match [not] destination-address { address-range { <ADDR>[-<ADDR>] | <IPV6-ADDR>[-<IPV6-ADDR>] } | prefix { <ADDR/LEN> | <IPv6-ADDR/LEN> } | object-group { network <OBJ-GROUP-NETWORK-NAME> | domain-name <OBJ-GROUP-DOMAIN-NAME> } | any }**	

Шаг	Описание	Команда	Ключи
42	Установить MAC-адрес отправителя, для которого должно срабатывать правило (необязательно).	**esr(config-security-zone-pair-rule)# match [not] source-mac {<mac-addr> | <OBJ-GROUP-NAME>}**	<mac-addr> – задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]. <OBJ-GROUP-NAME> – имя профиля MAC-адресов, задаётся строкой до 31 символа.
43	Установить MAC-адрес получателя, для которого должно срабатывать правило (необязательно).	**esr(config-security-zone-pair-rule)# match [not] destination-mac {<mac-addr> | <OBJ-GROUP-NAME>}**	
44	Установить TCP/UDP-порт отправителя, для которого должно срабатывать правило (если указан протокол).	**esr(config-security-zone-pair-rule)# match [not] source-port <TYPE> {<PORT-SET-NAME> | <FROM-PORT> - <TO-PORT>}**	<TYPE> – тип аргумента, устанавливаемый в качестве порта: • **object-group** – указать имя профиля; • **port-range** – указать диапазон портов; • **any** – установить в качестве порта любой порт. <PORT-SET-NAME> – задаётся строкой до 31 символа; <FROM-PORT> – начальный порт диапазона; <TO-PORT> – конечный порт диапазона.
45	Установить TCP/UDP-порт получателя, для которого должно срабатывать правило (если указан протокол).	**esr(config-security-zone-pair-rule)# match [not] destination-port <TYPE> {<PORT-SET-NAME> | <FROM-PORT> - <TO-PORT>}**	
46	Установить профиль приложений, который будет использоваться в механизме DPI.	**esr(config-security-zone-pair-rule)# match [not] application <OBJ-GROUP-NAME>**	<OBJ-GROUP-NAME> – имя профиля приложений, задаётся строкой до 31 символа.
47	Установить тип и код сообщений протокола ICMP, для которых должно срабатывать правило (если в качестве протокола выбран ICMP) (необязательно).	**esr(config-security-zone-pair-rule)# match [not] icmp <ICMP_TYPE> <ICMP_CODE>**	<ICMP_TYPE> – тип сообщения протокола ICMP, принимает значения [0..255]; <ICMP_CODE> – код сообщения протокола ICMP, принимает значения [0..255]. При указании значения «any» правило будет срабатывать для любого кода сообщения протокола ICMP.

Шаг	Описание	Команда	Ключи
48	Установить тип и код сообщений протокола ICMPv6, для которых должно срабатывать правило (если в качестве протокола выбран ICMP) (необязательно).	**esr(config-security-zone-pair-rule)# match [not] icmpv6 <ICMP_TYPE> <ICMP_CODE>**	<ICMP_TYPE> – тип сообщения протокола ICMPv6, принимает значения [0..255]; <ICMP_CODE> – код сообщения протокола ICMPv6, принимает значения [0..255]. При указании значения «any» правило будет срабатывать для любого кода сообщения протокола ICMP.
49	Установить ограничение, при котором правило будет срабатывать только для трафика, измененного сервисом трансляции IP-адресов и портов получателя.	**esr(config-security-zone-pair-rule)# match [not] destination-nat**	
50	Установить фильтрацию только для фрагментированных IP-пакетов (необязательно, доступно только для zone-pair any self и zone-pair <zone-name> any).	**esr(config-security-zone-pair-rule)# match [not] fragment**	
51	Установить фильтрацию для IP-пакетов, содержащих ip-option (необязательно, доступно только для zone-pair any self и zone-pair <zone-name> any).	**esr(config-security-zone-pair-rule)# match [not] ip-option**	
52	Включить правило межзонового взаимодействия.	**esr(config-security-zone-pair-rule)# enable**	
53	Активировать фильтрацию и режим отслеживания сессий при прохождении пакетов между участниками одной Bridge-группы (необязательно).	**esr(config-bridge)# ports firewall enable**	
54	Добавить работу правила безопасности по расписанию.	**esr(config-security-zone-pair-rule)# scheduler <SCHEDULER>**	<SCHEDULER> – название существующего в конфигурации расписания.

Порядок обработки транзитного трафика правилами firewall

1. Трафик проверяется правилами zone-pair user-zone any.
   Если трафик не попал ни под одно из правил текущей zone-pair, переходим к следующему шагу.
   Разрешение трафика правилом в этой zone-pair не пропускает проверки следующих zone-pair.

2. Если трафик передаётся с одного интерфейса на другой в пределах одной зоны (user-zone), то он проверяется правилами zone-pair user-zone user-zone.
   Если трафик не попал ни под одно из правил текущей zone-pair, переходим к следующему шагу.
3. Если трафик передаётся с одного интерфейса на другой в разных зонах, то он проверяется правилами zone-pair user-zone1 user-zone2.
   Если трафик не попал ни под одно из правил текущей zone-pair, переходим к следующему шагу.
4. Трафик проверяется правилами zone-pair any any.
   Если трафик не попал ни под одно из правил текущей zone-pair, он отбрасывается.

Порядок обработки трафика, терминируемого на маршрутизаторе

1. Трафик проверяется правилами zone-pair any self.
   Если трафик не попал ни под одно из правил текущей zone-pair, переходим к следующему шагу.
   Разрешение трафика правилом в этой zone-pair не пропускает проверки следующих zone-pair.
2. Трафик проверяется правилами zone-pair user-zone self.
   Если трафик не попал ни под одно из правил текущей zone-pair, переходим к следующему шагу.
3. Трафик проверяется правилами zone-pair any any.
   Если трафик не попал ни под одно из правил текущей zone-pair, он отбрасывается.

Каждая команда «match» может содержать ключ «not». При использовании данного ключа под правило будут подпадать пакеты, не удовлетворяющие заданному критерию.

Более подробная информация о командах для настройки межсетевого экрана содержится в справочнике команд CLI.

16.5.2 Пример настройки Firewall

Задача:

Разрешить обмен сообщениями по протоколу ICMP между устройствами R1, R2 и маршрутизатором ESR.

Решение:

Для каждой сети ESR создадим свою зону безопасности:

```
esr# configure
esr(config)# security zone LAN
esr(config-security-zone)# exit
esr(config)# security zone WAN
esr(config-security-zone)# exit
```

Настроим сетевые интерфейсы и определим их принадлежность к зонам безопасности:

```
esr(config)# interface gi1/0/2
esr(config-if-gi)# ip address 192.168.12.2/24
esr(config-if-gi)# security-zone LAN
esr(config-if-gi)# exit
esr(config)# interface gi1/0/3
esr(config-if-gi)# ip address 192.168.23.2/24
esr(config-if-gi)# security-zone WAN
esr(config-if-gi)# exit
```

Для настройки правил зон безопасности потребуется создать профиль адресов сети «LAN», включающий адреса, которым разрешен выход в сеть «WAN», и профиль адресов сети «WAN».

```
esr(config)# object-group network WAN
esr(config-object-group-network)# ip address-range 192.168.23.2
esr(config-object-group-network)# exit
esr(config)# object-group network LAN
esr(config-object-group-network)# ip address-range 192.168.12.2
esr(config-object-group-network)# exit
esr(config)# object-group network LAN_GATEWAY
esr(config-object-group-network)# ip address-range 192.168.12.1
esr(config-object-group-network)# exit
esr(config)# object-group network WAN_GATEWAY
esr(config-object-group-network)# ip address-range 192.168.23.3
esr(config-object-group-network)# exit
```

Для пропуска трафика из зоны «LAN» в зону «WAN» создадим пару зон и добавим правило, разрешающее проходить ICMP-трафику от R1 к R2. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair LAN WAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# match protocol icmp
esr(config-security-zone-pair-rule)# match destination-address object-group network WAN_GATEWAY
esr(config-security-zone-pair-rule)# match source-address object-group network LAN_GATEWAY
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Для пропуска трафика из зоны «WAN» в зону «LAN» создадим пару зон и добавим правило, разрешающее проходить ICMP-трафику от R2 к R1. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair WAN LAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# match protocol icmp
esr(config-security-zone-pair-rule)# match destination-address object-group network LAN_GATEWAY
esr(config-security-zone-pair-rule)# match source-address object-group network WAN_GATEWAY
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

На маршрутизаторе всегда существует зона безопасности с именем «self». Если в качестве получателя трафика выступает сам маршрутизатор, то есть трафик не является транзитным, то в качестве

параметра указывается зона «self». Создадим пару зон для трафика, идущего из зоны «WAN» в зону «self». Добавим правило, разрешающее проходить ICMP-трафику между R2 и маршрутизатором ESR, для того чтобы маршрутизатор начал отвечать на ICMP-запросы из зоны «WAN»:

```
esr(config)# security zone-pair WAN self
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# match protocol icmp
esr(config-security-zone-pair-rule)# match destination-address object-group network WAN
esr(config-security-zone-pair-rule)# match source-address object-group network WAN_GATEWAY
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Создадим пару зон для трафика, идущего из зоны «LAN» в зону «self». Добавим правило, разрешающее проходить ICMP-трафику между R1 и ESR, для того чтобы маршрутизатор начал отвечать на ICMP-запросы из зоны «LAN»:

```
esr(config)# security zone-pair LAN self
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# match protocol icmp
esr(config-security-zone-pair-rule)# match destination-address network object-group LAN
esr(config-security-zone-pair-rule)# match source-address object-group network LAN_GATEWAY
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
esr(config)# exit
```

Посмотреть членство портов в зонах можно с помощью команды:

```
esr# show security zone
```

Посмотреть пары зон и их конфигурацию можно с помощью команд:

```
esr# show security zone-pair
esr# show security zone-pair configuration
```

Посмотреть активные сессии можно с помощью команд:

```
esr# show ip firewall sessions
```

16.5.3 Пример настройки Firewall по доменным именам

Задача:

Фиксировать обращения на ресурсы компании Yandex из локальной сети в Syslog:

Решение:

Создадим зоны безопасности для локальной сети и uplink-интерфейса в сторону интернет-провайдера:

```
esr# configure
esr(config)# security zone LAN
esr(config-security-zone)# exit
esr(config)# security zone WAN
esr(config-security-zone)# exit
```

Настроим сетевые интерфейсы и определим их принадлежность к зонам безопасности:

```
esr(config)# interface gi1/0/1
esr(config-if-gi)# ip address 10.0.0.1/24
esr(config-if-gi)# security-zone WAN
esr(config-if-gi)# exit
esr(config)# interface gi1/0/2
esr(config-if-te)# ip address 192.168.0.1/24
esr(config-if-te)# security-zone LAN
esr(config-if-te)# exit
```

Настроим маршрут по умолчанию в сторону интернет-провайдера:

```
esr(config)# ip route 0.0.0.0/0 10.0.0.254
```

Настроим систему разрешения доменных имен, её настройка необходима для работы Firewall по доменным именам:

```
esr(config)# domain lookup enable
esr(config)# domain nameserver 10.0.0.254
```

Настроим простую конфигурацию Source-NAT для любого транзитного через ESR трафика:

```
esr(config)# nat source
esr(config-snat)# ruleset SNAT
esr(config-snat-ruleset)# to zone WAN
esr(config-snat-ruleset)# rule 1
esr(config-snat-rule)# action source-nat interface
esr(config-snat-rule)# enable
esr(config-snat-rule)# exit
esr(config-snat-ruleset)# exit
esr(config-snat)# exit
```

Создадим профиль доменных имен, включающий в себя популярные домены компании "Яндекс":

> ESR также позволяет работать с интернационализованными доменными именами (домены, использующие символы национальных алфавитов). Для использования интернационализованного доменного имени в конфигурации ESR его нужно преобразовать к виду ASCII-compatible encoding при помощи любого Punycode-преобразователя. Т. е. получим:
> яндекс.рф → xn--d1acpjx3f.xn–p1ai
> президент.рф → xn--d1abbgf6aiiy.xn–p1ai
> И уже ASCII-compatible encoding вариант доменного имени можно указывать в конфигурации ESR.

```
esr(config)# object-group domain-name YANDEX
esr(config-object-group-domain-name)# domain ya.ru
esr(config-object-group-domain-name)# domain yandex.ru
esr(config-object-group-domain-name)# domain dzen.ru
esr(config-object-group-domain-name)# domain xn--d1acpjx3f.xn--p1ai
esr(config-object-group-domain-name)# exit
```

Разрешим прохождение трафика из зоны «LAN» в зону «WAN» и отдельно создадим правило, которое будет фиксировать в Syslog обращения на домены компании "Яндекс":

```
esr(config)# security zone-pair LAN WAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action permit log
esr(config-security-zone-pair-rule)# match destination-address object-group domain-name YANDEX
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# rule 2
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Посмотреть членство портов в зонах можно с помощью команды:

```
esr# show security zone
```

Посмотреть пары зон и их конфигурацию можно с помощью команд:

```
esr# show security zone-pair
esr# show security zone-pair configuration
```

Посмотреть активные сессии можно с помощью команд:

```
esr# show ip firewall sessions
```

Посмотреть кэш DNS можно с помощью команд:

```
esr# show dns records
esr# show dns records negative
```

Пример сообщения Syslog при срабатывании правила:

```
<190>1 2025-07-07T17:40:10+07:00 esr firewalld - - - %FIREWALL-I-LOG: zone-pair 'LAN WAN' rule
1 permitted tcp 192.168.0.21:45574 (gi1/0/2) -> 77.88.44.55:443 dscp 48
```

16.5.4 Пример настройки фильтрации приложений (DPI)

⚠ Использование механизма фильтрации приложений многократно снижает производительность маршрутизатора из-за необходимости проверки определенного объема пакетов в каждой сессии. Производительность снижается с ростом количества выбранных приложений для фильтрации. Механизм фильтрации приложений работает только для транзитных пакетов и только в режиме черного списка (запрет прохождения трафика для указанных приложений).

Задача:

Блокировать доступ пользователей в локальной сети к Telegram, Facebook Messenger и Skype.

Решение:

Для каждой сети ESR создадим свою зону безопасности:

```
esr# configure
esr(config)# security zone LAN
esr(config-security-zone)# exit
esr(config)# security zone WAN
esr(config-security-zone)# exit
```

Настроим сетевые интерфейсы и определим их принадлежность к зонам безопасности:

```
esr(config)# interface gi1/0/1
esr(config-if-gi)# ip address 10.0.0.1/24
esr(config-if-gi)# security-zone WAN
esr(config-if-gi)# exit
esr(config)# interface gi1/0/2
esr(config-if-te)# ip address 192.168.0.1/24
esr(config-if-te)# security-zone LAN
esr(config-if-te)# exit
```

Для настройки правил зон безопасности потребуется создать профиль приложений, которые необходимо будет блокировать:

```
esr(config)# object-group application BLACKLIST
esr(config-object-group-application)# application telegram
esr(config-object-group-application)# application facebook-messenger
esr(config-object-group-application)# application skype-teams
esr(config-object-group-application)# exit
```

Для установки правил прохождения трафика из зоны «WAN» в зону «LAN» создадим пару зон и добавим правило, запрещающее проходить трафику приложений, и правило, разрешающее проходить остальному трафику. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair WAN LAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action deny
esr(config-security-zone-pair-rule)# match application BLACKLIST
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# rule 2
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Для установки правил прохождения трафика из зоны «LAN» в зону «WAN» создадим пару зон и добавим правило, запрещающее прохождение трафика приложений, и правило, разрешающее прохождение всего остального трафика. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair LAN WAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action deny
esr(config-security-zone-pair-rule)# match application BLACKLIST
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# rule 2
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Посмотреть членство портов в зонах можно с помощью команды:

```
esr# show security zone
```

Посмотреть пары зон и их конфигурацию можно с помощью команд:

```
esr# show security zone-pair
esr# show security zone-pair configuration
```

Посмотреть активные сессии можно с помощью команд:

```
esr# show ip firewall sessions
```

16.5.5 Пример настройки Firewall по расписанию

> ⚠ Использование механизма фильтрации приложений многократно снижает производительность маршрутизатора из-за необходимости проверки определенного объема пакетов в каждой сессии. Производительность снижается с ростом количества выбранных приложений для фильтрации. Механизм фильтрации приложений работает только для транзитных пакетов и только в режиме черного списка (запрет прохождения трафика для указанных приложений).

Задача:

Блокировать доступ пользователей из локальной сети к YouTube в рабочее время, начиная с 1 января 2026 года.

Решение:

Создадим зоны безопасности для локальной сети и uplink-интерфейса в сторону интернет-провайдера:

```
esr# configure
esr(config)# security zone LAN
esr(config-security-zone)# exit
esr(config)# security zone WAN
esr(config-security-zone)# exit
```

Настроим сетевые интерфейсы и определим их принадлежность к зонам безопасности:

```
esr(config)# interface gi1/0/1
esr(config-if-gi)# ip address 10.0.0.1/24
esr(config-if-gi)# security-zone WAN
esr(config-if-gi)# exit
esr(config)# interface gi1/0/2
esr(config-if-te)# ip address 192.168.0.1/24
esr(config-if-te)# security-zone LAN
esr(config-if-te)# exit
```

Настроим маршрут по умолчанию в сторону интернет-провайдера:

```
esr(config)# ip route 0.0.0.0/0 10.0.0.254
```

Создадим расписание, по которому будет активироваться правило.

```
esr(config)# scheduler WORK_HOURS
esr(config-scheduler)# periodic weekdays
esr(config-scheduler-weekdays)# start-time 08:00:00 stop-time 12:00:00
esr(config-scheduler-weekdays)# start-time 13:00:00 stop-time 17:00:00
esr(config-scheduler-weekdays)# exit
esr(config-scheduler)# start-date 00:00:00 01 January 2026
esr(config-scheduler)# exit
```

Для настройки правил зон безопасности потребуется создать профиль приложений, которые необходимо будет блокировать:

```
esr(config)# object-group application BLACKLIST
esr(config-object-group-application)# application youtube
esr(config-object-group-application)# exit
```

Для установки правил прохождения трафика из зоны «WAN» в зону «LAN» создадим пару зон и добавим правило, запрещающее проходить трафику приложений, и правило, разрешающее проходить остальному трафику. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair WAN LAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action deny
esr(config-security-zone-pair-rule)# match application BLACKLIST
esr(config-security-zone-pair-rule)# scheduler WORK_HOURS
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# rule 2
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Для установки правил прохождения трафика из зоны «LAN» в зону «WAN» создадим пару зон и добавим правило, запрещающее прохождение трафика приложений, и правило, разрешающее прохождение всего остального трафика. Действие правил разрешается командой **enable**:

```
esr(config)# security zone-pair LAN WAN
esr(config-security-zone-pair)# rule 1
esr(config-security-zone-pair-rule)# action deny
esr(config-security-zone-pair-rule)# match application BLACKLIST
esr(config-security-zone-pair-rule)# scheduler WORK_HOURS
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# rule 2
esr(config-security-zone-pair-rule)# action permit
esr(config-security-zone-pair-rule)# enable
esr(config-security-zone-pair-rule)# exit
esr(config-security-zone-pair)# exit
```

Посмотреть членство портов в зонах можно с помощью команды:

```
esr# show security zone
```

Посмотреть пары зон и их конфигурацию можно с помощью команд:

```
esr# show security zone-pair
esr# show security zone-pair configuration
```

16.6 Настройка списков доступа (IP ACL)

IP Access Control List или IP ACL — список контроля доступа, содержит правила, определяющие прохождение трафика через интерфейс.

16.6.1 Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Создать список контроля доступа и перейти в режим его конфигурирования.	**esr(config)# ip access-list extended <NAME>**	<NAME> – имя создаваемого списка контроля доступа, задаётся строкой до 31 символа.
2	Указать описание конфигурируемого списка контроля доступа (необязательно).	**esr(config-acl-ip)# description <DESCRIPTION>**	<DESCRIPTION> – описание списка контроля доступа, задаётся строкой до 255 символов.
3	Создать правило и перейти в режим его конфигурирования. Правила обрабатываются маршрутизатором в порядке возрастания их номеров.	**esr(config-acl-ip)# rule <ORDER>**	<ORDER> – номер правила, принимает значения [1...4094].

Шаг	Описание	Команда	Ключи
4	Указать действие, которое должно быть применено для трафика, удовлетворяющего заданным критериям.	**esr(config-acl-ip-rule)# action <ACT>**	<ACT> – назначаемое действие: • permit – прохождение трафика разрешается; • deny – прохождение трафика запрещается.
5	Установить имя/номер протокола, для которого должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match protocol <TYPE>**	<TYPE> – тип протокола, принимает значения: esp, icmp, icmp6, ah, eigrp, ospf, igmp, ipip, tcp, pim, udp, vrrp, rsvp, l2tp, gre. При указании значения «any» правило будет срабатывать для любых протоколов.
		esr(config-acl-ip-rule)# match protocol-id <ID>	<ID> – идентификационный номер IP-протокола, принимает значения [0x00-0xFF].
6	Установить IP-адреса отправителя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match source-address { <ADDR> <MASK> | any }**	<ADDR> – IP-адрес отправителя, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <MASK> – маска IP-адреса, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]. Биты маски, установленные в 0, задают биты IP-адреса, исключаемые из сравнения при поиске. При указании значения «any» правило будет срабатывать для любого IP-адреса отправителя/получателя.
7	Установить IP-адреса получателя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match destination-address { <ADDR> <MASK> | any }**	

Шаг	Описание	Команда	Ключи
8	Установить MAC-адреса отправителя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match source-mac <ADDR><WILDCARD>**	<ADDR> – MAC-адрес отправителя, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]; <WILDCARD> – маска MAC-адреса, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]. Биты маски, установленные в 0, задают биты MAC-адреса, исключаемые из сравнения при поиске.
9	Установить MAC-адреса получателя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match destination-mac <ADDR><WILDCARD>**	
10	Установить номер TCP/UDP-порта отправителя, для которого должно срабатывать правило (если указан протокол).	**esr(config-acl-ip-rule)# match source-port <TYPE> {<FROM-PORT> - <TO-PORT> | <PORT>}**	<TYPE> – тип аргумента, устанавливаемый в качестве порта: • **port-range** – указать диапазон портов; • **any** – установить в качестве порта любой порт. <FROM-PORT> – начальный порт диапазона; <TO-PORT> – конечный порт диапазона; <PORT> – указание единичного порта.
11	Установить номер TCP/UDP-порта получателя, для которого должно срабатывать правило (если указан протокол).	**esr(config-acl-ip-rule)# match destination-port <TYPE> {<FROM-PORT> - <TO-PORT> | <PORT>}**	
12	Установить значение 802.1p приоритета, для которого должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match cos <COS>**	<COS> – значение 802.1p приоритета, принимает значения [0..7].
13	Установить значение кода DSCP, для которого должно срабатывать правило (необязательно). Невозможно использовать совместно с IP Precedence.	**esr(config-acl-ip-rule)# match dscp <DSCP>**	<DSCP> – значение кода DSCP, принимает значения [0..63].
14	Установить значение кода IP Precedence, для которого должно срабатывать правило (необязательно). Невозможно использовать совместно с DSCP.	**esr(config-acl-ip-rule)# match ip-precedence <IPP>**	<IPP> – значение кода IP Precedence, принимает значения [0..7].

Шаг	Описание	Команда	Ключи
15	Установить значение идентификационного номера VLAN, для которого должно срабатывать правило (необязательно).	**esr(config-acl-ip-rule)# match vlan <VID>**	<VID> – идентификационный номер VLAN, принимает значения [1..4094].
16	Активировать правило.	**esr(config-acl-ip-rule)# enable**	
17	Указать список контроля доступа к конфигурируемому интерфейсу для фильтрации входящего трафика.	**esr(config-if-gi)# service-acl ip input <NAME>**	<NAME> – имя списка контроля доступа, задаётся строкой до 31 символа.
18	Указать список контроля доступа к конфигурируемому интерфейсу для фильтрации исходящего трафика.	**esr(config-if-gi)# service-acl ip output <NAME>**	<NAME> – имя списка контроля доступа, задаётся строкой до 31 символа.

Также списки доступа могут использоваться для организации политик QoS.

16.6.2 Пример настройки списка доступа

Задача:

Разрешить прохождения трафика только из подсети 192.168.20.0/24.

Решение:

Настроим список доступа для фильтрации по подсетям:

```
esr# configure
esr(config)# ip access-list extended white
esr(config-acl-ip)# rule 1
esr(config-acl-ip-rule)# action permit
esr(config-acl-ip-rule)# match source-address 192.168.20.0 255.255.255.0
esr(config-acl-ip-rule)# enable
esr(config-acl-ip-rule)# exit
esr(config-acl-ip)# exit
```

Применим список доступа на интерфейс Gi1/0/19 для входящего трафика:

```
esr(config)# interface gigabitethernet 1/0/19
esr(config-if-gi)# service-acl ip input white
```

Просмотреть детальную информацию о списке доступа возможно через команду:

```
esr# show ip access-list white
```

16.7 Настройка списков доступа (MAC ACL)

MAC Access Control List или MAC ACL — список контроля доступа, содержит правила на основе L2 заголовка, определяющие прохождение трафика через интерфейс.

16.7.1 Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Создать список контроля доступа и перейти в режим его конфигурирования.	**esr(config)# mac access-list extended <NAME>**	<NAME> – имя создаваемого списка контроля доступа, задаётся строкой до 31 символа.
2	Указать описание конфигурируемого списка контроля доступа (необязательно).	**esr(config-acl-mac)# description <DESCRIPTION>**	<DESCRIPTION> – описание списка контроля доступа, задаётся строкой до 255 символов.
3	Создать правило и перейти в режим его конфигурирования. Правила обрабатываются маршрутизатором в порядке возрастания их номеров.	**esr(config-acl-mac)# rule <ORDER>**	<ORDER> – номер правила, принимает значения [1...4094].
4	Указать действие, которое должно быть применено для трафика, удовлетворяющего заданным критериям.	**esr(config-acl-mac-rule)# action <ACT>**	<ACT> – назначаемое действие: • permit – прохождение трафика разрешается; • deny – прохождение трафика запрещается.
5	Установить имя/номер протокола, для которого должно срабатывать правило (необязательно).	**esr(config-acl-mac-rule)# match protocol <TYPE>**	<TYPE> – тип протокола, принимает значения: arp, ip, ipv6, lacp, lldp, cdp, stp, vtp. При указании значения «any» правило будет срабатывать для любых протоколов.
		esr(config-acl-mac-rule)# match ethertype <ID>	<ID> – идентификационный номер инкапсулированного протокола, принимает значения [0x0600-0xFFFF].

Шаг	Описание	Команда	Ключи
8	Установить MAC-адреса отправителя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-mac-rule)# match source-mac <ADDR><WILDCARD>**	<ADDR> – MAC-адрес отправителя, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]; <WILDCARD> – маска MAC-адреса, задаётся в виде XX:XX:XX:XX:XX:XX, где каждая часть принимает значения [00..FF]. Биты маски, установленные в 0, задают биты MAC-адреса, исключаемые из сравнения при поиске.
9	Установить MAC-адреса получателя, для которых должно срабатывать правило (необязательно).	**esr(config-acl-mac-rule)# match destination-mac <ADDR><WILDCARD>**	
12	Установить значение 802.1p приоритета, для которого должно срабатывать правило (необязательно).	**esr(config-acl-mac-rule)# match cos <COS>**	<COS> – значение 802.1p приоритета, принимает значения [0..7].
15	Установить значение идентификационного номера VLAN, для которого должно срабатывать правило (необязательно).	**esr(config-acl-mac-rule)# match vlan <VID>**	<VID> – идентификационный номер VLAN, принимает значения [1..4094].
16	Активировать правило.	**esr(config-acl-mac-rule)# enable**	
17	Указать список контроля доступа к конфигурируемому интерфейсу для фильтрации входящего трафика.	**esr(config-if-gi)# service-acl mac input <NAME>**	<NAME> – имя списка контроля доступа, задаётся строкой до 31 символа.
18	Указать список контроля доступа к конфигурируемому интерфейсу для фильтрации исходящего трафика.	**esr(config-if-gi)# service-acl mac output <NAME>**	<NAME> – имя списка контроля доступа, задаётся строкой до 31 символа.

16.7.2 Пример настройки списка доступа

Задача:

Разрешить прохождение ARP запросов только с VLAN 343.

