Общество с ограниченной ответственностью
«Научно-Технический Центр ПРОТЕЙ»
(ООО «НТЦ ПРОТЕЙ»)

СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «ПРОТЕЙ-SBC»

ПОГРАНИЧНЫЙ КОНТРОЛЛЕР СЕССИЙ «ОПЕРАТОР-ОПЕРАТОР»

РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

RUS.ПАМР.49300-01 32

Листов 110

2021

Подпись и дата	
Инв. № дубл.	
Взам. инв. №	
Подпись и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата

Литера ___

Аннотация

Настоящий документ «Специальное программное обеспечение «ПРОТЕЙ-SBC». Пограничный контроллер сессий «Оператор-оператор». Руководство системного программиста» разработан на Специальное программное обеспечение «ПРОТЕЙ-SBC» (далее — ПРОТЕЙ-SBC, SBC) производства Общества с ограниченной ответственностью «Научно-Технический Центр ПРОТЕЙ» (далее — ООО «НТЦ ПРОТЕЙ»). Настоящий документ предназначен для подачи в Минцифры России вместе с заявлением о внесении сведений о программном обеспечении ПРОТЕЙ-SBC в единый реестр российских программ для электронных вычислительных машин и баз данных.

Настоящий документ содержит сведения для выполнения обязанностей системного программиста и настройки ПРОТЕЙ-SBC, а также указания по установке программного обеспечения.

Настоящий документ построен на основании стандартов ООО «НТЦ ПРОТЕЙ».

Авторские права

Без предварительного письменного разрешения, полученного от ООО «НТЦ ПРОТЕЙ», настоящий документ и любые выдержки из него, с изменениями и переводом на другие языки, не могут быть воспроизведены или использованы.

СОДЕРЖАНИЕ

1 Термины и сокращения 6
2 Общие сведения 8
2.1 Назначение документа 8
2.2 Состав документа 8
2.3 Техническая поддержка 10
2.3.1 Производитель 10
2.3.2 Служба технической поддержки 10
3 Описание оборудования 11
3.1 Функциональные возможности 11
3.2 Структура сети связи с участием SBC 12
4 Пользовательский интерфейс приложения cli 16
4.1 Вызов приложения cli 16
4.2 Работа с приложением cli 17
4.3 Управляющие клавиши 18
4.4 Управляющие команды 18
4.4.1 Навигация 20
4.4.2 Создание и удаление объектов 21
4.4.3 Настройка параметров 22
4.4.4 Операции над объектами 25
4.4.5 Операции с массивами 25
4.4.6 Отображение конфигурации и состояния объектов 28
4.4.7 Применение и восстановление конфигурации 29
5 Настройка SBC 31
5.1 Раздел sip 32
5.2 Основные параметры SIP 33
5.3 Виртуальные шлюзы 34
5.4 Таймеры SIP-сигнализации 35
6 Раздел sbc 38

6.1 Основные параметры sbc 39
6.2 Оператор 39
6.2.1 Правила маршрутизации для оператора 43
6.2.2 Медиашлюзы оператора 46
6.3 Сервисная платформа 48
6.4 Шлюзы сервисной платформы 51
6.5 MCU 54
6.6 Подсети MCU 56
6.7 Медиапрофили 58
6.7.1 Списки аудиокодеков 59
6.7.2 Списки видеокодеков 61
6.8 Аудиокодеки 63
6.9 Видеокодеки 64
6.10 Маршрутизация и модификация контекста 66
6.10.1 Общие причины для перемаршрутизации 68
6.10.2 Шаблоны масок для правил маршрутизации 69
6.10.3 Шаблоны действий 70
6.10.4 Правила перемаршрутизации 71
6.10.5 Правила маршрутизации 73
6.10.6 Подсистема Flexible Direction 78
6.11 Общие глобальные параметры 79
6.12 Просмотр состояния 81
6.12.1 Состояние маршрута 82
6.12.2 Состояния MCU 83
6.12.3 Состояние оператора 84
7 Описание журналов 87
7.1 Журнал CDR 87
7.1.1 Формат CDR-файлов 89
7.1.2 RTP-статистика в плече 92

7.2 Диагностика 93
7.2.1 Формат файла диагностики 94
7.2.2 Коды причин и строки диагностики 95
8 Приложение 97
8.1 Алгоритм внутренней маршрутизации 97
8.1.1 Фаза 1: Определение источника вызова 98
8.1.2 Фаза 2: Маршрутизация и модификация контекста 99
8.1.3 Фаза 3: Определение получателя вызова 100
8.1.4 Перемаршрутизация 101
8.2 Типовые задачи конфигурирования I-SBC 102
8.2.1 Создать и настроить медиашлюз 102
8.2.2 Настроить MCU 103
8.2.3 Создать и настроить сервисную платформу 104
8.2.4 Создать и настроить оператора 105
8.2.5 Настроить кодеки в медиапрофиле 107
8.2.6 Переадресовать вызовы к сервисной платформе 108

1 Термины и сокращения

В таблице 1 приведены используемые в настоящем документе термины и сокращения.