Решение:

Настроим список доступа для фильтрации по протоколу и vlan:

```
esr# configure
esr(config)# mac access-list extended white
esr(config-acl-mac)# rule 1
esr(config-acl-mac-rule)# action permit
esr(config-acl-mac-rule)# match protocol arp
esr(config-acl-mac-rule)# match vlan 343
esr(config-acl-mac-rule)# enable
esr(config-acl-mac-rule)# exit
esr(config-acl)# exit
```

Применим список доступа на интерфейс Gi1/0/19 для входящего трафика:

```
esr(config)# interface gigabitethernet 1/0/19
esr(config-if-gi)# service-acl mac input white
```

Просмотреть детальную информацию о списке доступа возможно через команду:

```
esr# show mac access-list white
```

16.8 Проксирование HTTP/HTTPS-трафика

HTTP/HTTPS прокси-сервер ESBC выступает в роли промежуточного звена (посредника) между пользовательскими устройствами (клиентами) и целевыми веб-сервером. Прокси-сервер позволяет проксировать HTTP/HTTPS запросы на веб-сервер (группу веб-серверов), с возможностью балансировки трафика, изменения и фильтрации URL, модификации ответов, проверки SSL-сертификатов сервера и клиента.

Минимальная конфигурация для обеспечения проксирования трафика HTTP/HTTPS должна состоять из одного HTTP/HTTPS прокси-сервера (ip http proxy) и одной группы вышестоящих серверов (ip http upstream-group).

Для модификации тела (body) http-ответов требуется дополнительно настроить таблицу модификации.

> ⚠ Перед конфигурацией HTTP/HTTPS прокси-сервера (ip http proxy) рекомендуется предварительно настроить группу вышестоящих серверов (ip http upstream-group) и в случае необходимости таблицу модификации (ip http mod-table).

При использовании протокола https, требуется выбрать сертификаты и приватный ключ для установки TLS-соединения. Также при использовании https предоставлена возможность управления версией TLS и возможность проверки сертификатов клиента и сервера. Описание работы с сертификатами представлено в разделе Настройка криптопрофилей документации.

16.8.1 Алгоритм настройки группы вышестоящих серверов

В конфигурации ESBC поддерживается до 10 upstream-group. В каждой группе возможно использовать до 50 хостов (hosts).

Шаг	Действие	Команда	Ключи	Описание команды
1	Создать группу вышестоящих серверов.	**vesbc(config)# ip http upstream-group <NAME>**	<NAME> – название upstream-group	Переход в раздел настройки группы вышестоящих серверов.

Шаг	Действие	Команда	Ключи	Описание команды
2	Выбрать режим балансировки трафика (не обязательно).	**vesbc(config-http-upstream)# balancing {carp | round-robin | sourcehash| weighted-round-robin}**	{carp | round-robin | sourcehash| weighted-round-robin} – выбор режима балансировки между хостами данной группы	Описание режимов балансировки исходящих http/https запросов до вышестоящих серверов: • **carp** – балансировка на основе хэширования URL; • **round-robin (используется по умолчанию)** – алгоритм циклического распределения нагрузки, при котором каждый новый запрос передается следующему хосту в списке по очереди. После достижения конца списка процесс возобновляется с первого хоста; • **sourcehash** – алгоритм балансировки нагрузки, при котором выбор целевого хоста осуществляется на основе хэширования IP-адреса источника (клиента) запроса. Это гарантирует, что все запросы от одного и того же пользователя будут направляться на один и тот же хост в течение всей сессии; • **weighted-round-robin** – алгоритм циклического распределения нагрузки с учетом весовых коэффициентов. Каждому хосту назначается вес (командой **weight** в настройках хоста), пропорционально которому он получает долю запросов. Чем выше вес хоста, тем больше запросов на него направляется в рамках одного цикла.

Шаг	Действие	Команда	Ключи	Описание команды
3	Создать вышестоящий сервер.	**vesbc(config-http-upstream)# host <INDEX>**	<INDEX> – номер хоста от 0 до 49	Создание вышестоящего сервера или выбор созданного сервера для редактирования.
4	Указать IP-адрес или доменное имя вышестоящего сервера.	**vesbc(config-http-upstream-host)# address {IPv4 | IPv6 | DNS}**	{IPv4 | IPv6 | DNS} – IP-адрес или доменное имя вышестоящего сервера	IPv6-адрес или доменное имя сервера, на который будут направляться исходящие http/https запросы через данную upstream-group.
5	Указать http-порт вышестоящего сервера (необязательно для работы по https).	**vesbc(config-http-upstream-host)# http port <1-65535>**	<1-65535> – номер tcp-порта для http	Порт для обработки входящих HTTP-соединений вышестоящего сервера.
6	Указать https-порт вышестоящего сервера (необязательно для работы по http).	**vesbc(config-http-upstream-host)# https port <1-65535>**	<1-65535> – номер tcp-порта для https	Порт для обработки входящих HTTPS-соединений вышестоящего сервера.
7	Указать количество возможных одновременных tcp-соединений до вышестоящего сервера (необязательно).	**vesbc(config-http-upstream-host)# max-connections <0-65535>**	<0-65535> – количество возможных одновременных tcp-соединений до вышестоящего сервера	Количество возможных одновременных tcp-соединений до вышестоящего сервера. По умолчанию без ограничений.
8	Указать вес вышестоящего сервера (необязательно).	**vesbc(config-http-upstream-host)# weight <0-65535>**	<0-65535> – вес вышестоящего сервера	Используется при режиме балансировки **weighted-round-robin** в группе вышестоящих серверов. По умолчанию 100.

16.8.2 Алгоритм настройки таблицы модификации

В конфигурации ESBC поддерживается до 30 таблиц модификации ответов htpp/https. В каждой таблице поддерживается до 64 модификаций.

> ⚠ Модификаторы применяются только к **телу** (body) http/https **ответов**.

Модификации применяются к телу сообщения последовательно, сначала применяется таблица модификации, указанная в настройке прокси-сервера (ip http proxy), а затем из настройки локации этого сервера (location).

Шаг	Действие	Команда	Ключи	Описание команды
1	Создать таблицу модификации.	**vesbc(config)# ip http mod-table <NAME>**	<NAME> – название таблицы модификации	Переход в раздел настройки таблицы модификации.
2	Создать правило модификации.	**vesbc(config-http-mod-table)# modification <INDEX>**	<INDEX> – номер модификации от 0 до 63	Создание модификации или выбор созданной модификации для редактирования.
3	Указать PCRE-паттерн для отбора содержимого тела http/https ответа.	**vesbc(config-http-mod-table-modification)# value-pattern <WORD>**	<WORD> – PCRE-паттерн от 1 до 127 символов	Отбор содержимого тела http/https ответа для выполнения замены по PCRE-паттерну.
4	Указать текст для замены (необязательно).	**vesbc(config-http-mod-table-modification)# replacement <WORD>**	<WORD> – от 1 до 127 символов	Текст, на который будет заменено содержимое, выбранное параметром "value-pattern". **Если не указывать параметр "replacement", то содержимое, отобранное параметром "value-pattern", будет удалено из тела ответа.**

16.8.3 Алгоритм настройки HTTP/HTTPS-прокси сервера

В конфигурации ESBC поддерживается до 10 HTTP/HTTPS-прокси серверов. В каждом сервере возможно использовать до 40 локаций (location).

Шаг	Действие	Команда	Ключи	Описание команды
1	Создать HTTP/HTTPS прокси-сервер.	**vesbc(config)# ip http proxy <NAME>**	<NAME> – название HTTP/HTTPS-сервера	Переход в раздел настройки HTTP/HTTPS-прокси.
2	Привязать криптопрофиль (необязательно для работы по http, обязательно для работы по https).	**vesbc(config-http-proxy)# crypto profile <NAME>**	<NAME> – название криптопрофиля	Выбор криптопрофиля. Используется для работы в режиме HTTPS-прокси. Выбранный профиль будет использоваться для установления защищенного соединения для входящих HTTPS-запросов (ESBC выступает в роли сервера).
3	Указать http-порт (необязательно для работы по https).	**vesbc(config-http-proxy)# http port <1-65535>**	<1-65535> – номер tcp-порта для http	Порт для обработки входящих HTTP-соединений.

Шаг	Действие	Команда	Ключи	Описание команды
4	Указать https-порт (необязательно для работы по http).	**vesbc(config-http-proxy)# https port <1-65535>**	<1-65535> – номер tcp-порта для https	Порт для обработки входящих HTTPS-соединений.
5	Указать IP-адрес или интерфейс.	**vesbc(config-http-proxy)# ip address { <IP-address> | <interface> }**	IPv4-адрес или interface <TYPE> <INDEX>, где TYPE – тип интерфейса, INDEX – номер интерфейса	IPv4-адрес или интерфейс с IPv4-адресом для обработки входящих HTTP/HTTPS-соединений.
6	Указать IPv6-адрес или интерфейс (необязательно).	**vesbc(config-http-proxy)# ipv6 address { <IPv6-address> | <interface> }**	IPv6-адрес или interface <TYPE> <INDEX>, где TYPE – тип интерфейса, INDEX – номер интерфейса	IPv6-адрес или интерфейс с IPv6-адресом для обработки входящих HTTP/HTTPS-соединений.
7	Указать таблицу модификации (необязательно).	**vesbc(config-http-proxy)# mod-table <NAME>**	<NAME> – название таблицы модификации	Таблица модификации тела сообщения в HTTP/HTTPS-ответах, отправляемых через данный прокси-сервер.
8	Создать локацию и перейти в ее настройки.	**vesbc(config-http-proxy)# location <INDEX>**	<INDEX> – номер локации от 0 до 39	Локация определяет точку выхода запроса (проксирования) на вышестоящий сервер (upstream-group) по совпадению url-pattern из входящего HTTP/HTTPS-запроса.
9	Привязать криптопрофиль (необязательно для работы по http, обязательно для работы по https).	**vesbc(config-http-proxy-location)# crypto profile <NAME>**	<NAME> – название криптопрофиля	Выбор криптопрофиля. Используется для работы в режиме HTTPS-прокси. Выбранный профиль будет использоваться для установления защищенного соединения с вышестоящим сервером (ESBC выступает в роли клиента).
10	Указать IP-адрес или интерфейс.	**vesbc(config-http-proxy-location)# ip address { <IP-address> | <interface> }**	IPv4-адрес или interface <TYPE> <INDEX>, где TYPE – тип интерфейса, INDEX – номер интерфейса	IPv4-адрес или интерфейс с IPv4-адресом для передачи HTTP/HTTPS-запросов (проксирования) на вышестоящий сервер (upstream-group).

Шаг	Действие	Команда	Ключи	Описание команды
11	Указать IPv6-адрес или интерфейс (необязательно).	**vesbc(config-http-proxy-location)# ipv6 address { <IPv6-address> | <interface> }**	IPv6-адрес или interface <TYPE> <INDEX>, где TYPE – тип интерфейса, INDEX – номер интерфейса	IPv6-адрес или интерфейс с IPv6-адресом для передачи HTTP/HTTPS-запросов (проксирования) на вышестоящий сервер (upstream-group).
12	Указать таблицу модификации (необязательно).	**vesbc(config-http-proxy-location)# mod-table <NAME>**	<NAME> – название таблицы модификации	Таблица модификации тела сообщения в HTTP/HTTPS-ответах, обрабатываемых данной локацией.
13	Указать группу вышестоящих серверов.	**vesbc(config-http-proxy-location)# upstream group <NAME>**	<NAME> – название upstream group	Группа вышестоящих серверов, на которую будет перенаправлен входящий HTTP/HTTPS запрос, обработанный данной локацией.
14	Указать upstream path для замены URL оригинального запроса (необязательно).	**vesbc(config-http-proxy-location)# upstream path <PATH>**	<PATH> – путь URL от 1 до 127 символов, начинающийся с "/"	Путь URL, на который будет заменяться путь из входящих HTTP/HTTPS-запросов, обрабатываемых данной локацией при проксировании их через upstream group.
15	Выбрать режим работы с вышестоящим сервером (необязательно для работы по http).	**vesbc(config-http-proxy-location)# upstream scheme { http | https}**	{http | https} – выбор режима работы с вышестоящим сервером	Использование протокола HTTP или HTTPS для исходящих запросов, обрабатываемых данной локацией для передачи их на вышестоящий сервер. По умолчанию используется http.
16	Указать URL-pattern для отбора URL входящего сообщения.	**vesbc(config-http-proxy-location)# url-pattern <WORD>**	<WORD> – PCRE-паттерн от 1 до 127 символов для отбора URL входящего сообщения	Отбор URL входящего сообщения для обработки этого сообщения данной локацией.

При настройке http-https прокси-сервера на ESBC не требуется дополнительной конфигурации Firewall на сетевых интерфейсах, необходимые правила для пропуска входящего http/https-трафика будут созданы автоматически в соответствии с конфигурацией ip http proxy.

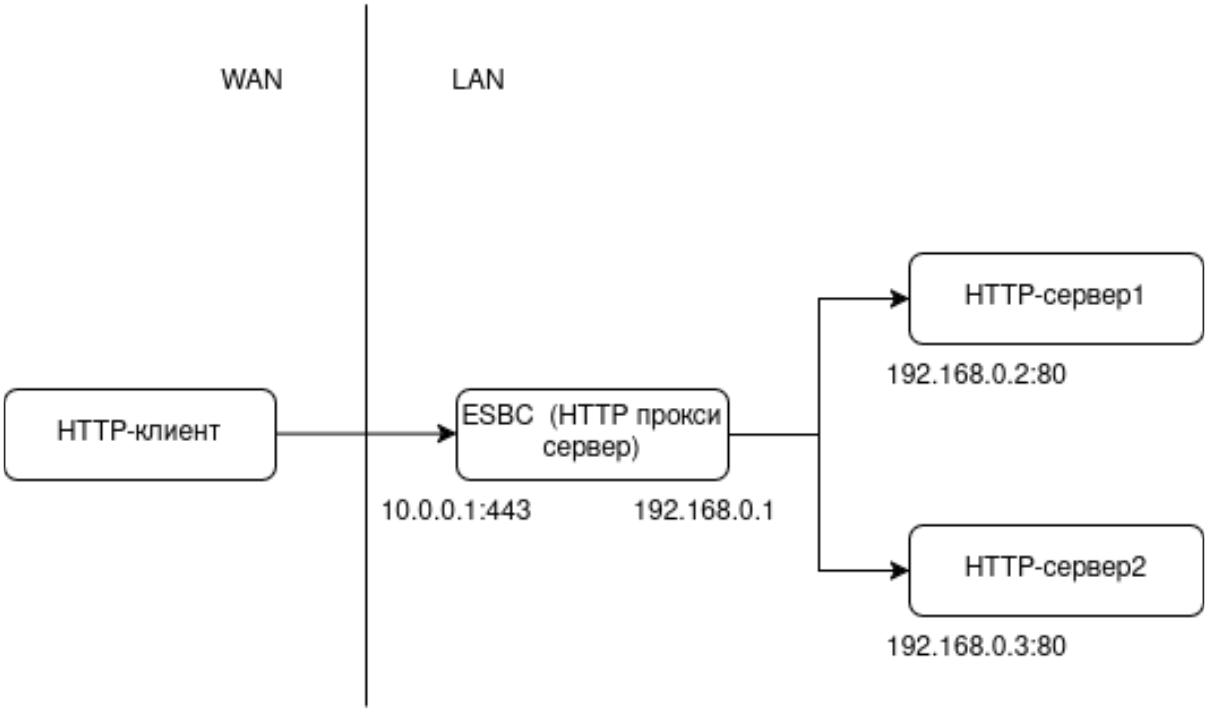
16.8.4 Пример настройки HTTP-прокси

Задача:

Настроить на ESBC HTTP прокси-сервер для передачи всех входящих https-запросов (по любым URL) на два вышестоящих http-сервера с приоритетной балансировкой weighted-round-robin на HTTP-сервер1.

Дополнительные условия:

- Для входящих запросов необходимо использовать https
- Для проксирования запросов необходимо использовать http
- В ответах от вышестоящего сервера необходимо заменить слово "test_inside" на "test_outside"
- Использовать проверку сертификата клиента



Решение:

1. Настроить IP-адреса на сетевых интерфейсах:

```
vesbc(config)#
vesbc(config)# interface gigabitethernet 1/0/1
vesbc(config-if-gi)# ip address 10.0.0.1/24
vesbc(config-if-gi)# exit
vesbc(config)#
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# ip address 192.168.0.1/24
vesbc(config-if-gi)# exit
vesbc(config)#
```

2. Настроить криптопрофиль (предполагается, что файлы сертификатов уже загружены на ESBC):

```
vesbc(config)#
vesbc(config)# crypto profile WAN
#Указать сертификат CA HTTP-клиента
vesbc(config-crypto-profile)# ca RootCA.pem
vesbc(config-crypto-profile)# cert client.crt
vesbc(config-crypto-profile)# private-key client.key
#Включить проверку сертификата клиента
vesbc(config-crypto-profile)# verify client-cert
vesbc(config-crypto-profile)# exit
vesbc(config)#
```

3. Настроить группу вышестоящих серверов с хостами HTTP-сервер1 и HTTP-сервер2:

```
vesbc(config)# ip http upstream-group LAN
vesbc(config-http-upstream)# balancing weighted-round-robin
vesbc(config-http-upstream)# host 0
vesbc(config-http-upstream-host)# address 192.168.0.2

#Указать вес больше, чем у хоста 1:
vesbc(config-http-upstream-host)# weight 2
vesbc(config-http-upstream-host)# http port 80
vesbc(config-http-upstream-host)# exit
vesbc(config-http-upstream)# host 1
vesbc(config-http-upstream-host)# address 192.168.0.3
vesbc(config-http-upstream-host)# weight 1
vesbc(config-http-upstream-host)# http port 80
vesbc(config-http-upstream-host)# exit
vesbc(config-http-upstream)# exit
```

4. Настроить таблицу модификации для изменения тела ответов:

```
vesbc(config)# ip http mod-table MOD_TABLE
vesbc(config-http-mod-table)# modification 0

#Поиск в теле ответа "test_inside":
vesbc(config-http-mod-table-modification)# value-pattern test_inside

#Замена на "test_outside":
vesbc(config-http-mod-table-modification)# replacement test_outside
vesbc(config-http-mod-table-modification)# exit
vesbc(config-http-mod-table)# exit
```

5. Настроить http прокси-сервер:

```
vesbc(config)# ip http proxy TEST_PROXY
#Указать IP-адрес WAN-интерфейса или сам интерфейс для получения входящих https-
соединений:
vesbc(config-http-proxy)# ip address 10.0.0.1

#Указать порт для обработки входящих https-соединений:
vesbc(config-http-proxy)# https port 443

#Указать криптопрофиль, настроенный в п.2:
vesbc(config-http-proxy)# crypto profile WAN

#Указать таблицу модификации, настроенную в п.4:
vesbc(config-http-proxy)# mod-table MOD_TABLE
vesbc(config-http-proxy)# location 0

#Указать IP-адрес LAN-интерфейса или сам интерфейс, с которого будут передаваться http-
сообщения на вышестоящие серверы:
vesbc(config-http-proxy-location)# ip address 192.168.0.1

#Указать паттерн PCRE, например ".*", под который попадают все URL:
vesbc(config-http-proxy-location)# url-pattern .*

#Указать группу вышестоящих серверов, настроенную в п.3:
vesbc(config-http-proxy-location)# upstream group LAN
vesbc(config-http-proxy-location)# exit
vesbc(config-http-proxy)# exit
vesbc(config)#
```

6. Применить и сохранить конфигурацию:

```
vesbc(config)# do commit
vesbc(config)# do confirm
```

16.9 Настройка IPS/IDS

⚠ Данная функция активируется только при наличии лицензии.

IPS/IDS (Intrusion Prevention System/Intrusion Detection System) – система предотвращения вторжений – программная система сетевой и компьютерной безопасности, обнаруживающая вторжения или нарушения безопасности и автоматически защищающая от них.

Работа системы основана на сигнатурном анализе трафика. Сигнатуры для систем IPS/IDS принято называть правилами. Устройства ESR позволяют скачивать актуальные правила с открытых источников в сети Интернет или с корпоративного сервера. Также с помощью CLI можно создавать свои специфические правила.

16.9.1 Алгоритм базовой настройки

Шаг	Описание	Команда	Ключи
1	Создать политику безопасности IPS/IDS.	**esr(config)# security ips policy <NAME>**	<NAME> – имя политики безопасности, задаётся строкой до 32 символов.
2	Задать описание политики (необязательно).	**esr(config-ips-policy)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.
3	Задать профиль IP-адресов, которые будет защищать IPS/IDS.	**esr(config-ips-policy)# protect network-group <OBJ-GROUP-NETWORK_NAME>**	<OBJ-GROUP-NETWORK-NAME> – имя профиля защищаемых IP-адресов, задается строкой до 32 символов.
4	Задать профиль IP-адресов, внешних для IPS/IDS (необязательно).	**esr(config-ips-policy)# external network-group <OBJ-GROUP-NETWORK_NAME>**	<OBJ-GROUP-NETWORK-NAME> – имя профиля внешних IP-адресов, задается строкой до 32 символов.
5	Перейти в режим конфигурирования IPS/IDS.	**esr(config)# security ips**	
6	Назначить политику безопасности IPS/IDS.	**esr(config-ips)# policy <NAME>**	<NAME> – имя политики безопасности, задаётся строкой до 32 символов.
7	Использовать все ресурсы ESR для IPS/IDS (необязательно).	**esr(config-ips)# perfomance max**	По умолчанию для IPS/IDS отдается половина доступных ядер процессора.

Шаг	Описание	Команда	Ключи
8	Задать параметры удаленного сервера для отправки статистики работы сервиса IPS/IDS в формате EVE (elasticsearch) (необязательно).	**esr(config-ips)# logging remote-server { <ADDR> | <IPV6-ADDR> } [<TRANSPORT>] [<PORT>] [source-address { <SRC-ADDR> | <IPV6-SRC-ADDR> | object-group <NETWORK_OBJ_GROUP_NAME> }]**	<ADDR> – IP-адрес, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF]; <TRANSPORT> – протокол передачи данных, по умолчанию – UDP, принимает значения: • TCP – передача данных осуществляется по протоколу TCP; • UDP – передача данных осуществляется по протоколу UDP. <PORT> – номер TCP/UDP-порта, опциональный параметр, принимает значения [1..65535], по умолчанию 514; <SRC-ADDR> – IPv4-адрес маршрутизатора, который будет использоваться в качестве IP-адреса источника в отправляемых syslog-пакетах, по умолчанию – IPv4-адрес интерфейса, с которого отправляются пакеты; <IPV6-SRC-ADDR> – IPv6-адрес маршрутизатора, который будет использоваться в качестве IPv6-адреса источника в отправляемых syslog-пакетах, по умолчанию – IPv6-адрес интерфейса, с которого отправляются пакеты; <NETWORK_OBJ_GROUP_NAME> – список адресов, которые будут использоваться в качестве source address.

Шаг	Описание	Команда	Ключи
9	Установить интервал отправки статистики работы сервиса IPS/IDS в формате EVE (elasticsearch) (необязательно).	**esr(config-ips)# logging update-interval <INTERVAL>**	<INTERVAL> – интервал отправки статистики работы сервиса IPS/IDS, задаётся в минутах.
10	Заблокировать передачу трафика при начальной загрузке до запуска сервиса IPS/IDS и загрузки хотя бы одного настроенного или существующего правила (необязательно).	**esr(config-ips)# fail-close enable**	
11	Установить размер виртуальных очередей (необязательно).	**esr(config-ips)# queue-limit <QUEUE-LIMIT>**	<QUEUE-LIMIT> – предельное количество пакетов в виртуальной очереди, принимает значения в диапазоне [32..4096] Размер очереди по умолчанию 1024
12	Активировать сервис IPS.	**esr(config-ips)# enable**	
13	Активировать IPS/IDS на интерфейсе.	**esr(config-if-gi)# service-ips { inline | monitor }**	inline – этот режим устанавливается, когда ESR с сервисом IPS/IDS ставится в разрыв сети. monitor – этот режим устанавливается, когда ESR с сервисом IPS/IDS мониторит зеркалируемый трафик.

16.9.2 Алгоритм настройки автообновления правил IPS/IDS из внешних источников

Шаг	Описание	Команда	Ключи
1	Задать имя внешнего хранилища скачиваемых правил (необязательно).	**esr(config-ips)# storage-path { usb://<USB-NAME>:/ | mmc://<MMC-NAME>:/ }**	<USB-NAME> – имя подключенного USB-носителя. Имя можно узнать в выводе команды show storage-devices usb; <MMC-NAME> – имя подключенного MMC-носителя. Имя можно узнать в выводе команды show storage-devices mmc. Для использования с системой IPS/IDS на внешнем носителе должен быть создан раздел файловой системы в формате exFAT.
2	Перейти в режим конфигурирования автообновлений.	**esr(config-ips)# auto-upgrade**	
3	Задать имя и перейти в режим конфигурирования пользовательского сервера обновлений.	**esr(config-ips-auto-upgrade)# user-server <WORD>**	<WORD> – имя сервера, задаётся строкой до 32 символов.
4	Задать описание пользовательского сервера обновлений (необязательно).	**esr(config-ips-upgrade-user-server)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.
5	Задать URL.	**esr(config-ips-upgrade-user-server)# url <URL>**	<URL> – текстовое поле, содержащее URL-ссылку длиной от 8 до 255 символов. В качестве URL-ссылки может быть указан: • файл правил с расширение .rule; • файл классификатора правил с именем classification.config; • каталог на сервере содержащий файлы правил и/или файл классификатора правил.

Шаг	Описание	Команда	Ключи
6	Задать частоту проверки обновлений (необязательно).	**esr(config-ips-upgrade-user-server)# upgrade interval <HOURS>**	<HOURS> – интервал обновлений в часах, от 1 до 240. Значение по умолчанию: 24 часа.
7	Активизировать пользовательский сервер обновлений.	**esr(config-ips-upgrade-user-server)# enable**	

Для правил IPS/IDS, загружаемых из внешних источников, на маршрутизаторах ESR выделена отдельная область энергозависимой памяти.
Размер этой области зависит от модели ESR:

- ESR-1X – 25 МБ;
- ESR-2X – 50 МБ;

Для всех остальных моделей – 100 МБ.
Если настроить слишком много источников правил или загружать правила, превышающие указанные лимиты, то маршрутизатор будет выдавать сообщения об ошибке %STORAGE_IPS_MGR-I-ERR: There no free space in rules directory.
В этом случае стоит уменьшить объем запрашиваемых правил или использовать внешнее хранилище.

16.9.3 Рекомендуемые открытые источники обновления правил

SSL Blacklist

Чёрный список SSL (SSLBL) – это проект abuse.ch, целью которого является обнаружение вредоносных SSL-соединений путём идентификации и внесения в чёрный список SSL-сертификатов, используемых серверами управления ботнетами.

https://sslbl.abuse.ch/blacklist/sslblacklist_tls_cert.rules – набор правил SSL-сертификатов от SSLBL используется для обнаружения и/или блокировки вредоносных SSL-соединений в вашей сети на основе отпечатка SSL-сертификата. Набор правил SSL-сертификатов генерируется каждые 5 минут. Рекомендуется запрашивать его не чаще, чем раз в 5 минут.

https://sslbl.abuse.ch/blacklist/ja3_fingerprints.rules – набор правил JA3 FingerprintRuleset от SSLBL используется для обнаружения и/или блокировки вредоносных SSL-соединений в вашей сети на основе отпечатка JA3. Набор правил для отпечатков пальцев Suricata JA3 генерируется каждые 5 минут. Рекомендуется запрашивать его не чаще, чем раз в 5 минут.

Отпечатки JA3, внесённые в чёрный список SSLBL, были собраны путём анализа более 25 000 000 PCAP-файлов с образцами вредоносного ПО. Эти отпечатки ещё не были протестированы на известном безопасном трафике и могут привести к значительному количеству ложных срабатываний.

Feodo Tracker

Feodo Tracker – это проект abuse.ch, целью которого является обмен информацией о серверах управления ботнетами, связанными с Dridex, Emotet (также известным как Heodo), TrickBot, QakBot (также известным как QuakBot/Qbot) и BazarLoader (также известным как BazarBackdoor).

https://feodotracker.abuse.ch/downloads/feodotracker.rules – набор правил используется для обнаружения и/или блокировки сетевых подключений к хост-серверам (комбинация IP-адреса и порта). Набор правил генерируется каждые 5 минут. Рекомендуется обновлять набор правил IDS каждые 5–15 минут, чтобы обеспечить лучшую защиту от Dridex, Emotet, TrickBot, QakBot и BazarLoader.

> ⚠ Поскольку IP-адреса перерабатываются и используются повторно, в этот список блокировки входят только C2-серверы ботнетов, которые либо активны, либо в последний раз использовались в течение последних 30 дней. Таким образом, процент ложных срабатываний в этом списке блокировки должен быть низким.

https://feodotracker.abuse.ch/downloads/feodotracker_aggressive.rules – набор правил IDS с полным списком всех C2-серверов ботнетов. Однако, поскольку IP-адреса используются повторно, количество ложных срабатываний в этом наборе правил намного выше.

> ✕ Не рекомендуется использовать агрессивную версию индикаторов компрометации ботнета C2 (IOC), так как она определённо вызовет ложные срабатывания.

Travis Green

Travis Green – набор правил для поиска угроз от специалиста по кибербезопасности Тревиса Грина.

https://raw.githubusercontent.com/travisbgreen/hunting-rules/master/hunting.rules

Etnetera Core

Набор правил с «агрессивными» IP-адресами от центра кибербезопасности компании Etnetera Core.

https://security.etnetera.cz/feeds/etn_aggressive.rules

16.9.4 Пример настройки IPS/IDS с автообновлением правил

Задача:

Организовать защиту локальной сети с автообновлением правил из открытых источников.

192.168.1.0/24 – локальная сеть.

Решение:

Создадим профиль адресов защищаемой локальной сети:

```
esr(config)# object-group network LAN
esr(config-object-group-network)# ip prefix 192.168.1.0/24
esr(config-object-group-network)# exit
```

Настроим на ESR DNS-клиента для разрешения имен источников обновления правил IPS/IDS:

```
esr(config)# domain lookup enable
esr(config)# domain nameserver 8.8.8.8
```

Создадим политику безопасности IPS/IDS:

```
esr(config)# security ips policy OFFICE
esr(config-ips-policy)# description "My Policy"
esr(config-ips-policy)# protect network-group LAN
```

Разрешим работу IPS/IDS на интерфейсе локальной сети bridge 1:

```
esr(config)# bridge 1
esr(config-bridge)# service-ips inline
```

Настроим параметры IPS/IDS:

```
esr(config)# security ips
esr(config-ips)# logging remote-server 192.168.10.1
esr(config-ips)# logging update-interval 15
esr(config-ips)# policy OFFICE
esr(config-ips)# enable
```

Устройство будет использоваться только как шлюз безопасности, поэтому отдадим сервису IPS/IDS все доступные ресурсы:

```
esr(config-ips)# perfomance max
```

Настроим автообновление правил с сайтов etnetera.cz и Abuse.ch:

```
esr(config-ips)# auto-upgrade
esr(config-auto-upgrade)# user-server Aggressive
esr(config-ips-upgrade-user-server)# description "Etnetera aggressive IP blacklist"
esr(config-ips-upgrade-user-server)# url https://security.etnetera.cz/feeds/
etn_aggressive.rules
esr(config-ips-upgrade-user-server)# upgrade interval 4
esr(config-ips-upgrade-user-server)# enable
esr(config-ips-upgrade-user-server)# exit
esr(config-auto-upgrade)# user-server SSL-BlackList
esr(config-ips-upgrade-user-server)# description "Abuse.ch SSL Blacklist"
esr(config-ips-upgrade-user-server)# url https://sslbl.abuse.ch/blacklist/
sslblacklist_tls_cert.rules
esr(config-ips-upgrade-user-server)# upgrade interval 4
esr(config-ips-upgrade-user-server)# enable
esr(config-ips-upgrade-user-server)# exit
```

16.9.5 Алгоритм настройки базовых пользовательских правил

Шаг	Описание	Команда	Ключи
1	Задать имя и перейти в режим конфигурирования набора пользовательских правил.	**esr(config)# security ips-category user-defined <WORD>**	<WORD> – имя набора пользовательских правил, задаётся строкой до 32 символов.
2	Задать описание набора пользовательских правил (не обязательно).	**esr(config-ips-category)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.
3	Создать правило и перейти в режим конфигурирования правила.	**esr(config-ips-category)# rule <ORDER>**	<ORDER> – номер правила, принимает значения [1..512].
4	Задать описание правила (необязательно).	**esr(config-ips-category-rule)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.

Шаг	Описание	Команда	Ключи
5	Указать действие данного правила.	**esr(config-ips-category-rule)# action { alert | reject | pass | drop }**	• alert – прохождение трафика разрешается, и сервис IPS/IDS генерирует сообщение; • reject – прохождение трафика запрещается. Если это TCP-трафик, отправителю и получателю посылается пакет TCP-RESET, для остального типа трафика посылается пакет ICMP-ERROR. Сервис IPS/IDS генерирует сообщение; • pass – прохождение трафика разрешается; • drop – прохождение трафика запрещается, и сервис IPS/IDS генерирует сообщение.
6	Установить имя IP-протокола, для которого должно срабатывать правило.	**esr(config-ips-category-rule)# protocol <PROTOCOL>**	<PROTOCOL> – принимает значения any/ip/icmp/http/tcp/udp. При указании значения «any» правило будет срабатывать для любых протоколов.

Шаг	Описание	Команда	Ключи
7	Установить IP-адреса отправителя, для которых должно срабатывать правило.	**esr(config-ips-category-rule)# source-address {ip <ADDR> | ip-prefix <ADDR/LEN> | object-group <OBJ_GR_NAME> | policy-object-group { protect | external } | any }**	<ADDR> – IP-адрес отправителя, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <ADDR/LEN> – IP-подсеть отправителя, задаётся в виде AAA.BBB.CCC.DDD/EE, где каждая часть AAA – DDD принимает значения [0..255] и LEN принимает значения [1..32]. < OBJ_GR_NAME> – имя профиля IP-адресов, который содержит IP-адреса отправителя, задаётся строкой до 31 символа. • protect – устанавливает в качестве адресов отправителя и protect-адреса определенные адреса в политике IPS/IDS; • external – устанавливает в качестве адресов отправителя и external-адреса определенные адреса в политике IPS/IDS. При указании значения «any» правило будет срабатывать для любого IP-адреса отправителя.
8	Установить номера TCP/UDP-портов отправителя, для которых должно срабатывать правило. Для значения protocol icmp, значение source-port может быть только any.	**esr(config-ips-category-rule)# source-port {any | <PORT> | object-group <OBJ-GR-NAME> }**	<PORT> – номер TCP/UDP-порта отправителя, принимает значения [1..65535]. <OBJ_GR_NAME> – имя профиля TCP/UDP-портов отправителя, задаётся строкой до 31 символа. При указании значения «any» правило будет срабатывать для любого TCP/UDP-порта отправителя.

Шаг	Описание	Команда	Ключи
9	Установить IP-адреса получателя, для которых должно срабатывать правило.	**esr(config-ips-category-rule)# destination-address {ip <ADDR> | ip-prefix <ADDR/LEN> | object-group <OBJ_GR_NAME> | policy-object-group { protect | external } | any }**	<ADDR> – IP-адрес получателя, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <ADDR/LEN> – IP-подсеть получателя, задаётся в виде AAA.BBB.CCC.DDD/EE, где каждая часть AAA – DDD принимает значения [0..255] и LEN принимает значения [1..32]; <OBJ_GR_NAME> – имя профиля IP-адресов, который содержит IP-адреса получателя, задаётся строкой до 31 символа. • protect – устанавливает в качестве адресов получателя protect-адреса, определенные в политике IPS/IDS; • external – устанавливает в качестве адресов получателя external-адреса, определенные в политике IPS/IDS. При указании значения «any» правило будет срабатывать для любого IP-адреса получателя.
10	Установить номера TCP/UDP-портов получателя, для которых должно срабатывать правило. Для значения protocol icmp, значение destination-port может быть только any.	**esr(config-ips-category-rule)# destination-port {any | <PORT> | object-group <OBJ-GR-NAME> }**	<PORT> – номер TCP/UDP-порта получателя, принимает значения [1..65535]; <OBJ_GR_NAME> – имя профиля TCP/UDP-портов получателя, задаётся строкой до 31 символа. При указании значения «any» правило будет срабатывать для любого TCP/UDP-порта получателя.

Шаг	Описание	Команда	Ключи
11	Установить направление потока трафика, для которого должно срабатывать правило.	**esr(config-ips-category-rule)# direction { one-way | round-trip }**	• one-way – трафик передаётся в одну сторону. • round-trip – трафик передаётся в обе стороны.
12	Определить сообщение, которое IPS/IDS будет записывать в лог при срабатывании этого правила.	**esr(config-ips-category-rule)# meta log-message <MESSAGE>**	<MESSAGE> – текстовое сообщение, задаётся строкой до 129 символов.

Шаг	Описание	Команда	Ключи
13	Определить классификацию трафика, которая будет записывать в лог при срабатывании этого правила (необязательно).	**esr(config-ips-category-rule)# meta classification-type { not-suspicious | unknown | bad-unknown | attempted-recon | successful-recon-limited | successful-recon-largescale | attempted-dos | successful-dos | attempted-user | unsuccessful-user | successful-user | attempted-admin | successful-admin | rpc-portmap-decode | shellcode-detect | string-detect | suspicious-filename-detect | suspicious-login | system-call-detect | tcp-connection | trojan-activity | unusual-client-port-connection | network-scan | denial-of-service | non-standard-protocol | protocol-command-decode | web-application-activity | web-application-attack | misc-activity | misc-attack | icmp-event | inappropriate-content | policy-violation | default-login-attempt }**	• not-suspicious – неподозрительный трафик. • unknown – неизвестный трафик. • bad-unknown – потенциально плохой трафик. • attempted-recon – попытка утечки информации. • successful-recon-limited – утечка информации. • successful-recon-largescale – масштабная утечка информации. • attempted-dos – попытка отказа в обслуживании. • successful-dos – отказ в обслуживании. • attempted-user – попытка получения привилегий пользователя. • unsuccessful-user – безуспешная попытка получения привилегий пользователя. • successful-user – успешная попытка получения привилегий пользователя. • attempted-admin – попытка получения привилегий администратора. • successful-admin – успешная попытка получения привилегий администратора. • rpc-portmap-decode – декодирование запроса RPC. • shellcode-detect – обнару жен исполняемый код. • string-detect – обнаружена подозрительная строка. • suspicious-filename-detect – было обнаружено подозрительное имя-файла.

Шаг	Описание	Команда	Ключи
			• suspicious-login – была обнаружена попытка входа с использованием подозрительного имени пользователя. • system-call-detect – обнаружен системный вызов. • tcp-connection – обнаружено TCP-соединение. • trojan-activity – был обнаружен сетевой троян. • unusual-client-port-connection – клиент использовал необычный порт. • network-scan – обнаружение сетевого сканирования. • denial-of-service – обнаружение атаки отказа в обслуживании. • non-standard-protocol – обнаружение нестандартного протокола или события. • protocol-command-decode – обнаружена попытка шифрования. • web-application-activity – доступ к потенциально уязвимому веб-приложению. • web-application-attack – атака на веб-приложение. • misc-activity – прочая активность. • misc-attack – прочие атаки. • icmp-event – общее событие ICMP. • inappropriate-content – обнаружено неприемлемое содержание. • policy-violation – потенциальное нарушение корпоративной конфиденциальности.

Шаг	Описание	Команда	Ключи
			• default-login-attempt – попытка входа с помощью стандартного логина/пароля.
14	Установить значение кода DSCP, для которого должно срабатывать правило (необязательно).	**esr(config-ips-category-rule)# ip dscp <DSCP>**	<DSCP> – значение кода DSCP, принимает значения [0..63].
15	Установить значение времени жизни пакета (TTL), для которого должно срабатывать правило (необязательно).	**esr(config-ips-category-rule)# ip ttl <TTL>**	<TTL> – значение TTL, принимает значения в диапазоне [1..255].
16	Установить номер IP-протокола, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol any.	**esr(config-ips-category-rule)# ip protocol-id <ID>**	<ID> – идентификационный номер IP-протокола, принимает значения [1..255].
17	Установить значения ICMP CODE, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol icmp.	**esr(config-ips-category-rule)# ip icmp code <CODE>**	<CODE> – значение CODE протокола ICMP, принимает значение в диапазоне [0..255].
		esr(config-ips-category-rule)# ip icmp code comparison-operator { greater-than | less-than }	Оператор сравнения для значения ip icmp code: • greater-than – больше чем.. • less-than – меньше чем..
18	Установить значения ICMP ID, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol icmp.	**esr(config-ips-category-rule)# ip icmp id <ID>**	<ID> – значение ID протокола ICMP, принимает значение в диапазоне [0.. 65535].
19	Установить значения ICMP Sequence-ID, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol icmp.	**esr(config-ips-category-rule)# ip icmp sequence-id <SEQ-ID>**	<SEQ-ID> – значение Sequence-ID протокола ICMP, принимает значение в диапазоне [0.. 4294967295].
20	Установить значения ICMP TYPE, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol icmp.	**esr(config-ips-category-rule)# ip icmp type <TYPE>**	<TYPE> – значение TYPE протокола ICMP, принимает значение в диапазоне [0..255].

Шаг	Описание	Команда	Ключи
		esr(config-ips-category-rule)# ip icmp type comparison-operator { greater-than | less-than }	Оператор сравнения для значения ip icmp type: • greater-than – больше чем.. • less-than – меньше чем..
21	Установить значения TCP Acknowledgment-Number, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol tcp.	**esr(config-ips-category-rule)# ip tcp acknowledgment-number <ACK-NUM>**	<ACK-NUM> – значение Acknowledgment-Number протокола TCP, принимает значение в диапазоне [0.. 4294967295].
22	Установить значения TCP Sequence-ID, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol tcp.	**esr(config-ips-category-rule)# ip tcp sequence-id <SEQ-ID>**	<SEQ-ID> – значение Sequence-ID протокола TCP, принимает значение в диапазоне [0.. 4294967295].
23	Установить значения TCP Window-Size, для которого должно срабатывать правило (необязательно). Применимо только для значения protocol tcp.	**esr(config-ips-category-rule)# ip tcp window-size <SIZE>**	<SIZE> – значение Window-Size протокола TCP, принимает значение в диапазоне [0.. 65535].
24	Установить ключевые слова протокола HTTP, для которых должно срабатывать правило (необязательно). Применимо только для значения protocol http.	**esr(config-ips-category-rule)# ip http { accept | accept-enc | accept-lang | client-body | connection | content-type | cookie | file-data | header | header-names | host | method | protocol | referer | request-line | response-line | server-body | start | start-code | start-msg | uri | user-agent }**	Значение ключевых слов см. в документации Suricata 4.X. https://suricata.readthedocs.io/en/suricata-4.1.4/rules/http-keywords.html
25	Установить значение ключевого слова URI LEN протокола HTTP, для которых должно срабатывать правило (необязательно). Применимо только для значения protocol http.	**esr(config-ips-category-rule)# ip http urilen <LEN>**	<LEN> – принимает значение в диапазоне [0.. 65535].
		esr(config-ips-category-rule)# ip http urilen comparison-operator { greater-than | less-than }	Оператор сравнения для значения ip http urilen: • greater-than – больше чем.. • less-than – меньше чем..

Шаг	Описание	Команда	Ключи
26	Установить значение содержимого пакетов (Payload content), для которых должно срабатывать правило (необязательно).	**esr(config-ips-category-rule)# payload content <CONTENT>**	<CONTENT> – текстовое сообщение, задаётся строкой до 1024 символов.
27	Не различать прописные и заглавные буквы в описании содержимого пакетов (необязательно). Применимо только совместно с командой payload content.	**esr(config-ips-category-rule)# payload no-case**	
28	Установить, сколько байтов с начала содержимого пакета будет проверено (необязательно). Применимо только совместно с командой payload content.	**esr(config-ips-category-rule)# payload depth <DEPTH>**	<DEPTH> – число байт с начала содержимого пакета, принимает значение в диапазоне [1.. 65535]. По умолчанию проверяется все содержимое пакета.
29	Установить число байт смещения от начала содержимого пакета для проверки (необязательно). Применимо только совместно с командой payload content.	**esr(config-ips-category-rule)# payload offset <OFFSET>**	<OFFSET> – число байт смещения от начала содержимого пакета, принимает значение в диапазоне [1.. 65535]. По умолчанию проверяется с начала содержимого.
30	Установить размер содержимого пакетов, для которых должно срабатывать правило (необязательно).	**esr(config-ips-category-rule)# payload data-size <SIZE>**	<SIZE> – размер содержимого пакетов, принимает значение в диапазоне [0.. 65535].
		esr(config-ips-category-rule)# payload data-size comparison-operator { greater-than | less-than }	Оператор сравнения для значения payload data-size: • greater-than – больше чем.. • less-than – меньше чем.
31	Указать пороговое значение количества пакетов, при котором сработает правило (необязательно).	**esr(config-ips-category-rule)# threshold count <COUNT>**	<COUNT> – число пакетов, принимает значение в диапазоне [1.. 65535].
32	Указать интервал времени, для которого считается пороговое количество пакетов. (Обязательно, если включен threshold count).	**esr(config-ips-category-rule)# threshold second <SECOND>**	<SECOND> – интервал времени в секундах, принимает значение в диапазоне [1.. 65535].

Шаг	Описание	Команда	Ключи
33	Указать по адресу отправителя или получателя будут считаться пороги. (Обязательно, если включен threshold count).	**esr(config-ips-category-rule)# threshold track { by-src | by-dst }**	• by-src – считать пороговое значение для пакетов с одинаковым IP-отправителя. • by-dst – считать пороговое значение для пакетов с одинаковым IP-получателя.
34	Указать метод обработки пороговых значений.	**esr(config-ips-category-rule)# threshold type {threshold | limit | both }**	• threshold – выдавать сообщение каждый раз по достижении порога. • limit – выдавать сообщение не чаще <COUNT> раз за интервал времени <SECOND>. • both – комбинация threshold и limit. Сообщение будет генерироваться, если в течении интервала времени <SECOND> было <COUNT> или более пакетов подходящих под условия правила, и сообщение будет отправлено только один раз в течении интервала времени <SECOND>.
35	Активировать правило.	**esr(config-ips-category-rule)# enable**	

16.9.6 Пример настройки базовых пользовательских правил

Задача:

Написать правило для защиты сервера с IP 192.168.1.10 от DOS-атаки ICMP-пакетами большого размера.

Решение:

Создадим набор пользовательских правил:

```
esr(config)# security ips-category user-defined USER
```

Создадим правило для защиты от атаки:

```
esr(config-ips-category)# rule 10
esr(config-ips-category-rule)# description "Big ICMP DoS"
```

Будем отбрасывать пакеты:

```
esr(config-ips-category-rule)# action drop
```

Настроим сообщение об атаке:

```
esr(config-ips-category-rule)# meta log-message "Big ICMP DoS"
esr(config-ips-category-rule)# meta classification-type successful-dos
```

Укажем тип протокола для правила:

```
esr(config-ips-category-rule)# protocol icmp
```

Так как был указан протокол icmp, то в качестве порта отправителя и получателя требуется указать any:

```
esr(config-ips-category-rule)# source-port any
esr(config-ips-category-rule)# destination-port any
```

В качестве адреса получателя укажем используемый сервер:

```
esr(config-ips-category-rule)# destination-address ip 192.168.1.10
```

Атакующий может отправлять пакеты с любого адреса:

```
esr(config-ips-category-rule)# source-address any
```

Зададим направление трафика:

```
esr(config-ips-category-rule)# direction one-way
```

Правило будет срабатывать на пакеты размером больше 1024 байт:

```
esr(config-ips-category-rule)# payload data-size 1024
esr(config-ips-category-rule)# payload data-size comparison-operator greate r-than
```

Правило будет срабатывать, если нагрузка на сервер будет превышать 3 Мбит/с, при этом сообщение об атаке будет генерироваться не чаще одного раза в минуту:

```
3 Мб/с = 3145728 бит в сек
Пакет размером 1Кбайт = 8192 бита
3145728 / 8192 = 384 пакета в сек
384 * 60 = 23040 пакетов в минуту
```

```
esr(config-ips-category-rule)# threshold count 23040
esr(config-ips-category-rule)# threshold second 60
esr(config-ips-category-rule)# threshold track by-dst
esr(config-ips-category-rule)# threshold type both
```

Активизируем правило:

```
esr(config-ips-category-rule)# enable
```

16.9.7 Алгоритм настройки расширенных пользовательских правил

Шаг	Описание	Команда	Ключи
1	Задать имя и перейти в режим конфигурирования набора пользовательских правил.	**esr(config)# security ips-category user-defined <WORD>**	<WORD> – имя набора пользовательских правил, задаётся строкой до 32 символов.
2	Задать описание набора пользовательских правил (необязательно).	**esr(config-ips-category)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.
3	Создать расширенное правило и перейти в режим его конфигурирования.	**esr(config-ips-category)# rule-advanced <SID>**	<SID> – номер правила, принимает значения [1.. 4294967295].
4	Задать описание правила (необязательно).	**esr(config-ips-category-rule-advanced)# description <DESCRIPTION>**	<DESCRIPTION> – описание задаётся строкой до 255 символов.
5	Указать действие данного правила.	**esr(config-ips-category-rule-advanced)# rule-text <LINE>**	<CONTENT> – текстовое сообщение в формате SNORT 2.X / Suricata 4.X, задаётся строкой до 1024 символов. При написании правил в тексте правила необходимо использовать только двойные кавычки (символ "), а само правило необходимо заключать в одинарные кавычки (символ ').
6	Активировать правило.	**esr(config-ips-category-rule-advanced)# enable**	

16.9.8 Пример настройки расширенных пользовательских правил

Задача:

Написать правило, детектирующее атаку типа Slowloris.

Решение:

Создадим набор пользовательских правил:

```
esr(config)# security ips-category user-defined ADV
```

Создадим расширенное правило:

```
esr(config-ips-category)# rule-advanced 1
esr(config-ips-category-rule-advanced)# description "Slow Loris rule 1"
esr(config-ips-category-rule-advanced)# rule-text 'alert tcp any any -> any 80 (msg:"Possible
Slowloris Attack Detected"; flow:to_server,established; content:"X-a|3a|"; distance:0; pcre:"/
\d\d\d\d/"; distance:0; content:"|0d 0a|"; sid:10000001;)'
esr(config-ips-category-rule-advanced)# enable
esr(config-ips-category-rule-advanced)# exit
```

Создадим ещё одно расширенное правило, работающее по схожему алгоритму, чтобы определить, какое из правил будет эффективнее:

```
esr(config-ips-category)# rule-advanced 2
esr(config-ips-category-rule-advanced)# description "Slow Loris rule 2"
esr(config-ips-category-rule-advanced)# rule-text 'alert tcp $EXTERNAL_NET any -> $HOME_NET
$HTTP_PORTS (msg:"SlowLoris.py DoS attempt"; flow:established,to_server,no_stream; content:"X-
a:"; dsize:<15; detection_filter:track by_dst, count 3, seconds 30; classtype:denial-of-
service; sid: 10000002; rev:1; )'
esr(config-ips-category-rule-advanced)#
```

17 Управление сертификатами и ключами

- Настройка криптопрофилей
- Автоматическое распространение ключей и сертификатов X.509
  - Общее описание инфраструктуры открытых ключей
  - Планирование инфраструктуры открытых ключей
  - Настройка PKI-сервера в роли корневого удостоверяющего центра
    - Алгоритм настройки
    - Пример настройки
  - Настройка PKI-клиента
    - Алгоритм настройки
    - Пример настройки PKI-клиента для подключения к корневому удостоверяющему центру
  - Процесс автоматического перевыпуска сертификата PKI-клиента
  - Процесс автоматического перевыпуска сертификата PKI-сервера
- Ручная генерация и распространение ключей и сертификатов X.509
  - Алгоритм генерации ключей и запросов на сертификацию
  - Пример ручного выпуска сертификата через внешний удостоверяющий центр

17.1 Настройка криптопрофилей

Криптопрофили (crypto profile) используются для настройки параметров безопасности, шифрования и аутентификации сигнального (SIP), медиа-трафика (RTP) и https-трафика.

При использовании протоколов TLS или WebSocket Secure (WSS) в качестве SIP-транспорта, а также протокола DTLS-SRTP для шифрования RTP-трафика, возможно использование сертификатов, автоматически сгенерированных самим ESBC, сгенерированных по требованию пользователя на ESBC или загруженных пользователем. По умолчанию используются сертификаты, автоматически сгенерированные самим ESBC, дополнительных настроек для их использования не требуется. Для проксирования https-трафика использование криптопрофиля является необходимым.

Для генерации сертификатов и ключей средствами ESBC используется команда *crypto generate*. Подробное описание команд для генерации сертификатов и ключей на ESBC приведено в разделе Управление ключами и сертификатами.

Для загрузки сертификатов и ключей на устройство через CLI используется команда *copy*. Пример команды для копирования через протокол tftp:

```
vesbc# copy tftp://10.0.0.1:/ca.crt crypto:cert/ca.crt
```

> ⚠ Путь для сохранения сертификатов ca и cert — crypto:cert/
> Путь для сохранения private-key — crypto:private-key/

Для управления пользовательскими сертификатами и версией TLS используется crypto profile.

Описание всех команд настройки криптопрофиля приведено в разделе Настройки криптопрофиля.

Пример настройки crypto profile:

```
vesbc#
vesbc# configure
vesbc(config)# esbc
vesbc(config-esbc)# crypto profile CRYPTO-PROFILE

#Установка сертификата удостоверяющего центра (CA certificate):
vesbc(config-esbc-crypto-profile)# ca default_ca.pem

#Установка клиентского сертификата (X.509):
vesbc(config-esbc-crypto-profile)# cert default_cert.pem

#Установка private-key:
vesbc(config-esbc-crypto-profile)# private-key default_cert_key.pem

#Установка пароля private-key (необязательно):
vesbc(config-esbc-crypto-profile)# password private-key PASSWORD

#Установка минимальной и максимальной версии TLS (необязательно):
vesbc(config-esbc-crypto-profile)# tls min 1.1
vesbc(config-esbc-crypto-profile)# tls max 1.2
vesbc(config-esbc-crypto-profile)# exit
```

Если не устанавливать версии TLS, то при установлении соединения будет использоваться любая версия 1.0–1.3.
Настройки *tls min* и *tls max* используются только при применении crypto profile для SIP-транспорта и проксировании https-трафика, и не используются для шифрования DTLS-SRTP при использовании crypto-profile в медиапрофиле.

Для того чтобы использовать crypto profile для SIP-транспорта, необходимо его указать в настройках нужного транспорта:

```
vesbc(config-esbc)# sip transport SIP-TRANSPORT
vesbc(config-esbc-sip-transport)# crypto profile CRYPTO-PROFILE
vesbc(config-esbc-sip-transport)# exit
```

Настройки crypto profile будут использоваться, только если выбран режим работы SIP-транспорта tls или wss, для остальных режимов настройки игнорируются.

Для того чтобы использовать crypto profile для шифрования DTLS-SRTP, необходимо его указать в настройках медиапрофиля:

```
vesbc(config-esbc)# media profile MEDIA-PROFILE
vesbc(config-esbc-media-profile)# crypto profile CRYPTO-PROFILE
vesbc(config-esbc-media-profile)# exit
```

Настройки crypto profile будут использоваться, только если выбран режим шифрования srtp keying dtls-srtp, для остальных режимов настройки игнорируются.

⚠ В текущей версии ПО проверка сертификатов командами **verify client-cert** и **verify server-cert** применима только для использования криптопрофиля в конфигурации HTTPS-прокси сервера. При использовании криптопрофиля в конфигурации **sip transport** и **media profile** ESBC проверки сертификатов не осуществляется.

Пример использования crypto profile:

Задача:

Использовать сертификат, загруженный пользователем на ESBC, для абонентских подключений по tls версии 1.3 и шифрования медиа DTLS-SRTP.