Таблица 1 — Используемые термины и сокращения

Термин	Описание
ALG	Application-Level Gateway (англ. «шлюз прикладного уровня»)
AOR	Address of Record (англ. «адрес регистрации»)
API	Application Programming Interface (англ. «программный интерфейс приложения»)
CdPN	Called Party Number (англ. «номер вызываемого абонента»)
CDR	Call Detail Record (англ. «запись с подробной информацией о вызове»)
CgPN	Calling Party Number (англ. «номер вызывающего абонента»)
cps	Calls per Second (англ. «количество вызовов в секунду»)
HTTP	HyperText Transfer Protocol (англ. «протокол передачи гипертекста»)
ID	Identifier (англ. «идентификатор»)
ISDN	Integrated Services Digital Network (англ. «цифровая сеть комплексного обслуживания») — набор стандартов для передачи медиафайлов по телефонным сетям.
MCU	Multipoint Control Unit — устройство, предназначенное для объединения аудио– и видеоконференции в многоточечный режим.
NGN	Next Generation Networks (англ. «сети нового поколения») — мультисервисные сети связи.
QoS	Quality of Service (англ. «качество обслуживания»)
Realm	Строковый параметр авторизации, вместе с адресом сервера позволяет разбивать ресурсы, расположенные на сервере, на части и для каждой части установить отдельные правила авторизации.
RFC	Request for Comments (англ. «рабочее предложение») — документы, содержащие общепринятые технические спецификации и стандарты.
rps	Requests per Second (англ. «количество запросов в секунду»)
RTCP	Real-Time Transport Control Protocol (англ «протокол управления передачей трафика в режиме реального времени»)
RTP	Real-Time Transport Protocol (англ. «протокол передачи трафика в режиме реального времени»)
RTT	Round-Trip Time (англ. «время приема-передачи»)

Термин	Описание
SCTP	Stream Control Transmission Protocol (англ. «протокол передачи и управления потоком»)
SDP	Session Description Protocol (англ. «протокол описания сессии»)
SG	Signalling (англ. «сигнализация»)
SI	Socket Interface (англ. «интерфейс на основе сокетов») — программный интерфейс обмена данными с использованием протоколов TCP/IP.
SIP	Session Initiation Protocol (англ. «протокол установления сессии»)
SNMP	Simple Network Management Protocol (англ. «простой протокол сетевого управления»)
TCP	Transmission Control Protocol (англ. «протокол управления передачей»)
ToS	Type of Service (англ. «тип обслуживания»)
UDP	User Datagram Protocol (англ. «протокол пользовательских датаграмм»)
URI	Uniform Resource Identifier (англ. «унифицированный идентификатор ресурса»)
VoIP	Voice over Internet Protocol (англ. «голосовые вызовы на базе интернет-протоколов») — телефонная связь по протоколу IP.
СКО	Среднеквадратичное отклонение
СОРМ	Система оперативно-розыскных мероприятий

Внимание!

Перед установкой и началом эксплуатации изделия необходимо внимательно ознакомиться с паспортом изделия и эксплуатационной документацией.

Данный документ должен постоянно находиться при изделии.

2 Общие сведения

2.1 Назначение документа

ПРОТЕЙ-SBC поставляется в двух модификациях — I-SBC (для функционирования на границах сетей связи) и A-SBC (для обеспечения работы между сетью оператора и конечными пользователями).

Данный документ является руководством системного программиста для модификации I-SBC.

Документ предназначен для использования сотрудниками технической поддержки и системными программистами для настройки и сопровождения пограничного контроллера сессий (SBC).