Решение:

1. Выполнить базовую настройку ESBC для обеспечения маршрутизации абонентских подключений в сторону ECSS-10:

```
vesbc(config)# esbc
vesbc(config-esbc)# media resource USERS
vesbc(config-esbc-media-resource)# ip address 20.0.0.1
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# media resource ECSS
vesbc(config-esbc-media-resource)# ip address 192.168.1.1
vesbc(config-esbc-media-resource)# exit
vesbc(config-esbc)# sip transport USERS
vesbc(config-esbc-sip-transport)# ip address 20.0.0.1
vesbc(config-esbc-sip-transport)# port 5061
vesbc(config-esbc-sip-transport)# mode tls
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# sip transport ECSS
vesbc(config-esbc-sip-transport)# ip address 192.168.1.1
vesbc(config-esbc-sip-transport)# port 5070
vesbc(config-esbc-sip-transport)# exit
vesbc(config-esbc)# route-table TO_ECSS
vesbc(config-esbc-route-table)# rule 0
vesbc(config-esbc-route-table-rule)# action direct-to-trunk ECSS
vesbc(config-esbc-route-table-rule)# exit
vesbc(config-esbc-route-table)# exit
vesbc(config-esbc)# trunk sip ECSS
vesbc(config-esbc-trunk-sip)# sip transport ECSS
vesbc(config-esbc-trunk-sip)# media resource 0 ECSS
vesbc(config-esbc-trunk-sip)# remote address 192.168.1.2
vesbc(config-esbc-trunk-sip)# remote port 5070
vesbc(config-esbc-trunk-sip)# exit
vesbc(config-esbc)# user-interface sip USERS
vesbc(config-esbc-user-interface-sip)# sip transport USERS
vesbc(config-esbc-user-interface-sip)# route-table TO_ECSS
vesbc(config-esbc-user-interface-sip)# media resource 0 USERS
vesbc(config-esbc-user-interface-sip)# exit
```

2. Загрузить файлы сертификата, CA и private-key на ESBC через CLI (в примере указан протокол tftp):

```
#Загрузка CA сертификата:
vesbc# copy tftp://10.0.0.1:/ca.crt crypto:cert/ca.crt

#Загрузка клиентского сертификата:
vesbc# copy tftp://10.0.0.1:/cert.crt crypto:cert/cert.crt

#Загрузка private-key:
vesbc# copy tftp://10.0.0.1:/key.pem crypto:private-key/key.pem
```

3. Создать crypto profile, указать в нем файлы сертификатов, private-key и версию TLS:

```
vesbc(config-esbc)# crypto profile CRYPTO_PROFILE
vesbc(config-esbc-crypto-profile)# ca ca.crt
vesbc(config-esbc-crypto-profile)# cert cert.crt
vesbc(config-esbc-crypto-profile)# private-key key.pem
vesbc(config-esbc-crypto-profile)# tls min 1.3
vesbc(config-esbc-crypto-profile)# tls max 1.3
vesbc(config-esbc-crypto-profile)# exit
```

4. Создать медиапрофиль для использования DTLS-SRTP и привязать к нему crypto profile:

```
vesbc(config-esbc)# media profile MP_USERS
vesbc(config-esbc-media-profile)# srtp mode mandatory
vesbc(config-esbc-media-profile)# srtp keying dtls-srtp
vesbc(config-esbc-media-profile)# crypto profile CRYPTO_PROFILE
vesbc(config-esbc-media-profile)# exit
```

5. Привязать media profile MP_USERS к user-interface sip USERS:

```
vesbc(config-esbc)# user-interface sip USERS
vesbc(config-esbc-user-interface-sip)# media profile MP_USERS
vesbc(config-esbc-user-interface-sip)# exit
```

6. Привязать crypto profile CRYPTO_PROFILE к sip transport USERS:

```
vesbc(config-esbc)# sip transport USERS
vesbc(config-esbc-sip-transport)# crypto profile CRYPTO_PROFILE
vesbc(config-esbc-sip-transport)# exit
```

7. Применить настройки:

```
vesbc(config-esbc)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
vesbc(config-esbc)# do confirm
Configuration has been confirmed. Commit timer canceled.
```

17.2 Автоматическое распространение ключей и сертификатов X.509

17.2.1 Общее описание инфраструктуры открытых ключей

Инфраструктура открытых ключей (Public Key Infrastructure, PKI) – комплекс средств, мер и политик, обеспечивающих работу систем на базе алгоритмов шифрования с открытым ключом.

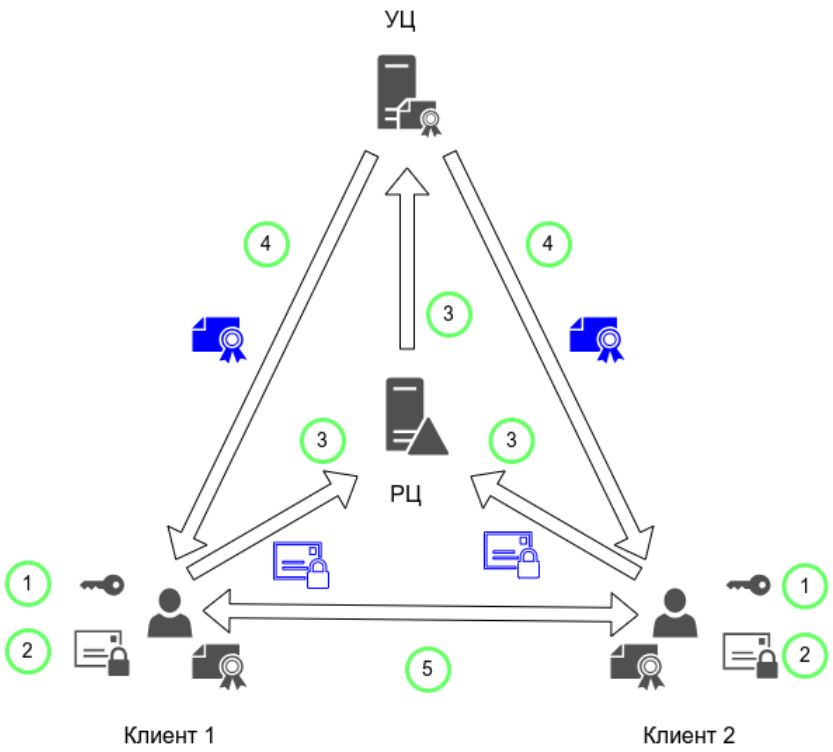
К участникам PKI обычно относятся:

- Удостоверяющий центр (УЦ),
- Регистрационный центр (РЦ),
- Клиент.

В процессе взаимодействия участников PKI появляются и используются следующие объекты:

- Приватный ключ,
- Публичный ключ,
- Запрос на сертификацию,
- Сертификат,
- Список отозванных сертификатов.

Схематично основные процедуры в инфраструктуре открытых ключей можно описать следующим образом:



1. Клиент генерирует у себя приватный ключ.
2. Клиент формирует запрос на сертификацию, включающий в себя публичный ключ, парный к раннее сгенерированному приватному ключу, информацию о владельце приватного ключа и запрашиваемые опции, которые затем могут быть добавлены в выпускаемый сертификат.
3. Клиент доставляет запрос на сертификацию в удостоверяющий центр. При наличии в структуре PKI регистрационного центра запрос от клиента будет приходить на него и после валидации будет уходить в удостоверяющий центр.
4. Удостоверяющий центр на основании клиентского запроса на сертификацию выпишет сертификат, содержащий всю клиентскую информацию и информацию о самом УЦ. Сам сертификат будет подписан приватным ключом УЦ, тем самым данные в клиентском сертификате не смогут быть изменены третьей стороной. От УЦ сертификат передается клиенту.
5. Теперь клиенты могут, используя свой приватный ключ, подписывать данные в рамках используемых процессов (в нашем случае – сетевых протоколов) при отправке и валидировать данные при помощи публичного сертификата на приёме.

17.2.2 Планирование инфраструктуры открытых ключей

Сервисные маршрутизаторы ESR могут принимать участие в структуре PKI в качестве корневого удостоверяющего центра и клиента PKI. Перед развертыванием инфраструктуры PKI крайне важно её спланировать для обеспечения безопасности и стабильной работы.

Основные рекомендации по планированию инфраструктуры открытых ключей:

1. Определите роли устройств в структуре PKI.
2. Определите схему именования сертификатов.
3. Определите политику доступа к центру сертификации и механизмы ограничения доступа к нему от недоверенных клиентов.
4. Обеспечьте актуальность времени на всех хостах, используемых в структуре PKI.

17.2.3 Настройка PKI-сервера в роли корневого удостоверяющего центра

Базовый вариант настройки PKI-сервера – корневой удостоверяющий центр, работающий на самоподписанных сертификатах, который напрямую обслуживает запросы конечных клиентов.

> Изменение отличительного имени сертификата удостоверяющего центра приводит к перегенерации приватного ключа и сертификата PKI-сервера и очистке базы выписанных сертификатов. Требуется заранее, на этапе планирования структуры PKI, определиться с идентификацией владельца удостоверяющего центра и после ввода удостоверяющего центра в эксплуатацию не менять эти настройки.

Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Настроить NTP-клиент согласно разделу Алгоритм настойки NTP. Маршрутизатор должен получать точное время от NTP-сервера или NTP-пира, либо доверять локально настроенному времени с включенным режимом NTP Master.		
2	Перейти в режим настройки PKI-клиента, который будет использоваться в настройках PKI-сервера как источник самоподписанных сертификатов.	**esr(config)# crypto pki trustpoint <TRUSTPOINT>**	<TRUSTPOINT> – имя PKI-клиента, задаётся строкой от 1 до 31 символа.
3	Перейти в режим настройки отличительного имени сертификата – набора атрибутов, уникально описывающих удостоверяющий центр.	**esr(config-trustpoint)# subject-name**	
4	Указать код страны (необязательно).	**esr(config-trustpoint-subject-name)# country <COUNTRY>**	<COUNTRY> – код страны, задаётся строкой длиной 2 символа. Рекомендуется использовать двухбуквенные обозначения стран "alpha-2" из стандарта ISO 3166-1.
5	Указать название штата, области или провинции (необязательно).	**esr(config-trustpoint-subject-name)# state <STATE>**	<STATE> – название штата, области или провинции, задаётся строкой от 1 до 128 символов.

Шаг	Описание	Команда	Ключи
6	Указать название населенного пункта или его территориальной единицы (необязательно).	**esr(config-trustpoint-subject-name)# locality <LOCATION>**	<LOCATION> – название населенного пункта или его территориальной единицы, задаётся строкой от 1 до 128 символов.
7	Указать название организации (необязательно).	**esr(config-trustpoint-subject-name)# organization <ORGANIZATION>**	<ORGANIZATION> – название организации, задаётся строкой от 1 до 64 символов.
8	Указать подразделение организации (необязательно).	**esr(config-trustpoint-subject-name)# organization-unit <ORGANIZATION-UNIT>**	<ORGANIZATION-UNIT> – название подразделения организации, задаётся строкой от 1 до 64 символов.
9	Указать общее имя сертификата.	**esr(config-trustpoint-subject-name)# common-name <COMMON-NAME>**	<COMMON-NAME> – общее имя, задаётся строкой от 1 до 64 символов. Как правило в качестве общего имени сертификата удостоверяющего центра используется имя домена, который обслуживает удостоверяющий центр или юридическое название удостоверяющего центра.
10	Перейти в режим настройки альтернативных имен сертификата (необязательно).	**esr(config-trustpoint)# subject-alt-name**	
11	Указать IP-адрес в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# ipv4 <ADDR>**	<ADDR> – IP-адрес, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255].
		esr(config-trustpoint-san)# ipv6 <IPV6-ADDR>	<IPV6-ADDR> – IPv6-адрес, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF].
12	Указать полное доменное имя хоста в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# dns <NAME>**	<NAME> – полное доменное имя хоста (FQDN), задается строкой от 1 до 235 символов. Пример записи доменного имени – router.example.loc.

Шаг	Описание	Команда	Ключи
13	Указать адрес электронной почты в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# email <EMAIL>**	<EMAIL> – адрес электронной почты, задается строкой от 6 до 254 символов. Пример записи электронной почты – router@example.loc.
14	Перейти в режим настройки механизма выпуска сертификата. Для PKI-клиента, выпускающего самоподписанные сертификаты укажем тип "selfsigned".	**esr(config-trustpoint)# enrollment selfsigned**	
15	Указать время жизни самоподписанных сертификатов (необязательно).	**esr(config-trustpoint-enrollment-selfsigned)# lifetime <DAYS>**	<DAYS> – количество дней, принимает значения в диапазоне [1..3650]. Значение по умолчанию: 3650.
16	Указать назначение генерируемых сертификатов. Для PKI-клиента, используемого в настройках PKI-сервера в качестве источника самоподписанных сертификатов необходимо указать значение "root-ca".	**esr(config-trustpoint)# usage root-ca**	
17	Указать экземпляр VRF, в котором будет работать PKI-клиент (необязательно). Если планируется работа PKI-сервера в определенном VRF, PKI-клиент с самоподписанными сертификатами также требуется вынести в этот VRF.	**esr(config-trustpoint)# ip vrf forwarding <VRF>**	<VRF> – имя VRF, задается строкой до 31 символа.
18	Включить PKI-клиент.	**esr(config-trustpoint)# enable**	
19	Перейти в режим настройки PKI-сервера.	**esr(config)# crypto pki server**	
20	Указать имя раннее настроенного PKI-клиента, который будет использоваться в настройках PKI-сервера как источник самоподписанных сертификатов.	**esr(config-pki-server)# trustpoint <TRUSTPOINT>**	<TRUSTPOINT> – имя PKI-клиента, задаётся строкой от 1 до 31 символа.
21	Указать экземпляр VRF, в котором будет работать PKI-сервер (необязательно).	**esr(config-pki-server)# ip vrf forwarding <VRF>**	<VRF> – имя VRF, задается строкой до 31 символа.

Шаг	Описание	Команда	Ключи
22	Указать IP-адрес или сетевой интерфейс, который будет будет прослушиваться PKI-сервером для обработки входящих запросов (необязательно). В случае, если настройка не будет указана PKI-сервер будет принимать запросы на всех настроенных IP-интерфейсах маршрутизатора.	**esr(config-pki-server)# source-address <ADDR>**	<ADDR> – IP-адрес, назначенный на локальном сетевом интерфейсе маршрутизатора, на котором PKI-сервер будет слушать входящие подключения, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255].
		esr(config-pki-server)# source-interface <IF>	<IF> – интерфейс или туннель, на котором PKI-сервер будет слушать входящие подключения.
23	Указать challenge-password, который будет использован для аутентификации PKI-клиентов (необязательно).	**esr(config-pki-server)# challenge-password { <CLEAR-TEXT> | encrypted <ENCRYPTED-TEXT> }**	<CLEAR-TEXT> – пароль, задаётся строкой от 8 до 32 символов; <ENCRYPTED-TEXT> – зашифрованный пароль размером от 8 байт до 32 байт (от 16 до 64 символов) в шестнадцатеричном формате (0xYYYY...) или (YYYY...).
24	Указать время жизни выписываемых клиентам сертификатов в днях (необязательно).	**esr(config-pki-server)# lifetime <DAYS>**	<DAYS> – количество дней, принимает значения в диапазоне [1..3650]. Значение по умолчанию: 365.
25	Включить PKI-сервер.	**esr(config-pki-server)# enable**	

Пример настройки

Задача:

Настроить на маршрутизаторе PKI-сервер в роли корневого удостоверяющего центра. Время жизни сертификата удостоверяющего центра должно быть ограничено одним годом, а время жизни клиентских сертификатов должно быть ограничено двумя неделями.

В качестве начальной конфигурации на маршрутизаторе уже настроен сетевой интерфейс в сторону Интернет-провайдера, прописан шлюз по умолчанию и настроена зона безопасности. Шлюз Интернет-провайдера также может служить источником синхронизации времени по протоколу NTP.

```
hostname ESR.CA

security zone WAN
exit

interface gigabitethernet 1/0/1.103
  security-zone WAN
  ip address 10.0.103.2/30
exit

ip route 0.0.0.0/0 10.0.103.1
```

Решение:

Настроим NTP-клиент на получение точного времени от шлюза Интернет-провайдера:

```
ESR.CA(config)# ntp enable
ESR.CA(config)# ntp server 10.0.103.1
ESR.CA(config-ntp-server)# exit
ESR.CA(config)#
```

Применим конфигурацию на маршрутизаторе и дождемся успешной синхронизации времени по протоколу NTP:

> ⚠ Если произвести настройку подключения к NTP-серверу одновременно с настройкой PKI-сервера, сгенерированные на этапе запуска PKI-сервера служебные файлы и сертификаты могут содержать некорректные дату и время, что приведет к ошибкам в дальнейшей работе инфраструктуры PKI.

```
ESR.CA(config)# end
Warning: you have uncommitted configuration changes.
ESR.CA# commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
ESR.CA# confirm
Configuration has been confirmed. Commit timer canceled.
ESR.CA#
ESR.CA# sh ntp peers
Clock is synchronized, stratum 6, reference is 10.0.103.1
    remote       vrf   refid         st  t  when  poll  reach  delay   offset  jitter
-   ----------  -----  -----------   --  -  ----  ----  -----  ------  ------  ------
*   10.0.103.1         127.127.1.0   5   u  16    64    37     0.601   8.533   0.019
ESR.CA#
```

Перейдем к настройке PKI-клиента, который будет использоваться в качестве источника самоподписанных сертификатов для PKI-сервера:

```
ESR.CA(config)# crypto pki trustpoint ROOT_CA
ESR.CA(config-trustpoint)#
```

Перейдем в раздел настройки отличительного имени сертификата удостоверяющего центра. Настроим те атрибуты, которые позволят однозначно идентифицировать удостоверяющий центр:

```
ESR.CA(config-trustpoint)# subject-name
ESR.CA(config-trustpoint-subject-name)# country RU
ESR.CA(config-trustpoint-subject-name)# state Moscow
ESR.CA(config-trustpoint-subject-name)# locality Moscow
ESR.CA(config-trustpoint-subject-name)# organization Company
ESR.CA(config-trustpoint-subject-name)# common-name ca.company.loc
ESR.CA(config-trustpoint-subject-name)# exit
ESR.CA(config-trustpoint)#
```

Перейдем в раздел настройки альтернативных имен сертификата. Укажем в качестве альтернативных имен IP-адрес и полное доменное имя:

```
ESR.CA(config-trustpoint)# subject-alt-name
ESR.CA(config-trustpoint-san)# ipv4 10.0.103.2
ESR.CA(config-trustpoint-san)# dns ca.company.loc
ESR.CA(config-trustpoint-san)# exit
ESR.CA(config-trustpoint)#
```

Перейдем в режим настройки механизма выпуска сертификата. Для PKI-клиента, выпускающего самоподписанные сертификаты, укажем тип "selfsigned". В нем настроим время жизни самоподписанных сертификатов:

```
ESR.CA(config-trustpoint)# enrollment selfsigned
ESR.CA(config-trustpoint-enrollment-selfsigned)# lifetime 365
ESR.CA(config-trustpoint-enrollment-selfsigned)# exit
ESR.CA(config-trustpoint)#
```

Укажем назначение генерируемых сертификатов. Для PKI-клиента, используемого в настройках PKI-сервера, в качестве источника самоподписанных сертификатов необходимо указать значение "root-ca":

```
ESR.CA(config-trustpoint)# usage root-ca
ESR.CA(config-trustpoint)#
```

Включим PKI-клиент:

```
ESR.CA(config-trustpoint)# enable
ESR.CA(config-trustpoint)# exit
ESR.CA(config)#
```

Перейдем к настройке PKI-сервера:

```
ESR.CA(config)# crypto pki server
ESR.CA(config-pki-server)#
```

Укажем имя ранее настроенного PKI-клиента, который будет использоваться в настройках PKI-сервера как источник самоподписанных сертификатов:

```
ESR.CA(config-pki-server)# trustpoint ROOT_CA
ESR.CA(config-pki-server)#
```

Привяжем PKI-сервер к адресу сетевого интерфейса, смотрящего в сторону Интернет-провайдера:

```
ESR.CA(config-pki-server)# source-interface gi 1/0/1.103
ESR.CA(config-pki-server)#
```

Зададим challenge-password, для корректного обращения к удостоверяющему центру PKI-клиенты должны использовать правильный challenge-password.

```
ESR.CA(config-pki-server)# challenge-password StR0nnGP+ss
ESR.CA(config-pki-server)#
```

Включим PKI-сервер:

```
ESR.CA(config-pki-server)# enable
ESR.CA(config-pki-server)# exit
ESR.CA(config)#
```

Добавим пару зон безопасности и правило, разрешающее прохождение входящего на PKI-сервер трафика:

```
ESR.CA(config)# security zone-pair WAN self
ESR.CA(config-security-zone-pair)# rule 10
ESR.CA(config-security-zone-pair-rule)# description "Allow access to PKI-server from WAN"
ESR.CA(config-security-zone-pair-rule)# match protocol tcp
ESR.CA(config-security-zone-pair-rule)# match destination-port port-range 80
ESR.CA(config-security-zone-pair-rule)# match destination-address address-range 10.0.103.2
ESR.CA(config-security-zone-pair-rule)# action permit
ESR.CA(config-security-zone-pair-rule)# enable
ESR.CA(config-security-zone-pair-rule)# exit
ESR.CA(config-security-zone-pair)# exit
ESR.CA(config)#
```

Применим конфигурацию на маршрутизаторе:

```
ESR.CA(config)# end
Warning: you have uncommitted configuration changes.
ESR.CA# commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
ESR.CA# confirm
Configuration has been confirmed. Commit timer canceled.
ESR.CA#
```

В результате получим запущенный корневой удостоверяющий центр, готовый к обслуживанию клиентских запросов.

В команде **show crypto pki trustpoints** можно увидеть PKI-клиент ROOT_CA, готовый к работе, а добавив его имя в конце команды, можно более подробно рассмотреть его текущий статус:

```
ESR.CA# show crypto pki trustpoints
Name     Enrollment  Usage    Subject name                    Status   Next action
-------  ----------  -------  ------------------------------  -------  -------------------
ROOT_CA  selfsigned  root-ca  CN=ca.company.loc/O=Company/L=  Ready    2026-12-17 08:59:16
                                        Moscow/ST=Moscow/C=RU

ESR.CA# show crypto pki trustpoints ROOT_CA
Name:                ROOT_CA
Usage:               root-ca
Subject name:        CN=ca.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:          selfsigned
Status:              Ready
Renew date:          2026-12-17 08:59:16
CRL:                 Inactive
ESR.CA#
```

В команде **show crypto pki server** можно увидеть fingerprint сертификата удостоверяющего центра, который необходимо использовать в клиентах PKI совместно с установленным challenge-password для корректной авторизации:

```
ESR.CA# show crypto pki server
Status:                     Enabled
Associated trustpoint:      ROOT_CA
Lifetime days:              14
Certificate fingerprint:    A2:0C:10:C7:0E:06:3A:2F:EE:5F:C2:72:39:36:31:0A
Source:                     10.0.103.2
Last issued serial number: --
Challenge password:         Active
Shadow:                     Inactive
ESR.CA#
```

17.2.4 Настройка PKI-клиента

⚠ Изменение отличительного имени клиентского сертификата или URL подключения к PKI-серверу приводит к немедленному перезапросу нового сертификата у удостоверяющего центра.

Алгоритм настройки

Шаг	Описание	Команда	Ключи
1	Перейти в режим настройки PKI-клиента.	**esr(config)# crypto pki trustpoint <TRUSTPOINT>**	<TRUSTPOINT> – имя PKI-клиента, задаётся строкой от 1 до 31 символа.
2	Перейти в режим настройки отличительного имени сертификата – набора атрибутов, уникально описывающих владельца сертификата.	**esr(config-trustpoint)# subject-name**	
3	Указать код страны (необязательно).	**esr(config-trustpoint-subject-name)# country <COUNTRY>**	<COUNTRY> – код страны, задаётся строкой длиной 2 символа. Рекомендуется использовать двухбуквенные обозначения стран "alpha-2" из стандарта ISO 3166-1.
4	Указать название штата, области или провинции (необязательно).	**esr(config-trustpoint-subject-name)# state <STATE>**	<STATE> – название штата, области или провинции, задаётся строкой от 1 до 128 символов.
5	Указать название населенного пункта или его территориальной единицы (необязательно).	**esr(config-trustpoint-subject-name)# locality <LOCATION>**	<LOCATION> – название населенного пункта или его территориальной единицы, задаётся строкой от 1 до 128 символов.
6	Указать название организации (необязательно).	**esr(config-server-subject-name)# organization <ORGANIZATION>**	<ORGANIZATION> – название организации, задаётся строкой от 1 до 64 символов.
7	Указать подразделение организации (необязательно).	**esr(config-trustpoint-subject-name)# organization-unit <ORGANIZATION-UNIT>**	<ORGANIZATION-UNIT> – название подразделения организации, задаётся строкой от 1 до 64 символов.

Шаг	Описание	Команда	Ключи
8	Указать общее имя сертификата.	**esr(config-trustpoint-subject-name)# common-name <COMMON-NAME>**	<COMMON-NAME> – общее имя, задаётся строкой от 1 до 64 символов. Как правило в качестве общего имени клиентского сертификата используется полное доменное имя хоста, который использует данный клиентский сертификат или ФИО пользователя, использующего клиентский сертификат.
11	Перейти в режим настройки альтернативных имен сертификата (необязательно).	**esr(config-trustpoint)# subject-alt-name**	
12	Указать IP-адрес в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# ipv4 <ADDR>**	<ADDR> – IP-адрес, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255].
		esr(config-trustpoint-san)# ipv6 <IPV6-ADDR>	<IPV6-ADDR> – IPv6-адрес, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF].
13	Указать полное доменное имя хоста в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# dns <NAME>**	<NAME> – полное доменное имя хоста (FQDN), задается строкой от 1 до 235 символов. Пример записи доменного имени – router.example.loc.
14	Указать адрес электронной почты в качестве альтернативного имени сертификата (необязательно).	**esr(config-trustpoint-san)# email <EMAIL>**	<EMAIL> – адрес электронной почты, задается строкой от 6 до 254 символов. Пример записи электронной почты – router@example.loc.
14	Перейти в режим настройки механизма выпуска сертификата. Для PKI-клиента, выпускающего сертификаты от вышестоящего удостоверяющего центра по протоколу SCEP необходимо указать значение "scep".	**esr(config-trustpoint)# enrollment scep**	

Шаг	Описание	Команда	Ключи
15	Указать URL для подключения к PKI-серверу.	**esr(config-trustpoint-enrollment-scep)# url <URL>**	<URL> – URL для подключения к PKI-серверу, задается в виде "http:// <ADDR>[:<PORT>]/", где: • <ADDR> – IP-адрес или доменное имя PKI-сервера; • <PORT> – порт, на котором запущен PKI-сервер, в случае если используется 80 порт по умолчанию, то настройку можно пропустить. В случае если PKI-сервер расположен на том же маршрутизаторе, на котором настраивается PKI-клиент необходимо указать в URL любой из IP-интерфейсов, которые слушает PKI-сервер.
16	Указать цифровой отпечаток сертификата PKI-сервера, к которому выполняется подключение.	**esr(config-trustpoint-enrollment-scep)# fingerprint <FINGERPRINT>**	<FINGERPRINT> – значение цифровой отпечаток сертификата, полученного при помощи алгоритма MD5. Имеет размер 16 байт и задается в виде HEX-строки длиной 32 символа без разделителей (YYYY...) или 47 символов с двоеточием в роли разделителя (YY:YY:YY...).
17	Указать challenge-password, который будет использован для аутентификации на PKI-сервере (необязательно).	**esr(config-trustpoint-enrollment-scep)# challenge-password { <CLEAR-TEXT> | encrypted <ENCRYPTED-TEXT> }**	<CLEAR-TEXT> – пароль, задаётся строкой от 8 до 32 символов; <ENCRYPTED-TEXT> – зашифрованный пароль размером от 8 байт до 32 байт (от 16 до 64 символов) в шестнадцатеричном формате (0xYYYY...) или (YYYY...).
16	Указать назначение генерируемых сертификатов. Для PKI-клиента, используемого в качестве источника ключевой информации в IPsec-туннелях, необходимо указать значение "ipsec".	**esr(config-trustpoint)# usage ipsec**	

Шаг	Описание	Команда	Ключи
9	Указать экземпляр VRF, в котором будет работать PKI-клиент (необязательно).	**esr(config-trustpoint)# ip vrf forwarding <VRF>**	<VRF> – имя VRF, задается строкой до 31 символа.
10	Указать IP-адрес или сетевой интерфейс, с которого PKI-клиент будет выполнять запросы к удостоверяющему центру (необязательно).	**esr(config-trustpoint)# source-address <ADDR>**	<ADDR> – IP-адрес, назначенный на локальном сетевом интерфейсе маршрутизатора, с которого PKI-клиент будет выполнять запросы, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255].
		esr(config-trustpoint)# source-interface <IF>	<IF> – интерфейс или туннель, с которого PKI-клиент будет выполнять запросы.
18	Включить PKI-клиент.	**esr(config-trustpoint)# enable**	

Пример настройки PKI-клиента для подключения к корневому удостоверяющему центру

Задача:

Настроить на маршрутизаторе PKI-клиент так, чтобы он успешно подключался к удостоверяющему центру. В качестве удостоверяющего центра рассмотрим полностью настроенный PKI-сервер из предыдущего примера. Помимо настройки отличительного имени сертификата, настроим в качестве альтернативного имени IP-адрес маршрутизатора и его полное доменное имя.

В качестве начальной конфигурации на маршрутизаторе уже настроен сетевой интерфейс в сторону Интернет-провайдера, прописан шлюз по умолчанию и настроена зона безопасности. Шлюз Интернет-провайдера также может служить источником синхронизации времени по протоколу NTP.

```
hostname ESR.CA

security zone WAN
exit

interface gigabitethernet 1/0/1.103
  security-zone WAN
  ip address 10.0.103.2/30
exit

security zone-pair WAN self
  rule 10
    description "Allow access to PKI-server from WAN"
    action permit
    match protocol tcp
    match destination-address address-range 10.0.103.2
    match destination-port port-range 80
    enable
  exit
exit

ip route 0.0.0.0/0 10.0.103.1

ntp enable
ntp server 10.0.103.1
exit

crypto pki server
  challenge-password StR0nnGPass
  subject-name
    common-name ca.company.loc
    organization Company
    locality Moscow
    state Moscow
    country RU
  exit
  lifetime 14
  source-interface gigabitethernet 1/0/1.103
  enable
exit
```

```
hostname ESR.R1

security zone WAN
exit

interface gigabitethernet 1/0/1.113
  security-zone WAN
  ip address 10.0.113.2/30
exit

ip route 0.0.0.0/0 10.0.113.1
```

Решение:

Настроим NTP-клиент на получение точного времени от шлюза Интернет-провайдера:

> Настройка NTP-клиента для работы PKI-клиента не является обязательной, в отличие от конфигурирования PKI-сервера. Но в связи с чувствительностью инфраструктуры открытых ключей к актуальности времени на узлах, использующих выписанные сертификаты, настройка NTP рекомендуется и на клиентской стороне.

```
ESR.R1(config)# ntp enable
ESR.R1(config)# ntp server 10.0.113.1
ESR.R1(config-ntp-server)# exit
ESR.R1(config)#
```

Применим конфигурацию на маршрутизаторе и дождемся успешной синхронизации времени по протоколу NTP:

> Если произвести настройку подключения к NTP серверу одновременно с настройкой PKI-клиента, сгенерированные на этапе запуска PKI-клиента служебные файлы и сертификаты могут содержать некорректные дату и время, что приведет к ошибкам в дальнейшей работе инфраструктуры PKI.

```
ESR.R1(config)# end
Warning: you have uncommitted configuration changes.
ESR.R1# commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
ESR.R1# confirm
Configuration has been confirmed. Commit timer canceled.
ESR.R1#
ESR.R1# sh ntp peers
Clock is synchronized, stratum 6, reference is 10.0.113.1
    remote      vrf   refid         st   t  when  poll  reach  delay  offset    jitter
-   ---------- -----  ------------  --   -  ----  ----  -----  -----  --------  ------
*   10.0.113.1        127.127.1.0   5    u  10    64    37     0.441  0.017     2.344
ESR.R1#
```

Перейдем к настройке PKI-клиента:

```
ESR.R1(config)# crypto pki trustpoint TP_R1
ESR.R1(config-trustpoint)#
```

Перейдем в раздел настройки отличительного имени клиентского сертификата. Настроим те атрибуты, которые позволят однозначно идентифицировать клиента:

```
ESR.R1(config-trustpoint)# subject-name
ESR.R1(config-trustpoint-subject-name)# country RU
ESR.R1(config-trustpoint-subject-name)# state Moscow
ESR.R1(config-trustpoint-subject-name)# locality Moscow
ESR.R1(config-trustpoint-subject-name)# organization Company
ESR.R1(config-trustpoint-subject-name)# common-name r1.company.loc
ESR.R1(config-trustpoint-subject-name)# exit
ESR.R1(config-trustpoint)#
```

Перейдем в раздел настройки альтернативных имен сертификата. Укажем в качестве альтернативных имен IP-адрес и полное доменное имя:

```
ESR.R1(config-trustpoint)# subject-alt-name
ESR.R1(config-trustpoint-san)# ipv4 10.0.113.2
ESR.R1(config-trustpoint-san)# dns r1.company.loc
ESR.R1(config-trustpoint-san)# exit
ESR.R1(config-trustpoint)#
```

Перейдем в режим настройки механизма выпуска сертификата. Для PKI-клиента, выпускающего сертификаты от вышестоящего удостоверяющего центра по протоколу SCEP необходимо указать значение "scep":

```
ESR.R1(config-trustpoint)# enrollment scep
ESR.R1(config-trustpoint-enrollment-scep)#
```

Укажем URL подключения к удостоверяющему центру:

```
ESR.R1(config-trustpoint-enrollment-scep)# url http://10.0.103.2/
ESR.R1(config-trustpoint-enrollment-scep)#
```

Укажем цифровой отпечаток сертификата удостоверяющего центра, при несовпадении настроенного в конфигурации PKI-клиента отпечатка с отпечатком сертификата, которым представится сам удостоверяющий центр процесс выпуска сертификата будет прерван. На сервисных маршрутизаторах ESR получить цифровой отпечаток сертификата удостоверяющего центра можно из вывода команды **show crypto pki server**:

```
ESR.R1(config-trustpoint-enrollment-scep)# fingerprint A2:0C:10:C7:0E:06:3A:2F:EE:5F:C2:72:39:3
6:31:0A
ESR.R1(config-trustpoint-enrollment-scep)#
```

Зададим challenge-password для корректного обращения к удостоверяющему центру:

```
ESR.R1(config-trustpoint-enrollment-scep)# challenge-password StR0nnGP+ss
ESR.R1(config-trustpoint-enrollment-scep)# exit
ESR.R1(config-trustpoint)#
```

Укажем назначение генерируемых сертификатов. Для PKI-клиента, используемого в качестве источника ключевой информации в IPsec-туннелях, необходимо указать значение "ipsec".

```
ESR.R1(config-trustpoint)# usage ipsec
ESR.R1(config-trustpoint)#
```

Привяжем PKI-клиент к адресу сетевого интерфейса, смотрящего в сторону Интернет-провайдера:

```
ESR.R1(config-trustpoint)# source-interface gi 1/0/1.113
ESR.R1(config-trustpoint)#
```

Включим PKI-клиент:

```
ESR.R1(config-trustpoint)# enable
ESR.R1(config-trustpoint)# exit
ESR.R1(config)#
```

Поскольку трафик PKI-клиента исходящий, дополнительных правил фильтрации в Zone-Based Firewall добавлять не требуется.

Применим конфигурацию на маршрутизаторе:

```
ESR.R1(config)# end
Warning: you have uncommitted configuration changes.
ESR.R1# commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 600 seconds.
ESR.R1# confirm
Configuration has been confirmed. Commit timer canceled.
ESR.R1#
```

В результате получим настроенный PKI-клиент, который сразу после запуска обратится к удостоверяющему центру за выпуском сертификата. Отследить состояние настроенных PKI-клиентов можно командой **show crypto pki trustpoints**:

```
ESR.R1# show crypto pki trustpoints
Name     Enrollment  Usage  Subject name                    Status  Next action
-------  ----------  -----  ------------------------------  ------  -------------------
TP_R1    scep        ipsec  CN=r1.company.loc/O=Company/L=  Ready   2026-03-12 11:40:00
                                     Moscow/ST=Moscow/C=RU

ESR.R1#
```

Более подробную информацию о PKI-клиенте можно посмотреть, если выполнить команду **show crypto pki trustpoints** с указанием имени PKI-клиента:

```
ESR.R1# show crypto pki trustpoints TP_R1
Name:               TP_R1
Usage:              ipsec
Subject name:       CN=r1.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:         scep
Challenge password: Active
Status:             Ready
Renew date:         2026-03-12 11:40:00
CRL:                Active
CRL status:         Exist
CRL renew date:     2026-03-01 08:59:16
ESR.R1#
```

17.2.5 Процесс автоматического перевыпуска сертификата PKI-клиента

При успешном получении сертификата от удостоверяющего центра PKI-клиент вычитывает период действия сертификата и за 20% времени до его истечения планирует процедуру перевыпуска нового сертификата (certificate auto-enroll). Дата запланированного перевыпуска сертификата соответствующего PKI-клиента присутствует в графе "Renew date" вывода команды **show crypto pki trustpoints** с указанием имени конкретного PKI-клиента:

```
ESR.R1# show crypto pki trustpoints TP_R1
Name:               TP_R1
Enrollment:         SCEP
Subject name:       CN=r1.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Challenge password: Active
Status:             Ready
Renew date:         2025-11-02 11:35:39
ESR.R1#
```

При наступлении этого времени PKI-клиент запустит процедуру перевыпуска сертификата и в случае успеха заменит текущий сертификат на новый. На это укажет как факт смены запланированной даты перевыпуска сертификата, так и новый серийный номер сертификата:

```
ESR.R1# show crypto pki trustpoints TP_R1 cert | include "Serial"
Serial:                                          85:71:EC:28:6C:C8:4E:23:B2:03:03:2F:4D:DF:B8:6
3
ESR.R1# show crypto pki trustpoints TP_R1
Name:               TP_R1
Usage:              ipsec
Subject name:       CN=r1.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:         scep
Challenge password: Active
Status:             Ready
Renew date:         2026-03-12 11:40:00
CRL:                Active
CRL status:         Exist
CRL renew date:     2026-03-01 08:59:16
ESR.R1#
```

```
ESR.R1# show date
"2025-11-02 11:35:46"
ESR.R1# show crypto pki trustpoints TP_R1
Name:               TP_R1
Usage:              ipsec
Subject name:       CN=r1.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:         scep
Challenge password: Active
Status:             Ready
Renew date:         2026-03-23 16:28:00
CRL:                Active
CRL status:         Exist
CRL renew date:     2026-03-13 11:40:00
ESR.R1# show crypto pki trustpoints TP_R1 cert | include "Serial"
Serial:                                          D4:F1:0C:10:9B:59:4C:88:BF:95:03:D8:F4:E2:40:
8B
ESR.R1#
```

В базе сертификатов удостоверяющего центра, в свою очередь, выписанный ранее клиентский сертификат будет отозван:

```
ESR.CA# show crypto pki server database
Serial                                           State       Issue date   Expiration
Revocation   Subject name
                                                                          date         date
-----------------------------------------------   ---------   ----------   ----------
----------   ------------------------------
85:71:EC:28:6C:C8:4E:23:B2:03:03:2F:4D:DF:B8:63   Revoked     2026-03-01   2026-03-15   2026-03-1
2   CN=r1.company.loc/O=Company/L=
                                                              06:52:00     06:52:00     11:40:0
0     Moscow/ST=Moscow/C=RU

D4:F1:0C:10:9B:59:4C:88:BF:95:03:D8:F4:E2:40:8B   Valid       2026-03-12   2026-03-26   --
CN=r1.company.loc/O=Company/L=
                                                              11:40:00     11:40:00
Moscow/ST=Moscow/C=RU

ESR.CA#
```

17.2.6 Процесс автоматического перевыпуска сертификата PKI-сервера

После завершения инициализации PKI-сервера обслуживающий его PKI-клиент вычитывает период действия сертификата и за 20% времени до его истечения планирует процедуру подготовки так называемого теневого сертификата (certificate auto-rollover). Данный сертификат предназначен для замены текущего сертификата удостоверяющего центра после его истечения. Наличие сгенерированного теневого сертификата отображается в графе "Shadow" вывода команды **show crypto pki server**. Дата запланированной генерации теневого сертификата присутствует в графе "Renew date" вывода команды **show crypto pki trustpoints** с указанием имени PKI-клиента, обслуживающего PKI-сервер:

```
ESR.CA# show crypto pki server
Status:                     Enabled
Associated trustpoint:      ROOT_CA
Lifetime days:              14
Certificate fingerprint:    A2:0C:10:C7:0E:06:3A:2F:EE:5F:C2:72:39:36:31:0A
Source:                     10.0.103.2
Last issued serial number: D4:F1:0C:10:9B:59:4C:88:BF:95:03:D8:F4:E2:40:8B
Challenge password:         Active
Shadow:                     Inactive
ESR.CA# show crypto pki trustpoints
Name               Enrollment   Usage     Subject name                      Status   Next action
----------------   ----------   -------   -------------------------------   -------
-------------------
ROOT_CA            selfsigned   root-ca   CN=ca.company.loc/O=Company/L=   Ready    2026-12-17
08:59:16
                                          Moscow/ST=Moscow/C=RU

ESR.CA# show crypto pki trustpoints ROOT_CA
Name:                ROOT_CA
Usage:               root-ca
Subject name:        CN=ca.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:          selfsigned
Status:              Ready
Renew date:          2026-12-17 08:59:16
CRL:                 Inactive
ESR.CA#
```

После успешной генерации теневого сертификата обновится графа "Shadow" вывода команды **show crypto pki server**, а также состояние соответствующего PKI-клиента в выводах команды **show crypto pki trustpoints**:

```
ESR.CA# show date
"2026-12-17 09:01:36"
ESR.CA#
ESR.CA# show crypto pki server
Status:                     Enabled
Associated trustpoint:      ROOT_CA
Lifetime days:              14
Certificate fingerprint:    A2:0C:10:C7:0E:06:3A:2F:EE:5F:C2:72:39:36:31:0A
Source:                     10.0.103.2
Last issued serial number: 0A:AF:EA:35:92:E6:48:08:B3:A3:7A:8B:29:85:3B:56
Challenge password:         Active
Shadow:                     Active
ESR.CA# show crypto pki trustpoints
Name              Enrollment   Usage     Subject name                      Status
Next action
---------------   ----------   -------   ------------------------------   ------------------
--------------------
ROOT_CA           selfsigned   root-ca   CN=ca.company.loc/O=Company/L=   Ready with shadow
2027-02-28 08:59:16
                                         Moscow/ST=Moscow/C=RU

ESR.CA# show crypto pki trustpoints ROOT_CA
Name:               ROOT_CA
Usage:              root-ca
Subject name:       CN=ca.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:         selfsigned
Status:             Ready with shadow
Renew date:         2027-02-28 08:59:16
CRL:                Inactive
ESR.CA#
```

Любой PKI-клиент, обращающийся за выпуском сертификата в удостоверяющий центр и получивший клиентский сертификат со сроком завершения валидности, эквивалентным сроку завершения жизни сертификата удостоверяющего центра, зафиксирует событие истечения жизни сертификата удостоверяющего центра и отдельным запросом запросит теневой сертификат уже для себя:

```
ESR.R1# sh crypto pki trustpoints
Name              Enrollment   Usage     Subject name                     Status
Next action
---------------   ----------   -------   ------------------------------   ------------------
--------------------
TP_R1             scep         ipsec     CN=r1.company.loc/O=Company/L=   Ready with shadow
2027-02-28 08:59:16
                                         Moscow/ST=Moscow/C=RU

ESR.R1# sh crypto pki trustpoints TP_R1
Name:               TP_R1
Usage:              ipsec
Subject name:       CN=r1.company.loc/O=Company/L=Moscow/ST=Moscow/C=RU
Enrollment:         scep
Challenge password: Active
Status:             Ready with shadow
Renew date:         2027-02-28 08:59:16
CRL:                Active
CRL status:         Exist
CRL renew date:     2027-02-26 15:00:00
ESR.R1# sh crypto pki trustpoints TP_R1 ca | include " after"
    Valid after:                                 2026-02-28 08:59:16
    Invalid after:                               2027-02-28 08:59:16
ESR.R1# sh crypto pki trustpoints TP_R1 cert | include " after"
    Valid after:                                 2027-02-25 15:00:00
    Invalid after:                               2027-02-28 08:59:16
ESR.R1#
```

В момент, когда текущий сертификат удостоверяющего центра истечет, и PKI-клиент самого удостоверяющего центра, и PKI-клиенты, выписывающие клиентские сертификаты, произведут замену текущего сертификата на теневой и продолжат штатную работу:

```
ESR.CA# show crypto pki trustpoints
Name      Enrollment   Usage     Subject name                      Status             Next
action
--------  ----------   -------   ------------------------------   -----------------
--------------------
ROOT_CA   selfsigned   root-ca   CN=ca.company.loc/O=Company/L=   Ready with shadow  2027-02-28
 08:59:16
                                        Moscow/ST=Moscow/C=RU

ESR.CA#
ESR.CA# show date
"2027-02-28 09:00:01"
ESR.CA# sh crypto pki server
Status:                     Enabled
Associated trustpoint:      ROOT_CA
Lifetime days:              14
Certificate fingerprint:    EA:13:09:02:46:6F:F3:91:46:F3:1F:AB:2F:90:CA:18
Source:                     10.0.103.2
Last issued serial number: F3:C4:74:88:E8:AC:44:4D:B9:DC:CD:E8:9E:95:0A:7E
Challenge password:         Active
Shadow:                     Inactive
ESR.CA# show crypto pki trustpoints
Name               Enrollment   Usage     Subject name                      Status   Next action
----------------   ----------   -------   ------------------------------   -------
--------------------
ROOT_CA            selfsigned   root-ca   CN=ca.company.loc/O=Company/L=   Ready    2027-12-17
08:59:16
                                        Moscow/ST=Moscow/C=RU

ESR.CA#
```

```
ESR.R1# show crypto pki trustpoints
Name       Enrollment   Usage     Subject name                      Status                Next
action
---------  ----------   -------   ------------------------------   ------------------
--------------------
TP_R1      scep         ipsec     CN=r1.company.loc/O=Company/L=   Ready with shadow     2027-02-2
8 08:59:16
                                        Moscow/ST=Moscow/C=RU

ESR.R1#
ESR.R1# show date
"2027-02-28 09:00:01"
ESR.R1# show crypto pki trustpoints
Name               Enrollment   Usage     Subject name                      Status     Next
action
----------------   ----------   -------   ------------------------------   ---------
--------------------
TP_R1              scep         ipsec     CN=r1.company.loc/O=Company/L=   Ready      2027-03-11
 13:47:16
                                        Moscow/ST=Moscow/C=RU

ESR.R1#
```

17.3 Ручная генерация и распространение ключей и сертификатов X.509

Процесс выпуска сертификатов и ключей для маршрутизаторов ESR может быть и ручным. Небольшой набор команд позволяет администратору формировать приватный ключ, сгенерировать по нему запрос на сертификацию и отправить его на удаленный хост, откуда его уже можно будет доставить в УЦ для выпуска сертификата. Пользовательский сертификат, а также сопутствующие файлы могут быть загружены обратно на ESR для дальнейшего использования в сервисах маршрутизатора.

17.3.1 Алгоритм генерации ключей и запросов на сертификацию

Шаг	Описание	Команда	Ключи
1	Сгенерировать приватный ключ RSA.	**esr# crypto generate private-key rsa <KEY-SIZE> filename <NAME>**	<KEY-SIZE> – размер ключа в битах. Значение может находится в диапазоне от 1024 до 4096; <NAME> – имя файла приватного ключа, задаётся строкой до 31 символа.

Шаг	Описание	Команда	Ключи
2	Сформировать запрос на сертификацию.	**esr# crypto generate csr private-key <PRIVATE-KEY> [[alternative-name <ALT-NAME>][common-name <COMMON-NAME>][country <COUNTRY>][email-address <E-MAIL>][locality <LOCATION>] [organization <ORGANIZATION>] [organizational-unit <ORGANIZATION-UNIT>][state <STATE>]] filename <NAME>**	<PRIVATE-KEY> – имя файла приватного ключа, задаётся строкой до 31 символа; <ALT-NAME> – альтернативное имя сертификата, задаётся строкой от 5 до 255 символов в формате <TYPE>:<VALUE>, где: • <TYPE> – спецификатор типа альтернативного имени, может принимать значения "IP" и "DNS".

Шаг	Описание	Команда	Ключи
			• <VALUE> – значение альтернативного имени сертификата, может принимать на сход корректно написанный IPv4-адрес или доменное имя. <COMMON-NAME> – общее имя, задаётся строкой от 1 до 64 символов. Чаще всего в качестве общего имени клиентского сертификата используется полное доменное имя хоста, который использует данный клиентский сертификат или ФИО пользователя, использующего клиентский сертификат; <COUNTRY> – код страны, задаётся строкой длиной 2 символа. Рекомендуется использовать двухбуквенные обозначения стран "alpha-2" из стандарта ISO 3166-1; <E-MAIL> – адрес электронной почты, задаётся строкой от 3 до 64 символов; <LOCATION> – название населенного пункта или его территориальной единицы, задаётся строкой от 1 до 128 символов; <ORGANIZATION> – название организации, задаётся строкой от 1 до 64 символов; <ORGANIZATION-UNIT> – название подразделения организации, задаётся строкой от 1 до 64 символов; <STATE> – название штата, области или провинции, задаётся строкой от 1 до 128 символов; <NAME> – имя файла запроса на сертификацию, задаётся строкой до 31 символа.

Шаг	Описание	Команда	Ключи
3	Выгрузить запрос на сертификацию с устройства.	**esr# copy crypto:csr/<CERT-REQ> <DESTINATION>**	<CERT-REQ> – имя файла запроса на сертификацию, задаётся строкой до 31 символа; <DESTINATION> – локальное или удаленное файловое хранилище, куда будет выгружен запрос на сертификацию. Полный список возможных вариантов и синтаксис их описания подробно указан в описании команды **copy** в справочнике команд CLI.
4	Загрузить на устройство клиентский сертификат, выписанный удостоверяющим центром.	**esr# copy <SOURCE> crypto:cert/ <CERT>**	<SOURCE> – локальное или удаленное файловое хранилище, откуда будет загружен файл клиентского сертификата. Полный список возможных вариантов и синтаксис их описания подробно указан в описании команды **copy** в справочнике команд CLI; <CERT> – имя файла клиентского сертификата, задаётся строкой до 31 символа.
5	Загрузить на устройство сертификат удостоверяющего центра.	**esr# copy <SOURCE> crypto:cert/ <CA-CERT>**	<SOURCE> – локальное или удаленное файловое хранилище, откуда будет загружен файл сертификата удостоверяющего центра. Полный список возможных вариантов и синтаксис их описания подробно указан в описании команды **copy** в справочнике команд CLI; <CA-CERT> – имя файла сертификата удостоверяющего центра, задаётся строкой до 31 символа.

Шаг	Описание	Команда	Ключи
6	Загрузить на устройство список отозванных сертификатов удостоверяющего центра.	**esr# copy <SOURCE> crypto:crl/ <CRL>**	<SOURCE> – локальное или удаленное файловое хранилище, откуда будет загружен файл со списком отозванных сертификатов. Полный список возможных вариантов и синтаксис их описания подробно указан в описании команды **copy** в справочнике команд CLI; <CRL> – имя файла со списком отозванных сертификатов, задаётся строкой до 31 символа.

17.3.2 Пример ручного выпуска сертификата через внешний удостоверяющий центр

Задача:

Сгенерировать на ESR приватный RSA-ключ и сформировать запрос на сертификацию. Помимо настройки отличительного имени сертификата настроим в качестве альтернативного имени его полное доменное имя.

Решение:

> Для удобства дальнейшей идентификации ключей и сертификатов рекомендуется использовать в качестве имен файлов доменное имя хоста, для которого формируется ключевая пара и префикс сервиса.

> Поскольку сертификаты (как клиентские, так и самого удостоверяющего центра) имеют ограниченный срок действия, то отслеживание истечения срока действия сертификатов и их своевременный перевыпуск полностью находится в зоне ответственности администратора маршрутизатора.

Сгенерируем приватный ключ RSA:

> При генерации ключей RSA рекомендуется использовать ключи длиной 2048 бит и более.

```
ESR.R1# crypto generate private-key rsa 2048 filename r1.company.loc.key
.....+.......+..+..........+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+*......+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*...+.........+..
+.+...+.........+..+...+......+....+...+........+
.......+........+.+......+.........+.......................+..+++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++
............++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.......+...++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.+...+......+.....+......+.......
+......+......+.....+...+...............+.+.....
.+.....+.+.....+...+.......+.....+...+....+...+..+...........+......+....+........+.+..
+............+.+...........+..................+...+....+...........+.........+....+..+.......
+......+......+........+...........................+.....
........+...+.........+.....+.+...+...........+..+..........+..+...+.+..+...+................
+.....+.+........+.+.....+....+...........+...+...+...+.......+........+..........+..+.........
+...+...+...+.+...+......+......+......+...+.....
.........+.+.....+.+...+..+...+.......+......+...........+......+.........+.....+.
+............+.....+....+...+...+..................+...+.......+..+.+...........+..+.+...+.....
+....+...............+...........+....+...+................
+.+...........+...+.+.....+.+...+...........+..........+.....+.........+...+....+.............
+.+........+...+....+......+........+.+......+..+.+....................+....+......+..+...
+................+...+........+.......+...........+.+
......+..+.+......+.....+...+......+.......+........+.+......+........+......+.+...
+...........................+........+.+...+...........+.+.....+.+.........+...+.....+.......
+.....+......+...+......+....................+..........+..+.
.....+......+.+..............+.+.....+....+...+...........+......+....+.....+....+...+...+..
+............+.+.................+...+.+..............+............+.........+......+....
+......+...+.........+..+.+..+.+..+.....................
+........+.......+..+....+......+......+..+...+..........+.........................
+............+.....+...+.+.....+....+...+.....+....+...+...........+......+.+...+.....+++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++
ESR.R1#
```

Сформируем запрос на сертификацию:

```
ESR.R1# crypto generate csr private-key r1.company.loc.key country RU state Moscow locality
Moscow organization Company common-name r1.company.loc email-address netmaster@company.loc
alternative-name DNS:r1.company.loc filename r1.company.loc.csr
ESR.R1#
```

Выгрузим сформированный запрос на сертификацию на внешний TFTP-сервер для дальнейшей доставки в удостоверяющий центр:

```
ESR.R1# copy crypto:csr/r1.company.loc.csr tftp://10.0.113.1:/r1.company.loc.csr
|****************************************| 100% (1094B) Success!
ESR.R1#
```

> Процесс доставки запроса на сертификацию в удостоверяющий центр и непосредственно сам выпуск сертификата индивидуален для каждого удостоверяющего центра и не рассматривается в данном разделе руководства.

После успешного выпуска сертификата удостоверяющим центром загрузим клиентский сертификат, сертификат удостоверяющего центра и список отозванных сертификатов на маршрутизатор:

```
ESR.R1# copy tftp://10.0.113.1:/ca.crt crypto:cert/ca.crt
|*****************************************| 100% (2264B) Crypto file loaded successfully!
ESR.R1# copy tftp://10.0.113.1:/ca.crl crypto:crl/ca.crl
|*****************************************| 100% (1064B) Crypto file loaded successfully!
ESR.R1# copy tftp://10.0.113.1:/r1.company.loc.crt crypto:cert/r1.company.loc.crt
|*****************************************| 100% (1931B) Crypto file loaded successfully!
ESR.R1#
```

Посмотреть итоговое содержимое сертификата можно командой **show crypto certificates**:

```
ESR.R1# show crypto certificates cert r1.company.loc.crt
Version:                                          3
Serial:                                           4096
Subject name:
    C(countryName):                               RU
    ST(stateOrProvinceName):                      Moscow
    O(organizationName):                          Company
    CN(commonName):                               r1.company.loc
    emailAddress(emailAddress):                   netmaster@company.loc
Issuer name:
    C(countryName):                               RU
    ST(stateOrProvinceName):                      Moscow
    L(localityName):                              Moscow
    O(organizationName):                          Company
    CN(commonName):                               Company Root Certificate Authority
Validity period:
    Valid after:                                  2025-10-28 03:26:41
    Invalid after:                                2025-11-27 03:26:41
Signature:
    Algorithm:                                    sha256WithRSAEncryption
    Value:                                        37:6D:30:DE:C3:EF:D8:06:D6:4B:AA:AC:6A:78:65:
C2:7C:7B:
                                                  EA:E9:F6:C0:A7:0F:9B:01:D2:C6:05:95:43:A1:C6:
7B:F7:43:
                                                  F7:BE:78:7F:BA:65:73:88:31:91:C5:4F:FA:BF:41:9
9:D6:28:
                                                  A5:29:72:85:20:52:2E:0C:1D:3E:37:78:10:B5:CC:
AE:D5:A9:
                                                  A6:79:FE:07:F4:93:E2:E9:F4:48:17:E5:A6:EB:36:
D9:3E:41:
                                                  2E:8D:E9:7A:D4:75:49:A5:98:8F:76:73:8A:A7:E6:
1D:89:CA:
                                                  46:B2:FC:A6:E7:96:F3:79:EB:5A:B1:B0:63:E4:AC:
7C:D9:29:
                                                  AE:2D:04:4E:45:B0:08:38:7F:C6:62:72:04:C6:A2:
7C:BC:77:
                                                  AF:CE:92:2F:66:75:33:8B:81:AB:98:40:61:74:9C:
6B:10:15:
                                                  78:A7:58:02:DA:D1:69:C2:C2:8B:DA:66:18:BD:13:
FB:4F:7D:
                                                  35:35:C3:21:6F:0A:A0:53:ED:56:F8:B9:E9:0E:6F:
6D:DD:E1:
                                                  A0:AD:4A:07:97:AE:79:3C:2F:7C:E6:76:DD:9F:37:5
0:EB:AD:
                                                  56:3D:BA:51:D0:C1:15:25:54:F6:E2:1D:12:39:46:
5F:E3:33:
                                                  1E:49:26:04:E0:23:FB:C5:FA:A8:0D:B7:16:23:C5:
C9:3E:0C:
                                                  85:E2:CE:72:B5:97:0D:3D:15:D6:5D:F7:12:78:9C:8
4:D2:21:
                                                  C8:EC:BE:45:90:A5:CB:38:87:AB:8C:04:4B:BA:42:
2B:40:95:
                                                  94:BE:F9:82:80:44:76:79:ED:42:5A:ED:28:07:E4:1
6:6D:C8:
                                                  80:D4:33:87:97:39:20:E9:CC:EB:F3:74:CE:F2:3C:
6E:4E:C8:
                                                  37:51:21:F8:CA:AD:C6:09:3B:19:07:B2:34:3C:17:3
1:B8:22:
```

```
                                                        CC:BA:73:5E:9F:CD:F4:8B:38:71:BB:2F:7A:A5:F5:4
3:A8:8E:
                                                        07:47:36:BA:8D:BA:DB:BB:8F:9C:EB:49:A4:6C:2E:3
0:30:C1:
                                                        AF:06:F1:0D:E6:C7:DA:7B:FD:94:68:FD:F0:B3:3F:3
0:45:8C:
                                                        ED:77:FE:09:64:0E:4D:02:03:82:3A:30:61:24:08:
4A:AF:BD:
                                                        C2:32:6B:70:78:E6:C1:F2:6E:2A:3E:30:1A:7A:A2:
BE:70:7F:
                                                        86:8A:9B:12:D0:92:7D:14:99:72:FA:30:29:BE:44:
8F:3C:D8:
                                                        75:16:AE:BD:23:97:E0:04:B5:8A:B9:71:F0:F7:15:
0A:A8:95:
                                                        CC:51:23:21:6E:3F:9B:64:B1:73:A7:2F:03:22:46:
6F:DD:A2:
                                                        90:A1:E4:7F:94:92:7F:E7:C2:C5:B9:F9:9D:D3:19:
CF:34:3D:
                                                        D0:C0:E0:30:F8:77:1A:E8
Public key info:
    Algorithm:                                          RSA
    Key size:                                           2048
    Exponent:                                           65537
    Modulus:                                            00:9D:41:BB:13:A8:99:9C:3E:E7:2C:0E:A5:B6:A8:
CA:22:64:
                                                        BB:B9:77:E5:CE:DE:5E:71:83:9A:90:22:D1:32:E1:6
6:45:FC:
                                                        6C:53:DA:65:D5:FF:C7:35:2C:24:F6:BA:AD:72:DD:2
7:A5:09:
                                                        30:CC:AA:E3:F8:33:B5:10:1C:23:D9:EA:DA:30:6F:
E4:2C:C4:
                                                        EC:08:E9:12:72:05:0C:C1:CF:6B:72:8F:B5:E8:5B:9
0:67:B1:
                                                        4C:59:D3:4D:CA:0C:73:94:47:F7:DB:BC:83:38:24:
E2:AC:19:
                                                        DF:7D:8F:99:E0:B2:72:E3:A3:5B:7E:B8:EC:7B:6C:1
7:8C:48:
                                                        5A:F2:F5:A5:14:D3:07:E3:7E:5A:CD:70:6A:9E:38:
2D:80:4F:
                                                        29:B3:60:F8:AC:7B:C5:09:09:B9:4B:92:D4:E0:44:
5D:9E:1A:
                                                        AF:0E:25:FA:E5:73:C3:51:8F:DE:BB:F5:71:0C:2F:
F3:AC:F5:
                                                        7D:79:8A:E6:87:0A:05:6A:D8:C8:6D:FE:BE:90:7D:
B5:A2:3D:
                                                        3B:75:96:CF:25:98:5C:0B:F2:E3:C1:E7:B5:30:58:2
7:13:64:
                                                        DD:DB:77:A8:10:9C:A5:25:AC:85:DA:30:21:87:71:
A8:D7:D8:
                                                        BC:60:40:C7:53:54:01:03:0E:60:5D:2B:43:99:97:
F2:26:6E:
                                                        8F:F7:47:CF:9F
X509v3 Basic Constraints:
    CA:                                                 No
    Critical:                                           Yes
X509v3 Subject key identifier:
    ID:                                                 40:6D:58:0E:0A:4C:CD:89:71:CA:DB:D2:BC:AD:FA:2
7:C9:1E:
                                                        4D:D8
```

```
    Critical:                                   No
X509v3 Authority key identifier:
    ID:                                         7C:E6:3C:E3:FB:76:C5:18:B3:21:52:9D:8F:71:29:2
8:55:CA:
                                                96:63
    Critical:                                   No
X509v3 Key Usage:
    Usage:                                      Digital Signature
                                                Non Repudiation
    Critical:                                   Yes
X509v3 Subject Alternative Name:
    Names:                                      DNS:r1.company.loc
    Critical:                                   No
ESR.R1#
```

Как можно заметить, выписанный сертификат действует 30 дней и в нем было сохранено заданное альтернативное имя. Теперь данный сертификат можно использовать в сервисах маршрутизатора.

> Для корректной работы сервисов, использующих сертификаты, рекомендуется настроить на маршрутизаторе синхронизацию времени по протоколу NTP.

18 Управление резервированием

Алгоритм настройки резервирования см. в документации ESR.

> Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

19 Управление кластеризацией

- Настройка кластера на ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350
  - Первичная настройка кластера
  - Настройка внешних сетевых интерфейсов
  - Настройка кластерного интерфейса
  - Настройка кластера
  - Настройка синхронизации сертификатов и ключей
- Настройка кластера на vESBC
  - Пример настройки кластера vESBC в гипервизоре VirtualBox
  - Пример настройки кластера vESBC в гипервизоре QEMU/KVM
  - Пример настройки кластера vESBC в гипервизоре XEN
  - Пример настройки кластера vESBC в гипервизоре XCP-ng
  - Особенности настройки гипервизора ESXi для организации кластера vESBC
  - Пример настройки кластера vESBC в гипервизоре zVirt
  - Подключение второго юнита в кластере vESBC с использованием ZTP

Кластер организуется из двух одинаковых устройств. Работает в режиме Active-Standby, т. е. на Active запущены все модули ESBC, и он занимается обработкой сигнальных сообщений SIP и медиаданных (RTP-потоков). Устройство Standby не обрабатывает сигнальные сообщений SIP и потоки RTP.

Резервирование соединения осуществляется протоколом VRRP.

Конфигурация, файлы ПО, зарегистрированные абоненты синхронизируются между устройствами в реальном времени. В случае обрыва соединения или отключения Active-устройства, все существующие вызовы будут разрушены.

> ⚠ Время до начала обработки вызовов при failover зависит от вызывной нагрузки на устройство и составляет от 2 до 12 секунд.

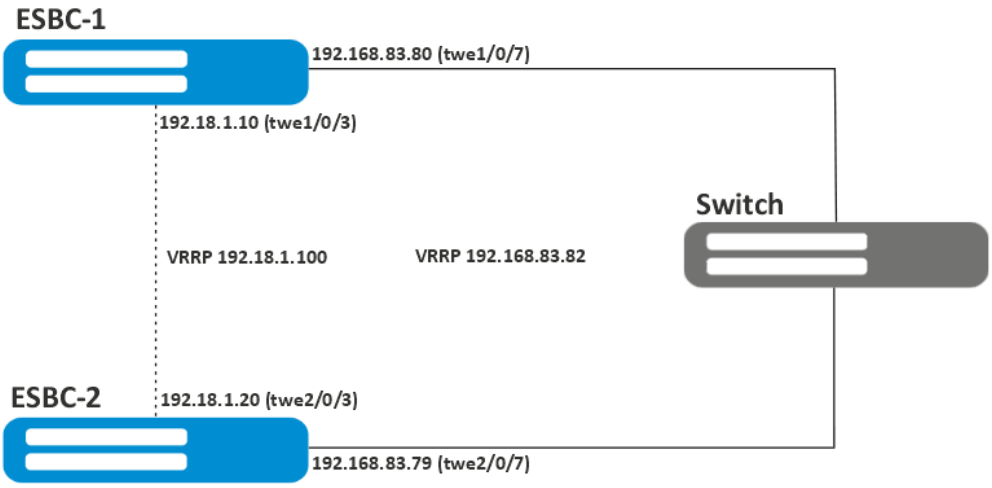
Более подробное описание настройки кластера приведено в документации ESR. Ниже представлен пример настройки ESBC-3200 для обработки вызовов.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

> ⚠ В текущей версии ПО поддерживается HA Cluster из 2 юнитов.

19.1 Настройка кластера на ESBC-15R, ESBC-31, ESBC-3200, ESBC-3250, ESBC-3350

Схема:



19.1.1 Первичная настройка кластера

Для начала работы необходимо полностью настроить одно устройство из кластера.

После включения устройства примените конфигурацию по умолчанию на устройствах, предназначенных для объединения в кластер:

ESBC-1

```
ESBC-3200# copy system:default-config system:candidate-config
Entire candidate configuration will be reset to default, all settings will be lost upon commit.
Do you really want to continue? (y/N): y
|****************************************| 100% (59B) Default configuration loaded
successfully.
```

Для более удобного и ясного восприятия рекомендуется переименовать устройства. В кластерной версии прошивки предусмотрена возможность указать имя устройства с привязкой к юниту. Устройство будет использовать только тот hostname, юнитом которого он является:

ESBC-1

```
ESBC-3200# configure
ESBC-3200(config)# hostname ESBC-1 unit 1
ESBC-3200(config)# hostname ESBC-2 unit 2
```

⚠ В конфигурации может одновременно находиться hostname с unit и hostname без unit. Более приоритетным является **hostname**, указанный с привязкой к **unit**.

Чтобы изменить юнит устройства, выполните следующие команды:

ESBC-1

```
ESBC-1# set unit id 1
Unit ID will be 1 after reboot
ESBC-1# reload system
Do you really want to reload system now? (y/N): y
```

Смена юнита устройства вступает в силу после перезагрузки.

При изменении номера юнита ESBC не происходит автоматической конвертации конфигурации. В случае если до ESBC настроен удаленный доступ и у него меняется номер юнита, необходимо до перезагрузки настроить ip-интерфейсы для нового юнита аналогично текущим.

В заводской конфигурации **ESBC-15R/31/3200/3250/3350** присутствуют настройки интерфейсов только для юнита по умолчанию (unit = 1).
При копировании и применении заводской конфигурации настройка номера юнита не изменяется на значение по умолчанию.
Установить номер юнита по умолчанию возможно следующими способами:
1. используя консольное подключение;
2. зажав функциональную кнопку "F" на 15 секунд.

Убедитесь в том, что настройки юнитов применились успешно:

ESBC-1

```
ESBC-1# show unit id
Unit ID is 1
Unit ID will be 1 after reboot
```

Объединение устройств в кластер невозможно, если они относятся к одному и тому же юниту. Исключение — процесс ZTP, так как в процессе ZTP нужный unit у устройства выставится автоматически.

19.1.2 Настройка внешних сетевых интерфейсов

На обоих устройствах необходимо настроить IP-адрес и VRRP на внешних интерфейсах. В текущей схеме это интерфейсы twe1/0/7 и twe2/0/7.

ESBC-1

```
ESBC-1(config)# interface twentyfivegigabitethernet 1/0/7
ESBC-1(config-if-twe)# ip address 192.168.83.80/22
ESBC-1(config-if-twe)# vrrp 10
ESBC-1(config-vrrp)# ip address 192.168.83.82/22
ESBC-1(config-vrrp)# group 2
ESBC-1(config-vrrp)# enable
ESBC-1(config-vrrp)# exit
ESBC-1(config-if-twe)# exit
ESBC-1(config)# interface twentyfivegigabitethernet 2/0/7
ESBC-1(config-if-twe)# ip address 192.168.83.79/22
ESBC-1(config-if-twe)# vrrp 10
ESBC-1(config-vrrp)# ip address 192.168.83.82/22
ESBC-1(config-vrrp)# group 2
ESBC-1(config-vrrp)# enable
ESBC-1(config-vrrp)# exit
ESBC-1(config-if-twe)# exit
```

Также на VRRP-интерфейсе можно назначить разные приоритеты для разных юнитов.

ESR-1

```
ESBC-1(config-vrrp)# priority 254 unit 1
ESBC-1(config-vrrp)# priority 253 unit 2
```

Допускается использование IP-адреса VRRP из подсети, отличной от подсети физического интерфейса.

19.1.3 Настройка кластерного интерфейса

Для полноценной работы кластера требуется сконфигурировать кластерный интерфейс, который будет использоваться для передачи control plane трафика. В качестве кластерного интерфейса назначен bridge. В качестве механизма, отвечающего за определение ролей устройств, участвующих в резервировании, назначен протокол VRRP. Настройки cluster-интерфейса должны быть идентичны для всех участников кластера.

Так как кластер выполняет синхронизацию состояний между устройствами, необходимо создать зону безопасности SYNC (synchronization):

ESBC-1

```
ESBC-1(config)# security zone SYNC
ESBC-1(config-zone)# exit
ESBC-1(config)#
ESBC-1(config)# security zone-pair SYNC self
ESBC-1(config-zone-pair)# rule 1
ESBC-1(config-zone-pair-rule)# action permit
ESBC-1(config-zone-pair-rule)# match protocol vrrp
ESBC-1(config-zone-pair-rule)# enable
ESBC-1(config-zone-pair-rule)# exit
ESBC-1(config-zone-pair)# exit
```

Далее перейдите к настройкам кластерного интерфейса:

ESBC-1

```
ESBC-1# configure
ESBC-1(config)# bridge 1
ESBC-1(config-bridge)# vlan 1
ESBC-1(config-bridge)# security-zone SYNC
ESBC-1(config-bridge)# ip address 192.18.1.10/24 unit 1
ESBC-1(config-bridge)# ip address 192.18.1.20/24 unit 2
ESBC-1(config-bridge)# vrrp 1
ESBC-1(config-vrrp)# group 2
ESBC-1(config-vrrp)# ip address 192.18.1.100/24
ESBC-1(config-vrrp)# enable
ESBC-1(config-vrrp)# exit
ESBC-1(config-bridge)# enable
```

⚠ В текущей версии ПО в качестве cluster-интерфейса поддержан только bridge.

⚠ Для работы кластерного интерфейса поддерживается только IPv4-адресация.
На cluster-интерфейсе необходима настройка адресов с привязкой к unit.

В текущей схеме служебная информация по управлению кластером будет передаваться через выделенный линк синхронизации между интерфейсами twe1/0/3 и twe2/0/3.

ESBC-1

```
ESBC-1(config)# interface twentyfivegigabitethernet 1/0/3
ESBC-1(config-if-twe)# description "Network: SYNC"
ESBC-1(config-if-twe)# mode switchport
ESBC-1(config-if-twe)# exit
ESBC-1(config)# interface twentyfivegigabitethernet 2/0/3
ESBC-1(config-if-twe)# description "Network: SYNC"
ESBC-1(config-if-twe)# mode switchport
ESBC-1(config-if-twe)# exit
```

19.1.4 Настройка кластера

Для запуска кластера нужно только указать заранее настроенный кластерный интерфейс и юниты, которые будут выполнять роли Active и Standby.

Перейдите в настройку кластера:

ESBC-1

```
ESBC-1# configure
ESBC-1(config)# cluster
ESBC-1(config-cluster)# unit 1
ESBC-1(config-cluster-unit)# mac-address 68:13:e2:e1:28:90
ESBC-1(config-cluster-unit)# exit
ESBC-1(config-cluster)# unit 2
ESBC-1(config-cluster-unit)# mac-address 68:13:e2:e1:25:30
ESBC-1(config-cluster-unit)# exit
```

⚠ В качестве mac-address указывается системный MAC-адрес устройства, его можно узнать с помощью команды show system | include MAC.

Укажите кластерный интерфейс, созданный ранее, и активируйте кластер:

ESBC-1

```
ESBC-1(config-cluster)# cluster-interface bridge 1
ESBC-1(config-cluster)# enable
ESBC-1(config-cluster)# do commit
ESBC-1(config-cluster)# do confirm
```

Первое устройство полностью настроено и готово к работе.
Аналогичные настройки необходимо произвести на втором устройстве, предварительно сменив у него юнит на требуемый.

Также возможна настройка второго устройства средствами ZTP.

> Для активации процесса ZTP необходимо на втором устройстве запустить dhcp-client на bridge-интерфейсе, логический или физический интерфейс которого будет включен в кластерный интерфейс первого устройства.
> В качестве примера такой конфигурации подойдет factory-конфигурация.
> В процессе ZTP устройство автоматически выставит себе:
> 1) Конфигурацию;
> 2) Юнит;
> 3) Версию ПО, на котором работает Active ESBC;
> 4) Лицензию, если она предварительно загружена на Active ESBC.

После выполнения этих шагов кластер будет успешно запущен. Текущее состояние кластера можно узнать, выполнив команду:

ESBC-1

```
ESBC-1# show cluster status
Unit   Hostname                Role         MAC address         State            IP address
----   --------------------    ----------   -----------------   --------------   ---------------
1*     ESBC-1                  Active       68:13:e2:e1:28:90   Joined           192.18.1.10
2      ESBC-2                  Standby      68:13:e2:e1:25:30   Joined           192.18.1.20
```

> После включения кластера и установления юнитов в состояние Joined дальнейшее конфигурирование устройств осуществляется настройкой Active-устройства.
> Синхронизируются команды конфигурации, а также команды: *commit, confirm, rollback, restore, save, copy <source> system:candidate-config*.
> В случае если конфигурирование осуществляется на Standby, то внесенные изменения в конфигурацию засинхронизированы не будут. Все внесённые изменения в конфигурацию Standby будут потеряны при выполнении *commit* на Active-устройстве.
> Есть возможность отключения синхронизации командой *sync config disable*.

Для проверки работы протокола VRRP выполните следующую команду:

ESBC-1

```
ESBC-1# show vrrp
Virtual router   Virtual IP         Priority   Preemption   State    Synchronization group ID
--------------   -----------------  --------   ----------   ------   -------------------------
1                192.18.1.100/24      100        Enabled     Master   2
10               192.168.83.82/22     100        Enabled     Master   2
```

Также можно посмотреть состояние синхронизации различных подсистем в кластере, выполнив команду:

ESBC-1

```
ESBC-1# show cluster sync status
System part              Synced
----------------------   ------
candidate-config         Yes
running-config           Yes
SW version               Yes
licence                  Yes
licence (After reboot)   Yes
date                     Yes
E-SBC version            Yes
```

Через минуту после включения кластера синхронизируется время, на Standby установится время Active-юнита.
Синхронизация времени проверяется раз в минуту, в случае расхождения время синхронизируется.

Работа с лицензиями в кластере описана в разделе Лицензирование в кластере.

19.1.5 Настройка синхронизации сертификатов и ключей

При использовании в конфигурации ESBC файлов сертификатов и ключей, например, в конфигурации криптопрофилей, загружаемых на устройство пользователем, необходимо наличие данных файлов на обоих юнитах в кластере.

Загрузить файлы можно отдельно на каждый юнит кластера или использовать механизм синхронизации crypto-sync.

Пример настройки crypto-sync

Задача:

Использовать сертификаты CA.pem, CERT.pem и приватный ключ PRIVATE_KEY.key в конфигурации криптопрофиля ESBC.

Решение:

1. После описанной выше настройки кластера, настройте ip failover:
   - Создайте две object-group, укажите в них IP-адреса кластерных интерфейсов обоих юнитов:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# object-group network SYNC_DST
ESBC-1(config-object-group-network)# ip address-range 192.18.1.20 unit 1
ESBC-1(config-object-group-network)# ip address-range 192.18.1.10 unit 2
ESBC-1(config-object-group-network)# exit
ESBC-1(config)# object-group network SYNC_SRC
ESBC-1(config-object-group-network)# ip address-range 192.18.1.10 unit 1
ESBC-1(config-object-group-network)# ip address-range 192.18.1.20 unit 2
ESBC-1(config-object-group-network)# exit
ESBC-1(config)#
```

   - Настройте ip failover:

```
ESBC-1(config)# ip failover
ESBC-1(config-failover)# local-address object-group SYNC_SRC
ESBC-1(config-failover)# remote-address object-group SYNC_DST
ESBC-1(config-failover)# vrrp-group 2
ESBC-1(config-failover)# exit
ESBC-1(config)#
```

2. Включите синхронизацию файлов crypto и примените настройки:

```
ESBC-1(config)# crypto-sync
ESBC-1(config-crypto-sync)# remote-delete
ESBC-1(config-crypto-sync)# enable
ESBC-1(config-crypto-sync)# exit
ESBC-1(config)# do commit
ESBC-1(config)# do confirm
```

3. Загрузите сертификаты и приватный ключ на мастер, используя, например, tftp:

```
ESBC-1#
ESBC-1# copy tftp://192.168.1.1:/CA.pem crypto:cert/CA.pem
|*****************************************| 100% (1277B) Crypto file loaded successfully!
ESBC-1#
ESBC-1# copy tftp://192.168.1.1:/CERT.pem crypto:cert/CERT.pem
|*****************************************| 100% (1155B) Crypto file loaded successfully!
ESBC-1#
ESBC-1# copy tftp://192.168.1.1:/PRIVATE_KEY.key crypto:private-key/PRIVATE_KEY.key
|*****************************************| 100% (1675B) Crypto file loaded successfully!
```

4. Проверьте, что файлы сертификатов и приватного ключа были скопированы на второй юнит:

```
#Проверка состояния синхронизации crypto на мастер-юните:
ESBC-1# sh crypto-sync
    Role:                        Master
    State:                       Synchronized
    Last synchronization:        2026-03-04 08:56:10


#Проверка наличия файлов на втором юните:
ESBC-2# dir crypto:cert/
Name                                 Type         Size            Last modified
-----------------------------------  ----------   --------   --   -------------------------
CA.pem                               File         1.13       KB   Wed Mar  4 08:37:43 2026

default_ca.pem                       File         1.38       KB   Wed Feb 25 11:06:17 2026

default_cert.pem                     File         1.22       KB   Wed Feb 25 11:06:17 2026

CERT.pem                             File         1.25       KB   Wed Mar  4 08:37:25 2026

ESBC-2# dir crypto:private-key/
Name                                 Type         Size            Last modified
-----------------------------------  ----------   --------   --   -------------------------
default_ca_key.pem                   File         1.66       KB   Wed Feb 25 11:06:17 2026

default_cert_key.pem                 File         1.66       KB   Wed Feb 25 11:06:17 2026

PRIVATE_KEY.key                      File         1.64       KB   Wed Mar  4 08:38:04 2026
```

5. Настройте криптопрофиль с загруженными сертификатами и приватным ключем на мастер-устройстве:

```
ESBC-1#
ESBC-1# configure
ESBC-1(config)# esbc
ESBC-1(config-esbc)# crypto profile TEST
ESBC-1(config-esbc-crypto-profile)# ca CA.pem
ESBC-1(config-esbc-crypto-profile)# cert CERT.pem
ESBC-1(config-esbc-crypto-profile)# private-key PRIVATE_KEY.key
ESBC-1(config-esbc-crypto-profile)# exit
ESBC-1(config-esbc)# exit
ESBC-1(config)# do commit
Configuration has been successfully applied and saved to flash. Commit timer started, changes
will be reverted in 120 seconds.
ESBC-1(config)# do confirm
Configuration has been confirmed. Commit timer canceled.
vesbc1(config)#
```

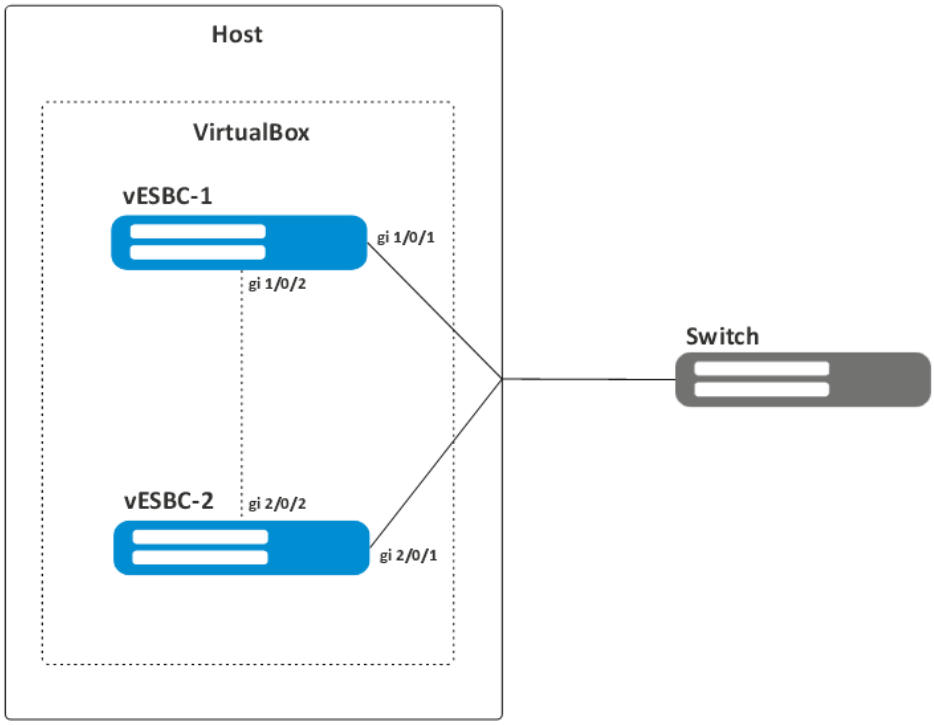
19.2 Настройка кластера на vESBC

19.2.1 Пример настройки кластера vESBC в гипервизоре VirtualBox

Задача:

Развернуть кластер из двух vESBC в среде виртуализации VirtualBox 7.1.0.

Схема:



Решение:

1. Создайте виртуальную машину в VirtualBox, подробно описано в разделе Создание виртуальных машин Руководства по установке vESBC в среде виртуализации VirtualBox.
2. Установите vESBC, подробно описано в разделе Установка vESBC в системе виртуализации VirtualBox.
3. Настройте сетевые интерфейсы:
   - Настройка внешнего интерфейса(ов):
     - Перейдите в настройки виртуальной машины, на которой установлен vESBC.
     - Откройте меню "Сеть" и перейдите к настройкам "Адаптер 1".
     - Включите сетевой интерфейс.
     - Выберите тип подключения "Сетевой мост".
     - Выберите, через какой физический интерфейс хостовой системы будет происходить подключение.
     - Выберите тип адаптера "Intel PRO/1000MT Server (82545EM)".
     - Выберите "Неразборчивый режим: Разрешить всё".
   - Настройка кластерного интерфейса:
     - Перейдите к настройкам второго интерфейса "Адаптер 2".
     - Включите сетевой интерфейс.

    - Выберите тип подключения "Внутренняя сеть".
    - Укажите имя сети (по умолчанию intnet). При настройке второй виртуальной машины.
    - Выберите тип адаптера "Паравитуальная сеть (virtio-net)".
    - Выберите "Неразборчивый режим: Разрешить всё".
  - Нажмите кнопку "OK"
4. Выполните пункты 1-3 для второй виртуальной машины vESBC.

После успешного выполнения описанных пунктов можно приступать к настройке юнитов кластера vESBC. Алгоритм настройки полностью аналогичен настройке аппаратных пограничных контроллеров сессий и описан в разделе Первичная настройка кластера, за исключением конфигурации второго юнита с помощью ZTP.

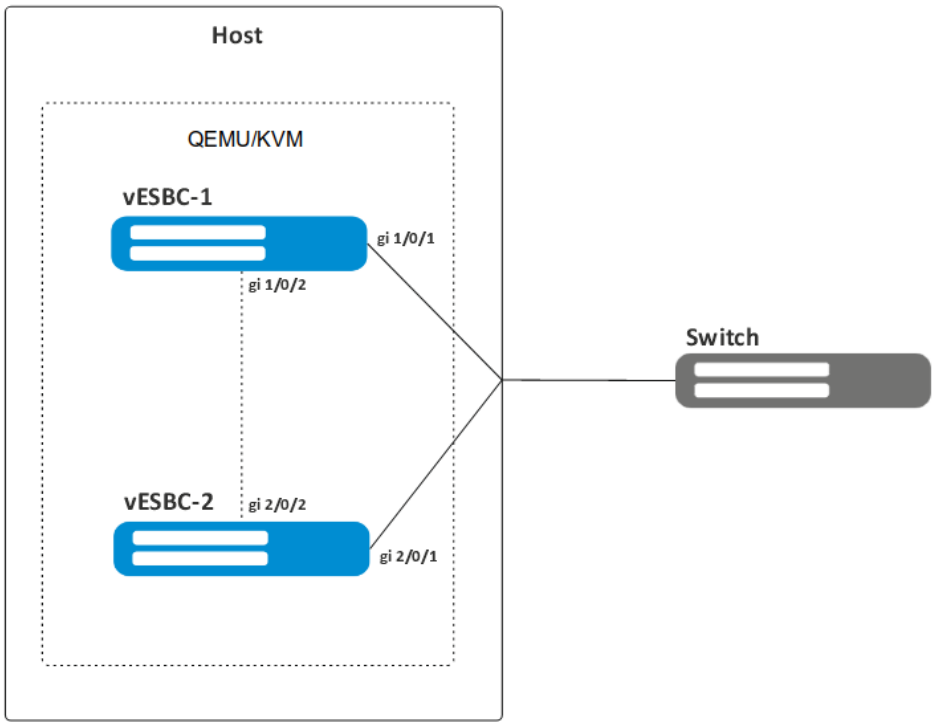
> ⚠ Для объединения в кластер vESBC предварительно необходимо сделать разные системные MAC-адреса на устройствах путем смены серийного номера.

19.2.2 Пример настройки кластера vESBC в гипервизоре QEMU/KVM

Задача:

Развернуть кластер из двух vESBC в среде виртуализации QEMU/KVM на базе ОС Ubuntu 20.04.6 TLS.

Схема:



Решение:

1. Создайте виртуальную машину в QEMU/KVM и установите vESBC удобным для вас способом. Подробное описание создания виртуальной машины и процесс установки vESBC приведены в разделе документации Установка vESBC в системе виртуализации QEMU/KVM.
2. Настройте сетевые интерфейсы:
   - Настройка внешнего интерфейса(ов):
     - Подключите сетевой интерфейс к виртуальной машине в одном из режимов, описанных в документации.

       > **Не используйте** режим интерфейса **macvtap** при организации каластера vESBC, т. к. при использовании протокола VRRP MAC-адрес vESBC будет отличаться от MAC-адреса сетевого интерфейса, и трафик не будет передаваться в виртуальную машину из-за особенностей реализации драйвера macvtap.
   - Настройка кластерного интерфейса:
     - Подключение кластерного интерфейса осуществляется аналогично подключению внешнего интерфейса. Для организации кластерного соединения двух vESBC можно использовать изолированный бридж хоста (без привязки физического интерфейса).
3. Выполните пункты 1-2 для второй виртуальной машины vESBC.

После успешного выполнения описанных пунктов можно приступать к настройке юнитов кластера vESBC. Алгоритм настройки полностью аналогичен настройке аппаратных пограничных контроллеров сессий и описан в разделе Первичная настройка кластера, за исключением конфигурации второго юнита с помощью ZTP.

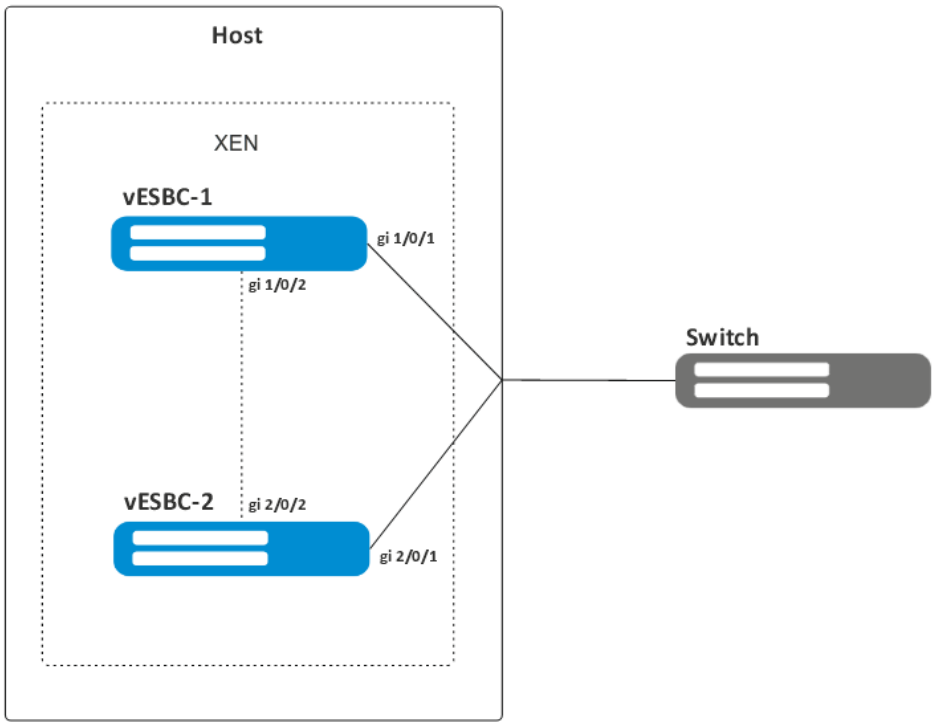
⚠ Для объединения в кластер vESBC предварительно необходимо сделать разные системные MAC-адреса на устройствах путем смены серийного номера.

19.2.3 Пример настройки кластера vESBC в гипервизоре XEN

Задача:

Развернуть кластер из двух vESBC в среде виртуализации XEN на базе ОС Ubuntu 22.04.5 TLS.

Схема:



Решение:

1. Создайте виртуальную машину в XEN и установите vESBC удобным для вас способом. Подробное описание создания виртуальной машины и процесс установки vESBC приведены в разделе документации Установка vESBC в системе виртуализации Xen.
2. Настройте сетевые интерфейсы:
   - Настройка внешнего интерфейса(ов):
     - Подключите сетевой интерфейс к виртуальной машине в одном из режимов, описанных в документации.

       > ⚠ **Не используйте** режим интерфейса **macvtap** при организации каластера vESBC, т. к. при использовании протокола VRRP MAC-адрес vESBC будет отличаться от MAC-адреса сетевого интерфейса, и трафик не будет передаваться в виртуальную машину из-за особенностей реализации драйвера macvtap.
   - Настройка кластерного интерфейса:
     - Подключение кластерного интерфейса осуществляется аналогично подключению внешнего интерфейса. Для организации кластерного соединения двух vESBC можно использовать изолированный бридж хоста (без привязки физического интерфейса).
3. Выполните пункты 1-2 для второй виртуальной машины vESBC.

После успешного выполнения описанных пунктов можно приступать к настройке юнитов кластера vESBC. Алгоритм настройки полностью аналогичен настройке аппаратных пограничных контроллеров сессий и описан в разделе Первичная настройка кластера, за исключением конфигурации второго юнита с помощью ZTP.

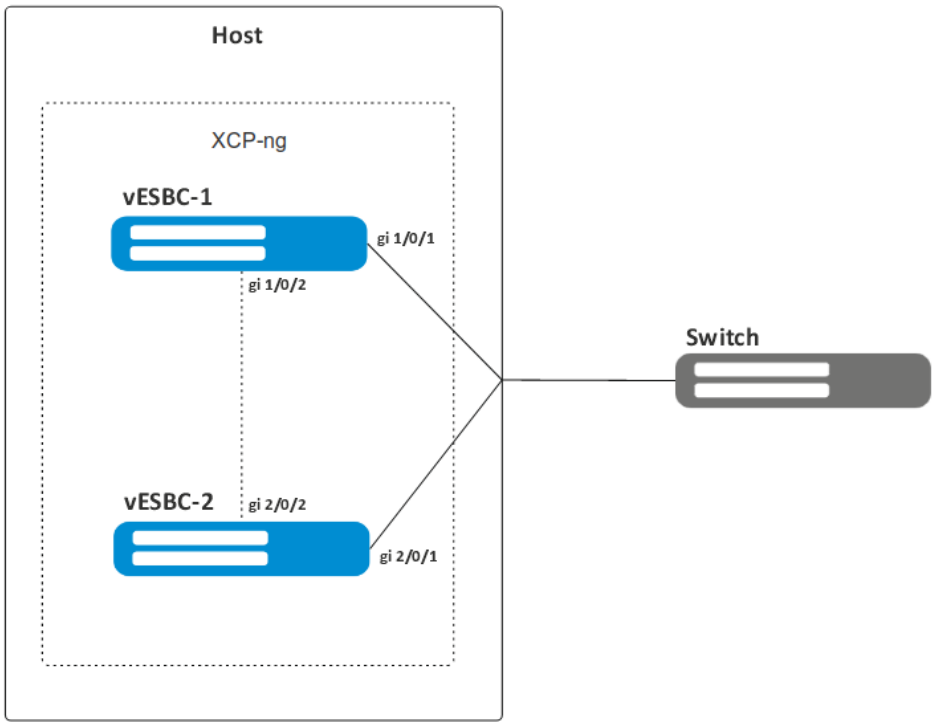
⚠ Для объединения в кластер vESBC предварительно необходимо сделать разные системные MAC-адреса на устройствах путем смены серийного номера.

19.2.4 Пример настройки кластера vESBC в гипервизоре XCP-ng

Задача:

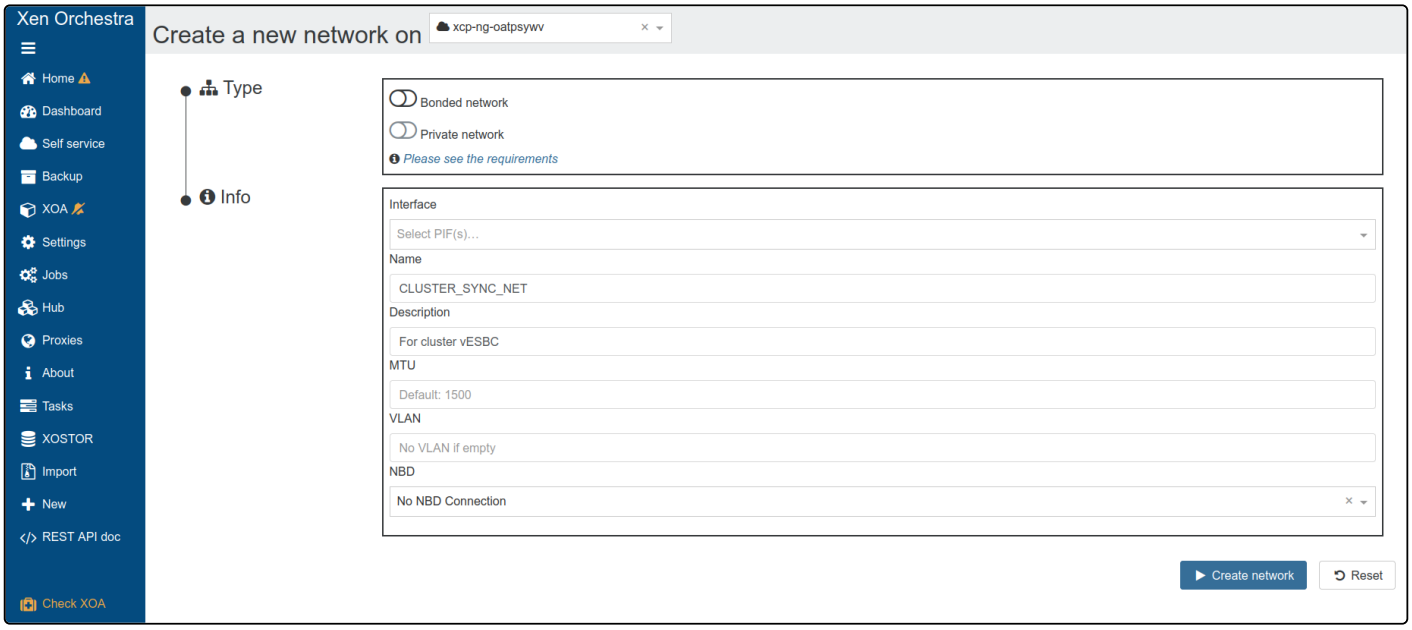
Развернуть кластер из двух vESBC в среде виртуализации XCP-ng 8.3 LTS + XEN Orchestra.

Схема:



Решение:

1. Создайте виртуальную машину в XCP-ng, подробно описано в разделе Установка vESBC в системе виртуализации XCP-ng.
2. Настройте сетевые интерфейсы:
   - Настройка внешнего интерфейса(ов):
     - Подключите сетевой интерфейс к виртуальной машине в одном из режимов, описанных в документации.
   - Настройка кластерного интерфейса:
     - Подключение кластерного интерфейса осуществляется аналогично подключению внешнего интерфейса. Для организации кластерного соединения двух vESBC можно использовать изолированный бридж хоста (без привязки физического интерфейса). Пример создания бриджа через XEN Orchestra:
       - С помощью браузера подключитесь к веб-интерфейсу XEN ORCHESTRA для управления гипервизором XCP-ng. Введите логин и пароль и нажмите "**Sign in with password**"
       - Перейдите в раздел "**+New**" → "**Network**" в левом меню
       - В выпадающем списке "**Create a new network on**" выберите Ваш гипервизор
       - Укажите название сети в поле "**Name**", описание в поле "**Description**" (опционально), выберите "**NBD**" → **No NBD Connection** и нажмите кнопку "**Create network**"



3. Выполните пункты 1-2 для второй виртуальной машины vESBC.

После успешного выполнения описанных пунктов можно приступать к настройке юнитов кластера vESBC. Алгоритм настройки полностью аналогичен настройке аппаратных пограничных контроллеров сессий и описан в разделе Первичная настройка кластера, за исключением конфигурации второго юнита с помощью ZTP.

> ⚠ Для объединения в кластер vESBC предварительно необходимо сделать разные системные MAC-адреса на устройствах путем смены серийного номера.

19.2.5 Особенности настройки гипервизора ESXi для организации кластера vESBC

1. Создайте виртуальную машину в ESXi, подробно описано в разделе Установка vESBC в системе виртуализации VMware ESXi.
2. Для корректной работы протокола VRRP требуются включить **"Promiscuous mode"** и **"Forged transmits"** в разделе **Security** гипервизора;
3. Для предотвращения дублирования пакетов из-за работы "Promiscuous mode" на портах с VRRP, необходимо настраивать на VRRP-интерфейсе подсеть, отличную от подсети на физическом порту (не относится к cluster интерфейсу).
   Пример конфигурации:

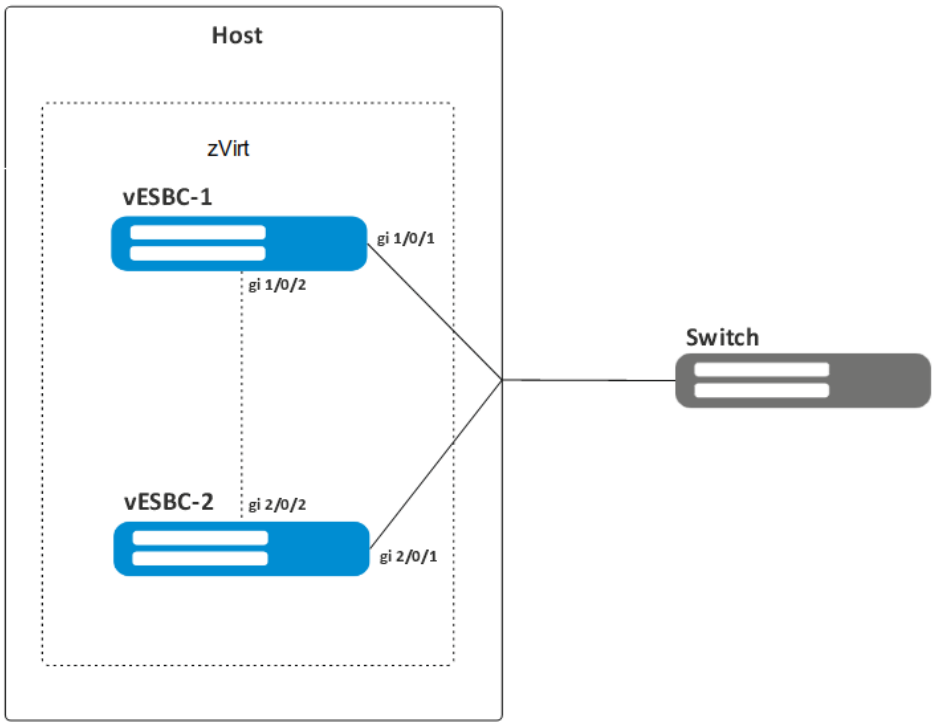
```
interface gigabitethernet 1/0/1
  security-zone WAN
  ip address 1.0.0.1/24              // IP-адрес физического интерфейса
  vrrp 35
    ip address 192.168.34.144/20     // IP-адрес VRRP из другой подсети
    group 1
    enable
  exit
exit
```

19.2.6 Пример настройки кластера vESBC в гипервизоре zVirt

Задача:

Развернуть кластер из двух vESBC в среде виртуализации zVirt 5.0

Схема:

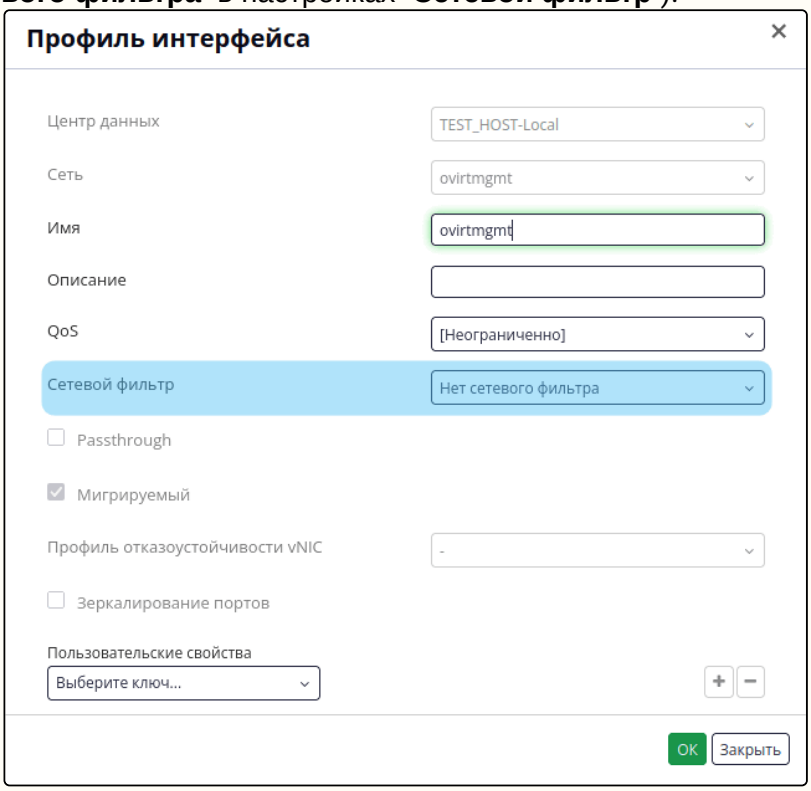


Решение:

1. Создайте виртуальную машину в zVirt, подробно описано в разделе Установка vESBC в системе виртуализации zVirt.
2. Настройте сетевые интерфейсы:

- Настройка внешнего интерфейса(ов):
  - Подключите сетевой интерфейс к виртуальной машине в одном из режимов, описанных в документации.

Для корректной работы протокола VRRP необходимо в настройках профиля вашей сети (**Страница Сеть** → **Профили vNIC** → **<Название профиля>**) отключить все сетевые фильтры (выбрать "**Нет сетевого фильтра**" в настройках "**Сетевой фильтр**").

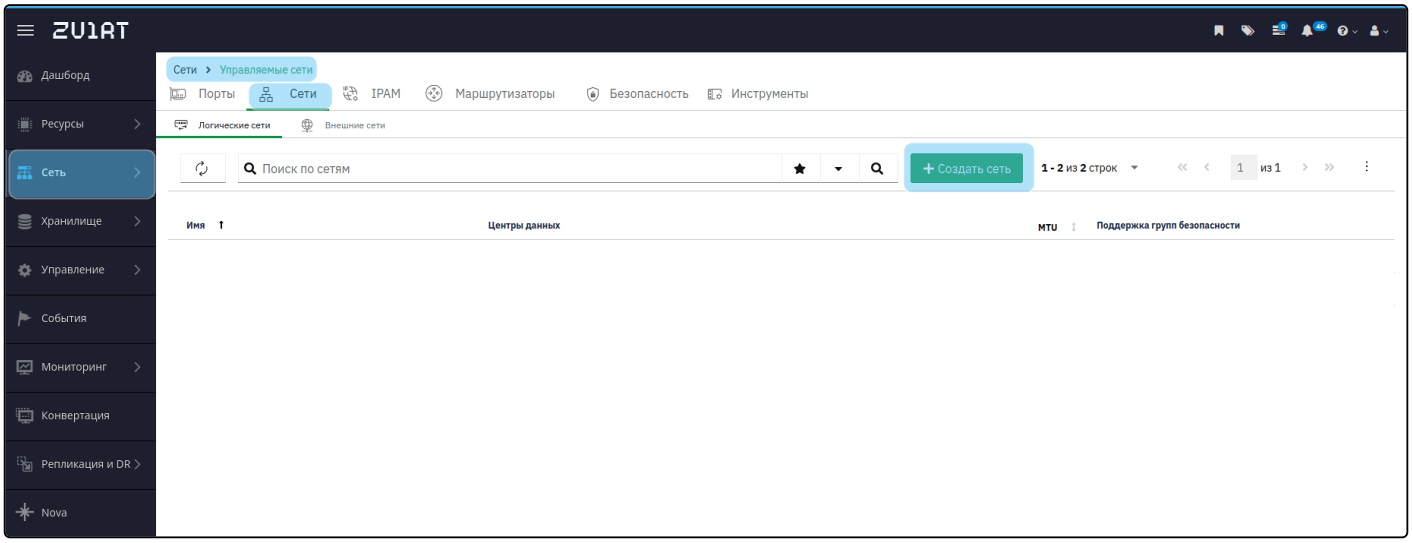


- Настройка кластерного интерфейса:

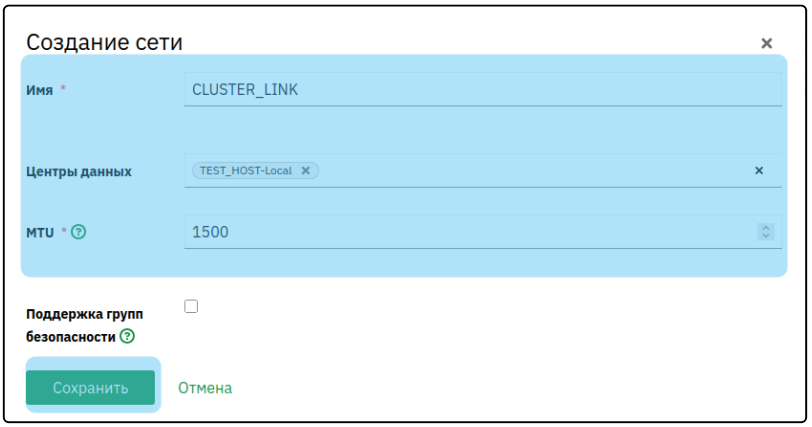
Подключение кластерного интерфейса осуществляется аналогично подключению внешнего интерфейса. Для организации кластерного соединения двух vESBC можно использовать изолированную логическую сеть SDN (без привязки физического интерфейса).

Порядок создания логической сети SDN:

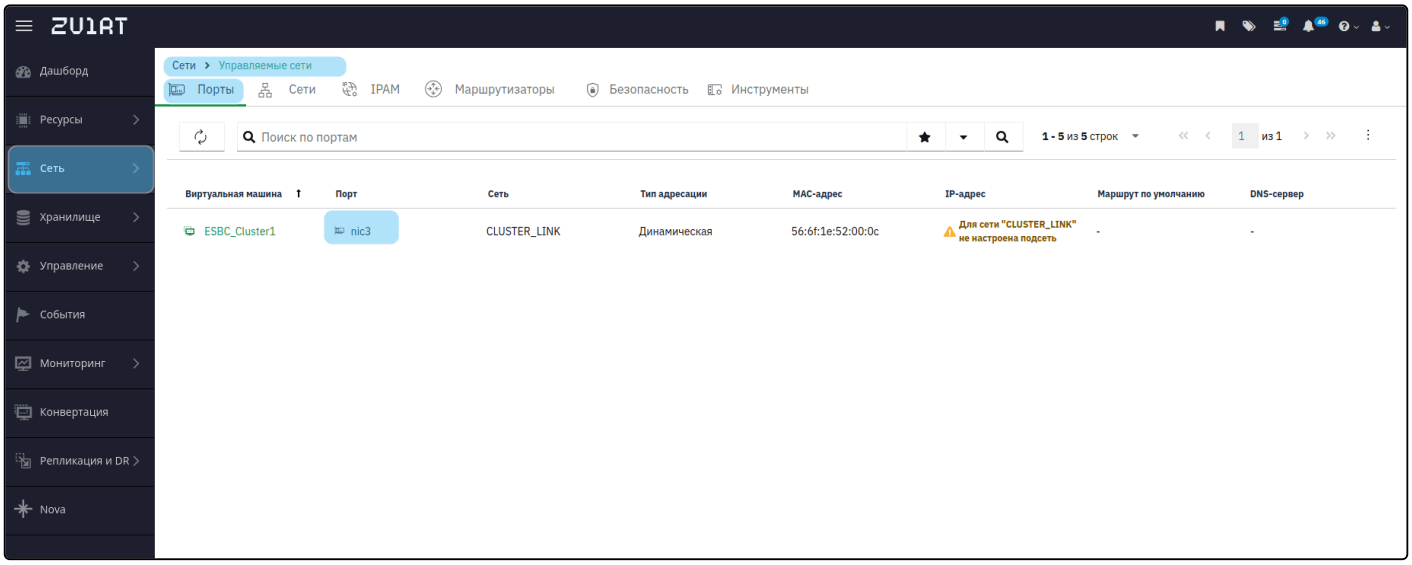
- С помощью браузера подключитесь к веб-интерфейсу zVirt.
- Перейдите в раздел ***"Сеть"*** → ***"Управляемые сети"*** в левом меню.
- Откройте вкладку ***"Сети"*** и нажмите кнопку ***"+Создать сеть"***.



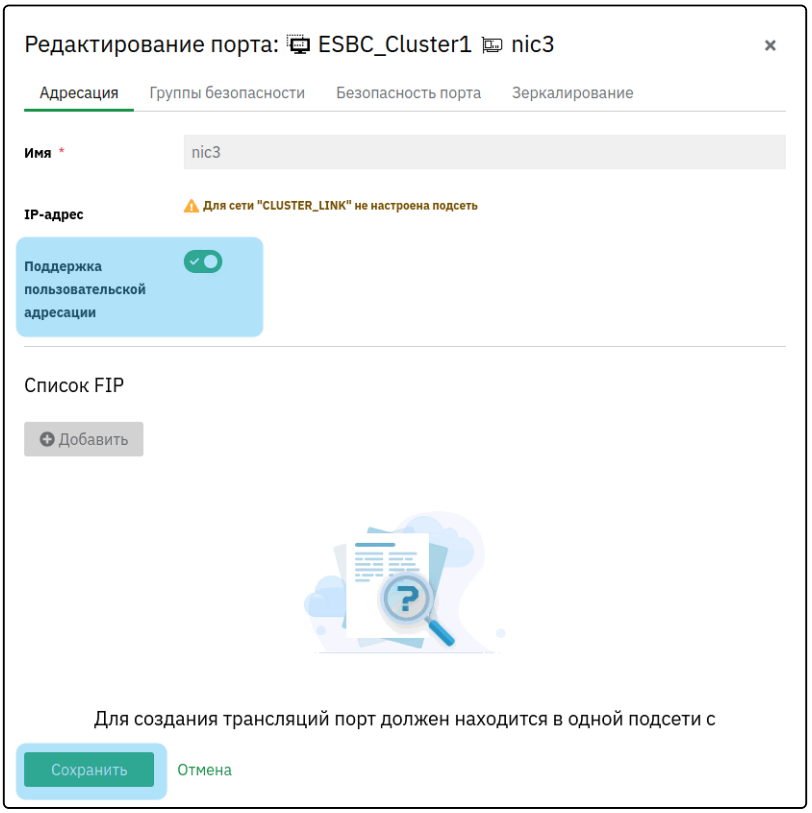
- В открывшемся окне ***"Создание сети"*** укажите название сети, выберите ваш Центр данных, укажите MTU — 1500 и нажмите кнопку ***"Сохранить"***.



- Подключите сетевой интерфейс к виртуальной машине vESBC (см. раздел "Подключение сетевых интерфейсов" Установка vESBC в системе виртуализации zVirt).
- Снова перейдите в раздел ***"Сеть"*** → ***"Управляемые сети"*** в левом меню.
- Откройте вкладку ***"Порты"***, наведите курсор на порт виртуальной машины vESBC и нажмите кнопку ***"Редактировать"***, которая появится справа.



- Включите переключатель ***"Поддержка пользовательской адресации"*** и нажмите кнопку ***"Сохранить"***.



3. Выполните пункты 1–2 для второй виртуальной машины vESBC.

После успешного выполнения описанных пунктов можно приступать к настройке юнитов кластера vESBC. Алгоритм настройки полностью аналогичен настройке аппаратных пограничных контроллеров сессий и описан в разделе Первичная настройка кластера, за исключением конфигурации второго юнита с помощью ZTP.

> ⚠ Для объединения в кластер vESBC предварительно необходимо сделать разные системные MAC-адреса на устройствах путем смены серийного номера.

19.2.7 Подключение второго юнита в кластере vESBC с использованием ZTP

Для активации процесса ZTP необходимо на втором устройстве vESBC запустить dhcp-client на bridge-интерфейсе, логический или физический интерфейс которого будет включен в кластерный интерфейс первого устройства.
В процессе ZTP устройство автоматически выставит себе:
1) Конфигурацию;
2) Юнит;
3) Версию ПО, на котором работает Active ESBC;
4) Лицензию, если она предварительно загружена на Active ESBC.

Алгоритм подключения vESBC с использованием ZTP:

1. Обеспечить сетевую связность (L2) обеих виртуальных машин между собой по всем интерфейсам.
2. Настроить первый юнит по алгоритму Первичная настройка кластера.
3. На втором юните установить конфигурацию по умолчанию и убедиться, что юнит имеет ID 1:

```
vesbc# copy system:default-config system:candidate-config
|*****************************************| 100% (52B) Configuration loaded successfully.
vesbc# show unit id
Unit ID is 1
Unit ID will be 1 after reboot
vesbc# commit
vesbc# confirm
```

4. Перевести в режим switchport интерфейс, включенный в кластерный интерфейс первого юнита:

```
vesbc(config)# interface gigabitethernet 1/0/2
vesbc(config-if-gi)# mode switchport
vesbc(config-if-gi)# exit
```

5. В настройках bridge включить DHCP-клиент:

```
vesbc(config)# bridge 1
vesbc(config-bridge)# vlan 1
vesbc(config-bridge)# ip address dhcp
vesbc(config-bridge)# enable
vesbc(config-bridge)# do commit
vesbc(config-bridge)# do confirm
```

Через несколько секунд после применения изменений начнётся процедура синхронизации:

```
2026-03-30T05:40:12+00:00 %CLUSTER-I-ZTP_INFO: cluster detected by DHCP client
2026-03-30T05:40:12+00:00 %CLUSTER-I-SYNC_SYSTEM_INFO: start system synchronization with Active
unit
2026-03-30T05:40:13+00:00 %FILE_MGR-I-INFO: operation started: 'copy tftp://192.168.16.100:/
ESR-running-config system:candidate-config' (index: 2, origin: Cfgsync-mgr)
2026-03-30T05:40:13+00:00 %FILE_MGR-I-INFO: operation is finished: 'copy tftp://
192.168.16.100:/ESR-running-config system:candidate-config' (index: 2, origin: Cfgsync-mgr)
2026-03-30T05:40:13+00:00 %FILE_MGR-I-INFO: operation started: 'copy tftp://192.168.16.100:/
vESBC-active-firmware system:firmware' (index: 3, origin: Cfgsync-mgr)
2026-03-30T05:40:44+00:00 %FIRMWARE-I-INFO: Verify firmware...
2026-03-30T05:40:45+00:00 %FIRMWARE-I-INFO: Extracting firmware...
2026-03-30T05:40:46+00:00 %FIRMWARE-I-INFO: Remove old image2...
2026-03-30T05:40:49+00:00 %FIRMWARE-I-INFO: Copy kernel to image2 ...
2026-03-30T05:40:49+00:00 %FIRMWARE-I-INFO: Copy rootfs to image2 ...
2026-03-30T05:40:49+00:00 %FIRMWARE-I-INFO: Copy version to image2 ...
2026-03-30T05:40:51+00:00 %FILE_MGR-I-INFO: operation is finished: 'copy tftp://
192.168.16.100:/vESBC-active-firmware system:firmware' (index: 3, origin: Cfgsync-mgr)
2026-03-30T05:40:51+00:00 %FILE_MGR-I-INFO: operation started: 'boot system image-2' (index: 4,
origin: Cfgsync-mgr)
2026-03-30T05:40:51+00:00 %FILE_MGR-I-INFO: operation is finished: 'boot system image-2'
 (index: 4, origin: Cfgsync-mgr)
2026-03-30T05:40:53+00:00 %CLUSTER-I-SYNC_SYSTEM_INFO: system will be rebooted to apply all
changes
```

После перезагрузки второго юнита проверьте статус кластера:

```
vesbc-slave# show cluster status
Unit   Hostname                Role         MAC address         State            IP address
----   --------------------    ----------   -----------------   --------------   ---------------
1      vesbc-master            Active       aa:00:00:03:90:00   Joined           192.168.16.10
2*     vesbc-slave             Standby      aa:00:00:04:10:00   Joined           192.168.16.20

vesbc-slave# show cluster sync status
System part              Synced
----------------------   ------
candidate-config         Yes
running-config           Yes
SW version               Yes
licence                  Yes
licence (After reboot)   Yes
date                     Yes
E-SBC version            Yes
```

Только для vESBC. При подключении второго юнита vESBC с использованием ZTP, синхронизация программного обеспечения и конфигурации занимает до 30 минут, т. к. без лицензии vESBC имеет ограничение скорости полосы пропускания в 1 Мбит/с.

20 Управление удаленным доступом

Алгоритм и примеры настройки функций управления удаленным доступом см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

21 Управление сервисами

Алгоритм и примеры настройки функций управления сервисами см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

22 Мониторинг

Данный раздел см. в документации ESR.

> ⚠ Значения параметров для ESBC-15R/31/3200/3250/3350 идентичны значениям для ESR-15R/31/3200/3250/3350.

23 Управление лицензированием

- Виды лицензий ESBC
  - vESBC
  - ESBC-15R/31/3200/3250/3350
- Способы получения лицензии
- Статусы лицензий
- Лицензирование через ELM
  - Периодичность обращений к ELM
  - Условия преждевременного сброса лицензии
  - Алгоритм настройки
    - Особенности настройки лицензирования на vESBC
  - Пример настройки
- Загрузка и активация файловой лицензии (ESBC-15R/31/3200/3250/3350)
- Лицензирование в кластере
  - Синхронизация файловых лицензий
  - Установка файловых лицензий

Использование лицензирования позволяет гибко управлять производительностью ESBC.

> ⚠ Без использования лицензий ESBC будет работать в демо-режиме. В данном режиме установлено ограничение по количеству одновременно установленных сессий и количеству вызовов в секунду (CPS).
> Для vESBC дополнительно будет установлено ограничение скорости полосы пропускания в 1 Мбит/с.

23.1 Виды лицензий ESBC

23.1.1 vESBC

Название лицензии	Функционал	Ограничения в демо-режиме
ESBC-LIMIT-MAX-CALLS	Ограничение одновременно установленных сессий на ESBC.	6
ESBC-LIMIT-MAX-CPS	Ограничение количества вызовов в секунду на ESBC.	1
ESBC-VIRTUAL-LIMIT-NET	Ограничение скорости полосы пропускания виртуального ESBC.	1 Мбит/с
ESBC-VIRTUAL-LIMIT-DEFAULT	Неизменяемый параметр, предоставляется при выдаче любой лицензии vESBC. Увеличивает лимиты RIB для: • BGP до 65000 • OSPF до 500000 • IS-IS до 500000 • RIP до 10000 Активирует учетную запись techsupport для доступа в режим shell.	Лимиты RIB для: • BGP до 1024 • OSPF до 1000 • IS-IS до 1000 • RIP до 1000 Учетная запись techsupport неактивна
IPS	Система предотвращения вторжений – программная система сетевой и компьютерной безопасности, обнаруживающая вторжения или нарушения безопасности и автоматически защищающая от них.	Функционал недоступен

23.1.2 ESBC-15R/31/3200/3250/3350

Название лицензии	Функционал	Ограничения в демо-режиме
ESBC	Требуется для активации функционала ESBC, поставляется вместе с устройством в заводской комплектации.	-
ESBC-LIMIT-MAX-CALLS	Ограничение одновременно установленных сессий на ESBC.	6
ESBC-LIMIT-MAX-CPS	Ограничение количества вызовов в секунду на ESBC.	1
IPS	Система предотвращения вторжений – программная система сетевой и компьютерной безопасности, обнаруживающая вторжения или нарушения безопасности и автоматически защищающая от них.	Функционал недоступен

23.2 Способы получения лицензии

	Online ELM	Offline ELM	File
vESBC	✓	✓	✗
ESBC-15R	✗	✗	✓
ESBC-31/3200/3250/3350	✓	✗	✓

23.3 Статусы лицензий

Active	Лицензия активна.
Candidate	Лицензия будет применена после перезагрузки.
Unsupported	Лицензия не поддерживается в рамках текущей версии ПО или вообще не поддерживается устройством.

23.4 Лицензирование через ELM

Сервер лицензий Eltex License Manager (далее — ELM), осуществляющий функцию лицензирования программных и аппаратных продуктов компании «Элтекс». ELM используется в процессе активации лицензии и последующей эксплуатации для подтверждения легитимности приобретенного программного обеспечения и предоставления прав на его использование.

Существует 2 варианта работы с ELM:

- *Online ELM* — сервер лицензирования, расположенный на стороне компании «Элтекс». Установка дополнительного ПО не требуется.

Адрес ELM
`https://elm.eltex-co.ru:8099`

- *Offline ELM* — сервер лицензирования, устанавливаемый локально на стороне заказчика. Подходит для эксплуатации в закрытом контуре. Подробная информация об Offline ELM доступна в официальной документации Offline ELM.

> ✗ В обоих вариантах необходимо, чтобы между сервером и устройством была сетевая доступность.

23.4.1 Периодичность обращений к ELM

1. При включении менеджера лицензирования на ESBC устройство сразу попытается обратиться за лицензией:

- в случае успешного обращения устройство начнет обращаться к серверу раз в час,
- в случае неуспешного обращения устройство будет пытаться получить лицензию раз в 15 минут.

2. В случае если менеджер лицензирования был включен на момент перезапуска устройства, то после перезапуска устройство сразу обратится за лицензией. При неудачной попытке:

- устройство будет пытаться получить лицензию раз в 5 минут (до сброса лицензии), если до перезапуска имелась действующая ELM-лицензия,
- устройство будет пытаться получить лицензию раз в 15 минут, если на момент перезапуска ELM-лицензии не было.

3. В случае сброса лицензии устройство будет пытаться получить лицензию раз в 15 минут.

23.4.2 Условия преждевременного сброса лицензии

Лицензия на устройстве будет преждевременно сброшена, если:

1. был отключен менеджер лицензирования на устройстве;
2. устройство не смогло в течение 168 часов успешно обратиться на ELM;
3. устройство после перезапуска не смогло в течение 15 минут обратиться на ELM.

Чтобы снова получить лицензию, необходимо успешное обращение устройства на ELM.

23.4.3 Алгоритм настройки

Особенности настройки лицензирования на vESBC

Перед подключением vESBC к серверу лицензирования (ELM) необходимо указать серийный номер (serial number) и ключ лицензии (licence key) в системных настройках устройства. Выполнить это можно следующими командами:

```
vesbc# set system serial-number <SERIAL-NUMBER>
vesbc# set system licence-key <LICENCE-KEY>
```

где:
<SERIAL-NUMBER> – серийный номер вида ESBCXXXXXXX;
<LICENCE-KEY> – ключ лицензии, полученный при заказе vESBC.

После чего устройство необходимо будет перезагрузить.

Не включайте менеджер лицензирования до задания серийного номера и ключа лицензии. После 10 попыток подключения к серверу лицензирования с некорректными учётными данными ваш IP-адрес будет автоматически заблокирован системой защиты сервера лицензирования.

При миграции, копировании, восстановлении, экспорте или других аналогичных операциях, связанных с состоянием виртуальной машины, требуется перезагрузка vESBC командой *reload system*.

Настройка менеджера лицензирования через web-интерфейс описана в разделе Меню «Лицензии» документации ESBC-Series. Управление через web-интерфейс. Версия 1.9.0.

Шаг	Описание	Команда	Ключи
1	Перейти в конфигурирование менеджера лицензирования.	**esbc (config)# licence-manager**	

Шаг	Описание	Команда	Ключи
2	Задать IP-адрес ELM-сервера.	**esbc (config-licence-manager)# host address <A.B.C.D | WORD | X:X:X:X::X>**	<IP-ADDR> – IP-адрес, задаётся в виде AAA.BBB.CCC.DDD, где каждая часть принимает значения [0..255]; <IPV6-ADDR> – IPv6-адрес, задаётся в виде X:X:X:X::X, где каждая часть принимает значения в шестнадцатеричном формате [0..FFFF]. WORD(1-31) – DNS-имя сервера.
3	Задать порт для подключения к ELM-серверу (необязательно).	**esbc (config-licence-manager)# host port <PORT>**	<PORT> – номер TCP/UDP-порта отправителя, принимает значения [1..65535], по умолчанию 8099.
4	Включить менеджер лицензирования.	**esbc (config-licence-manager)# enable**	
5	Установить описание (необязательно).	**esbc (config-licence-manager)# description <LINE>**	<LINE> – описание, задаётся строкой до 255 символов.
6	Задать текстовое имя устройства, которое передаётся на сервер ELM (необязательно).	**esbc (config-licence-manager)# system-name <WORD>**	<WORD> – имя, задаётся строкой до 255 символов.
7	Задать текстовое описание, которое передаётся на сервер ELM (необязательно).	**esbc (config-licence-manager)# location <WORD>**	<WORD> – описание, задаётся строкой до 255 символов.
8	Указать экземпляр VRF, в котором будет работать менеджер лицензирования (необязательно).	**esbc (config-licence-manager)# ip vrf forwarding <VRF>**	<VRF> – имя VRF, задается строкой до 31 символа.
9	Задать значение DSCP, которое будет использоваться для маркировки трафика при обращениях к ELM-серверу (необязательно).	**esbc (config-licence-manager)# dscp <DSCP>**	<DSCP> – значение DSCP, принимает значения [0..63], по умолчанию 48.

23.4.4 Пример настройки

В режиме конфигурирования licence-manager укажем адрес и порт ELM-сервера. Между устройством и ELM-сервером должна быть сетевая доступность.

```
vesbc# configure
vesbc(config)# licence-manager
vesbc(config-licence-manager)# host address elm.eltex-co.ru
vesbc(config-licence-manager)# enable
vesbc(config-licence-manager)# end
```

После включения менеджера лицензирования устройство попытается обратиться на указанный сервер.

Команда **show licence-manager status** выведет подробную информацию об обращениях на сервер лицензирования:

```
vesbc# show licence-manager status
ELM server type:                root
Last request status:            success
Last request to licence server: 2025-04-17 10:24:22
Next request to licence server: 2025-04-17 10:24:43
```

Команда **update licence-manager licence** позволит принудительно отправить запрос на ELM-сервер, не дожидаясь времени следующего запроса:

```
vesbc# update licence-manager licence
2024-08-12T04:01:00+00:00 %LICENCE-W-EVENT: Licence recieved from Eltex Licence Manager server
```

Команда **show licence** позволит вывести информацию о лицензиях на устройстве:

```
vesbc# sh licence
System serial number:  ESBC0000051
System licence key:    XXXXXXXXXXX

ELM licence:
    Licence is valid from: --
    Licence expires:       --

Feature                      Source    State     Value         Valid from  Expiries
---------------------------  --------  --------  -----------   ----------  --------
ESBC-LIMIT-MAX-CALLS         ELM       Active    50000         --          --
ESBC-LIMIT-MAX-CPS           ELM       Active    1000          --          --
ESBC-VIRTUAL-LIMIT-DEFAULT   ELM       Active    true          --          --
ESBC-VIRTUAL-LIMIT-NET       ELM       Active    10000000000   --          --
IPS                          ELM       Active    true          --          --
```

23.5 Загрузка и активация файловой лицензии (ESBC-15R/31/3200/3250/3350)

Загрузка файловой лицензии через web-интерфейс описана в разделе Меню «Лицензии» документации ESBC-Series. Управление через web-интерфейс. Версия 1.9.0.

Для загрузки лицензии через CLI введите одну из нижеописанных команд. В качестве параметра *<server>* должен быть указан IP-адрес используемого сервера. Для обновления с FTP- или SCP-сервера потребуется ввести имя пользователя (параметр *<user>*) и пароль (параметр *<password>*). В качестве параметра *<file_name>* укажите имя файла лицензии, помещенного на сервер (при использовании SCP нужно указать полный путь – параметр *<folder>*). После ввода команды устройство скопирует файл во внутреннюю память, проверит целостность данных и сохранит его. В качестве примера приведен ESBC-3200.

TFTP:

```
ESBC-3200# copy tftp://<server>:/<file_name> system:licence
```

FTP:

```
ESBC-3200# copy ftp://[<user>[:<password>]@]<server>:/<file_name> system:licence
```

SCP:

```
ESBC-3200# copy scp://[<user>[:<password>]@]<server>://<folder>/<file_name> system:licence
```

SFTP:

```
ESBC-3200# copy sftp://[<user>[:<password>]@]<server>:/<file_name> system:licence
```

Пример загрузки лицензии через SCP:

```
ESBC-3200# copy scp://adm:password123@192.168.16.168://home/tftp/licence system:licence
|****************************************| 100% (670B) Licence loaded successfully. Please
reboot system to apply changes.
```

Для активации лицензии формата "File (v1)" необходимо перезагрузить устройство:

```
ESBC-3200# reload system
```

После перезагрузки проверьте, что лицензия активирована:

```
ESBC-3200# show licence
Feature                  Source     State         Value    Valid from    Expiries
-----------------------  --------   -----------   -------  ------------  --------
ESBC                     Boot       Active        true         --           --
ESBC                     Boot       Candidate     true         --           --
ESBC-LIMIT-MAX-CALLS     File (v1)  Active        8500         --           --
ESBC-LIMIT-MAX-CALLS     File (v1)  Candidate     8500         --           --
ESBC-LIMIT-MAX-CPS       File (v1)  Active        400          --           --
ESBC-LIMIT-MAX-CPS       File (v1)  Candidate     400          --           --
```

Файловые лицензии формата "File (v2)" применяются при копировании на устройство автоматически, поэтому перезагрузка ESBC не требуется.

23.6 Лицензирование в кластере

23.6.1 Синхронизация файловых лицензий

Для синхронизации файлов лицензий в кластере необходимо загрузить их на Active-устройство командой *copy* в директорию **system:cluster-unit-licences**.

Все загруженные лицензии в данной директории передаются остальным участникам кластера.

Пример

```
ESBC-1# copy tftp://<IP_address>:/licence system:cluster-unit-licences
|*************************| 100% (680B) Licence loaded successfully.
```

> ⚠ На каждый ESBC нужна отдельная лицензия. Для активации функций кластера не нужна отдельная лицензия.

23.6.2 Установка файловых лицензий

Установить лицензию в кластере можно двумя способами:

1. Загрузить индивидуально лицензию на каждое устройство, как в случае с обычным ESBC вне кластера.
2. Загрузить лицензию для Active-юнита в **system:licence** (данная лицензия также автоматически загрузится и в **system:cluster-unit-licences**), активировать её перезагрузкой, лицензии для Standby загрузить в **system:cluster-unit-licences** на Active-юните, после чего либо выполнить команду *sync cluster system force*, либо подключить Standby по ZTP.

Пример

```
ESBC-1# copy tftp://<IP_address>:/licence system:cluster-unit-licences
|*************************| 100% (680B) Licence loaded successfully.
ESBC-1#
ESBC-1#
ESBC-1#
ESBC-1# show cluster-unit-licences
Serial number     Features
---------------   ------------------------------------------------------------
VIBE000065        ESBC,ESBC-LIMIT-MAX-CPS,ESBC-LIMIT-MAX-CALLS
VIBE000033        ESBC,ESBC-LIMIT-MAX-CPS,ESBC-LIMIT-MAX-CALLS
ESBC-1# sync cluster system force
```

24 Часто задаваемые вопросы

Закрываются сессии SSH/Telnet, проходящие через пограничный контроллер сессий ESBC

Для поддержания сессии активной необходимо настроить передачу keepalive-пакетов. Опция отправки keepalive настраивается в клиенте SSH, например, для клиента PuTTY раздел “Соединение”.

В свою очередь, на пограничном контроллере сессий можно выставить время ожидания до закрытия неактивных сессий TCP (в примере выставлен 1 час):

```
esbc(config)# ip firewall sessions tcp-estabilished-timeout 3600
```

На интерфейсе был отключен firewall (ip firewall disable). После внесения этого интерфейса в security zone, удаления из конфигурации ip firewall disable и применения изменений – доступ для активных сессий с данного порта не закрылся согласно правилам security zone-pair

Изменения в конфигурации Firewall будут действовать только для новых сессий, сброса активных сессий в Firewall не происходит. Отчистить активные сессии в firewall можно командой:

```
esbc# clear ip firewall session
```

Как полностью очистить конфигурацию ESBC и как сбросить на заводскую конфигурацию?

Очистка конфигурации происходит путем копирования пустой конфигурации в candidate-config и применения его в running-config.

```
esbc# copy system:default-config system:candidate-config
```

Процесс сброса на заводскую конфигурацию аналогичен.

```
esbc# copy system:factory-config system:candidate-config
```

В случае невозможности аутентификации на пограничном контроллере сессий (неизвестен логин/пароль) конфигурацию можно сбросить к заводской следующим образом:

1. дождаться полной загрузки устройства
2. зажать функциональную кнопку "F" на 15 секунд
3. отпустить функциональную кнопку "F"
4. дождаться полной загрузки устройства с заводской конфигурацией

Как привязать subinterface к созданным VLAN?

При создании саб-интерфейса VLAN создается и привязывается автоматически (прямая зависимость индекс sub – VID).

```
esbc(config)# interface gigabitethernet 1/0/1.100
```

После применения можно наблюдать информационные сообщения:

```
2016-07-14T012:46:24+00:00 %VLAN: creating VLAN 100
```

Есть ли функционал в пограничном контроллере серии ESBC для анализа трафика?

В пограничных контроллерах сессий серии ESBC реализована возможность анализировать трафик на интерфейсах из CLI. Сниффер запускается командой monitor.

```
esbc# monitor gigabitethernet 1/0/1
```

Как настроить ip prefix-list 0.0.0.0/0?

Ниже приведен пример конфигурации префикс-листа, разрешающего прием маршрута по умолчанию.

```
esbc(config)# ip prefix-list eltex
esbc(config-pl)# permit 0.0.0.0/0
```

Проблема прохождения асинхронного трафика

В случае организации сети с асинхронной маршрутизацией, Firewall будет запрещать "неправильный (ошибочный)" входящий трафик (не открывающий новое соединение и не принадлежащий никакому установленному соединению) из соображений безопасности.

Разрешающее правило в Firewall не решит поставленную задачу для подобных схем.

Решить задачу можно, отключив Firewall на входном интерфейсе:

```
esbc(config-if-gi)# ip firewall disable
```

Как можно сохранить локальную копию конфигурации пограничного контроллера сессий?

Если необходимо скопировать текущую running или candidate – конфигурацию на самом пограничном контроллере сессий – можно воспользоваться командой copy с указанием в качестве источника копирования "system:running-config" или "system:candidate-config", а в качестве назначения – файл в разделе "flash:data/".

```
esbc# copy system:candidate-config flash:data/temp.txt
```

Также существует возможность копирования ранее сохраненных конфигурационных файлов (автоматически из раздела flash:backup/ или вручную из раздела flash:data/) в candidate-конфигурацию:

```
esbc# copy flash:data/temp.txt system:candidate-config
esbc# copy flash:backup/config_20190918_164455 system:candidate-config
```

25 Приложение A. Packet Flow

- Порядок обработки входящего/исходящего трафика сетевыми службами пограничного контроллера сессий ESBC
- Порядок обработки транзитного трафика сетевыми службами пограничного контроллера сессий ESBC

25.1 Порядок обработки входящего/исходящего трафика сетевыми службами пограничного контроллера сессий ESBC

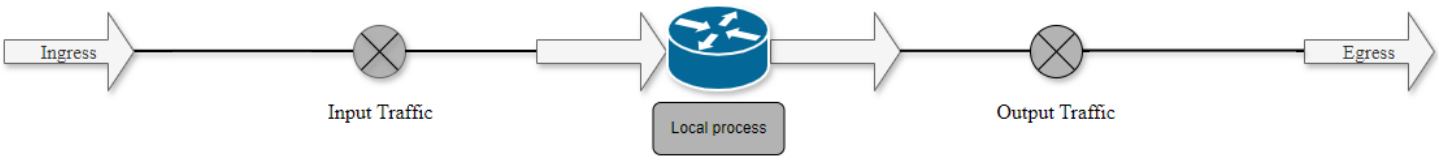


Таблица 1 – Порядок обработки входящего трафика

Шаг	Описание
1	Выполнение функций ACL на входящем трафике
2	Выполнение функций QoS (ограничение полосы пропускания, классификация и т. д.)
3	Выполнение функций DOS defense[1]. На данном этапе выполняются функции защиты от DDOS из разделов firewall screen dos-defense, firewall screen spy-blocking, firewall screen suspicious-packets
4	Инспектирование пакета сервисом IDS/IPS в режиме service-ips monitor[1]
5	Отключение функций Firewall командой ip firewall disable. Разрешение трафика исключает проверки на этапах 6, 13, 15
6	Выполнение правил между зонами any/self
7	Выполнение дефрагментации пакета
8	Функции Destination NAT[1]
9	Routing Decision (FIB)
10	Выполнение функций DOS defense[1]. На этапе данном этапе выполняются специфичные функции защиты от DDOS из разделов firewall screen dos-defense, firewall screen spy-blocking, firewall screen suspicious-packets: ip firewall screen suspicious-packets large-icmp ip firewall screen dos-defense winnuke ip firewall screen spy-blocking port-scan
11	Выполнение правил между пользовательскими зонами / self
12	Разрешение служебного трафика кластера[1]
13	Передача пакета в DPI[1]

14	Передача пакета в Netflow/sFlow (Ingress)[1]
15	IPsec (decode)[1]. После выполнения этого шага происходит переход к п.1

Таблица 2 – Порядок обработки исходящего трафика

Шаг	Описание
1	Local Policy Based Routing[1]
2	Route Decision
3	Передача пакета в DPI[1]
4	tcp adjust-mss[1]
5	Netflow/sFlow (Egress)[1]
6	Выполнение функций Source NAT[1]
7	IPsec (encode)[1]
8	Выполнение фрагментации пакетов
9	Выполнение функций QoS (ограничение полосы пропускания, классификация и т. д.)

[1]Данная функция выполняется только при наличии необходимых настроек.

25.2 Порядок обработки транзитного трафика сетевыми службами пограничного контроллера сессий ESBC

Таблица 3 – Порядок обработки транзитного трафика

Шаг	Описание
1	Выполнение функций ACL на входящем трафике
2	Выполнение функций QoS (ограничение полосы пропускания, классификация и т. д.)
3	Выполнение функций DOS defense[1] На данном этапе выполняются функции защиты от DDOS из раздела firewall screen dos-defense, firewall screen spy-blocking, firewall screen suspicious-packets
4	Отключение функций Firewall командой ip firewall disable. Разрешение трафика исключает проверки на этапах 5, 15, 16
5	Выполнение правил между пользовательскими зонами / any
6	Выполнение дефрагментации пакета
7	Функции Destination NAT[1]
8	Policy Based Routing
9	Routing Decision (FIB)
Если пакет перед передачей необходимо обработать протоколом более высокого уровня, выполняются следующие действия:	
9.1	Выполнение функций DOS defense[1]. На данном этапе выполняются специфичные функции защиты от DDOS из разделов firewall screen dos-defense, firewall screen spy-blocking, firewall screen suspicious-packets: ip firewall screen suspicious-packets large-icmp ip firewall screen dos-defense winnuke ip firewall screen spy-blocking port-scan
9.2	Передача пакета в DPI[1]
9.3	Передача пакета в Netflow/sFlow (Ingress)[1]
9.4	IPsec (decode)[1] После выполнения этого шага происходит переход к п.1

Шаг	Описание
10	tcp adjust-mss[1]
11	Выполнение функций DOS defense[1] На данном этапе выполняются специфичные функции защиты от DDOS из разделов firewall screen dos-defense, firewall screen spy-blocking, firewall screen suspicious-packets: ip firewall screen suspicious-packets large-icmp ip firewall screen dos-defense winnuke ip firewall screen spy-blocking port-scan
12	Выполнение правил между специальными зонами, any/any
13	Передача пакета в DPI[1]
14	Разрешение трафика, исходящий порт которого voice-port. Разрешение трафика включается только в случае наличия настроенного voice-port
15	Netflow/sFlow (Egress)[1]
16	Инспектирование пакета сервисом IPS/IDS в режиме service-ips inline[1]
17	Выполнение функций Source NAT[1]
18	IPsec (encode)[1]
Если необходимо шифрование, то после этого процесса, выполняются следующие операции:	
18.1	Передача пакета в DPI[1]
18.2	tcp adjust-mss[1]
18.3	Netflow/sFflow (Egress)[1]
18.4	Выполнение функций Source NAT[1]
19	Выполнение фрагментации пакетов
20	Выполнение функций QoS (ограничение полосы пропускания, классификация и т. д.)

[1]Данная функция выполняется только при наличии необходимых настроек.

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Для получения технической консультации по вопросам эксплуатации оборудования ООО «Предприятие «ЭЛТЕКС» вы можете обратиться в Сервисный центр компании:

Форма обратной связи на сайте: https://eltex.ru/support/

Servicedesk: https://servicedesk.eltex-co.ru

На официальном сайте компании вы можете найти техническую документацию и программное обеспечение для продукции ООО «Предприятие «ЭЛТЕКС», обратиться к базе знаний, оставить интерактивную заявку:

Официальный сайт компании: https://eltex.ru

База знаний: https://docs.eltex-co.ru/display/EKB/Eltex+Knowledge+Base

Центр загрузок: https://eltex.ru/download