2.2 Состав документа

Настоящее руководство состоит из следующих основных частей:

1. «Термины и сокращения» — раздел, содержащий расшифровку аббревиатур и понятий, используемых в документе.
2. «Общие сведения» — раздел, который описывает назначение и состав документа, а также сведения о производителе и технической поддержке.
3. «Описание оборудования» — раздел, который содержит основные сведения о поставляемом оборудовании, информацию о функциональных возможностях и технических характеристиках оборудования.
4. «Пользовательский интерфейс приложения» — раздел содержит общее описание интерфейса командной строки для настройки SBC.
5. «Настройка SBC» — раздел содержит детальное описание интерфейса командной строки для настройки подразделов SBC, кроме sbc.
6. «Раздел sbc» — раздел содержит детальное описание интерфейса командной строки для настройки подразделов sbc раздела SBC.
7. «Описание журналов» — раздел содержит описание формата CDR-файлов.

8. «Приложение» — раздел содержит схемы каждого этапа маршрутизации, пример настройки раздела SBC и примеры реализации типовых задач конфигурирования.

2.3 Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе установки и эксплуатации изделия, осуществляются производителем и службой технической поддержки.

2.3.1 Производитель

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «Телеком»

Тел.: (812) 449-47-27

Факс: (812) 449-47-29

WEB: http://www.protei.ru

E-mail: sales@protei.ru

2.3.2 Служба технической поддержки

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «Телеком»

Тел.: (812) 449-47-27 доб. 5999 (круглосуточно)

(812) 449-47-31

Факс: (812) 449-47-29

WEB: http://www.protei.ru

E-mail: mak.support@protei.ru, support.mak@protei.ru

3 Описание оборудования

ПРОТЕЙ I-SBC представляет собой программное оборудование операторского класса, широко используемое при построении сетей NGN. Кроме этого, это программное обеспечение сохраняет свою функциональность при работе исключительно в виртуальной среде.

SBC решает целый ряд задач, связанных с доступом и пакетной коммутацией, с управлением вызовами, чтобы снизить нагрузку с элементов внутри «домашней» сети оператора. Особое значение SBC имеет в сетях сервис-провайдеров для управления SIP-трафиком. В этом случае он осуществляет взаимодействие разнородного VoIP-оборудования, поддержку транскодирования и реализацию некоторых функций, которые обычно не решают брандмауэры и маршрутизаторы.

SBC построен на базе современных технологий, что позволяет использовать его с различными видами IP-устройств.

3.1 Функциональные возможности

Пограничный контроллер сессий выполняет следующие функции:

1. Трансляция сигнальных сообщений и медиаданных между интерфейсами.
2. Транскодирование медиаданных из одного кодека в другой: G.711 A/μ Law, G.722, G.722.2 (AMR WB), G.723.1, G.729 A.
3. Анализ качества медиаканалов, по которым осуществляется маршрутизация голосового трафика: задержка, джиттер, процент потерь пакетов и пр.
4. Автоматическая блокировка вызовов от неизвестных источников.
5. Обеспечение качества обслуживания (QoS).
6. Модификация атрибутов сигнальных сообщений.
7. Ограничение полосы пропускания для защиты от DoS атак.
8. Сокрытие сетевой топологии с помощью трансляции IP адресов и портов (NAPT).
9. Создание альтернативных маршрутов и балансировка нагрузки.

10. Лицензирование количества одновременных сессий, попыток соединений в секунду и сеансов с транскодированием.

11. Контроль статического и динамического доступа.

12. API для проверки маршрутизации, интерфейс для отображения/принудительного разрушения установленных соединений.

13. Контроль за установленными соединениями с помощью внутренних инструментов.

14. Формирование журналов трассировки.

15. Формирование RTCP статистики по направлениям.

16. Формирование CDR журналов с возможностью передачи по протоколам FTP/SFTP/SCP и мониторинг по протоколу SNMP.

17. Анализ вызовов, запись переговоров в *cap формате и формирование отчетов.

SBC поддерживает следующие протоколы:

1. Протоколы сигнализации VoIP — SIP.
2. Протокол пакетной передачи аудио-потока — RTP/RTCP.

Управление осуществляется при помощи приложения Command Line Interface, представляющего собой удобный, интуитивно понятный пользовательский интерфейс командной строки со строгой иерархической структурой данных.

3.2 Структура сети связи с участием SBC

Как уже было упомянуто выше, пограничный контроллер сессий устанавливается на границе сети оператора и обеспечивает взаимодействие VoIP-оборудования домашней сети с «внешним» миром.

На рисунке 1 приведена условная схема сети связи с участием оборудования «Session Board Controller» компании ООО «НТЦ ПРОТЕЙ».

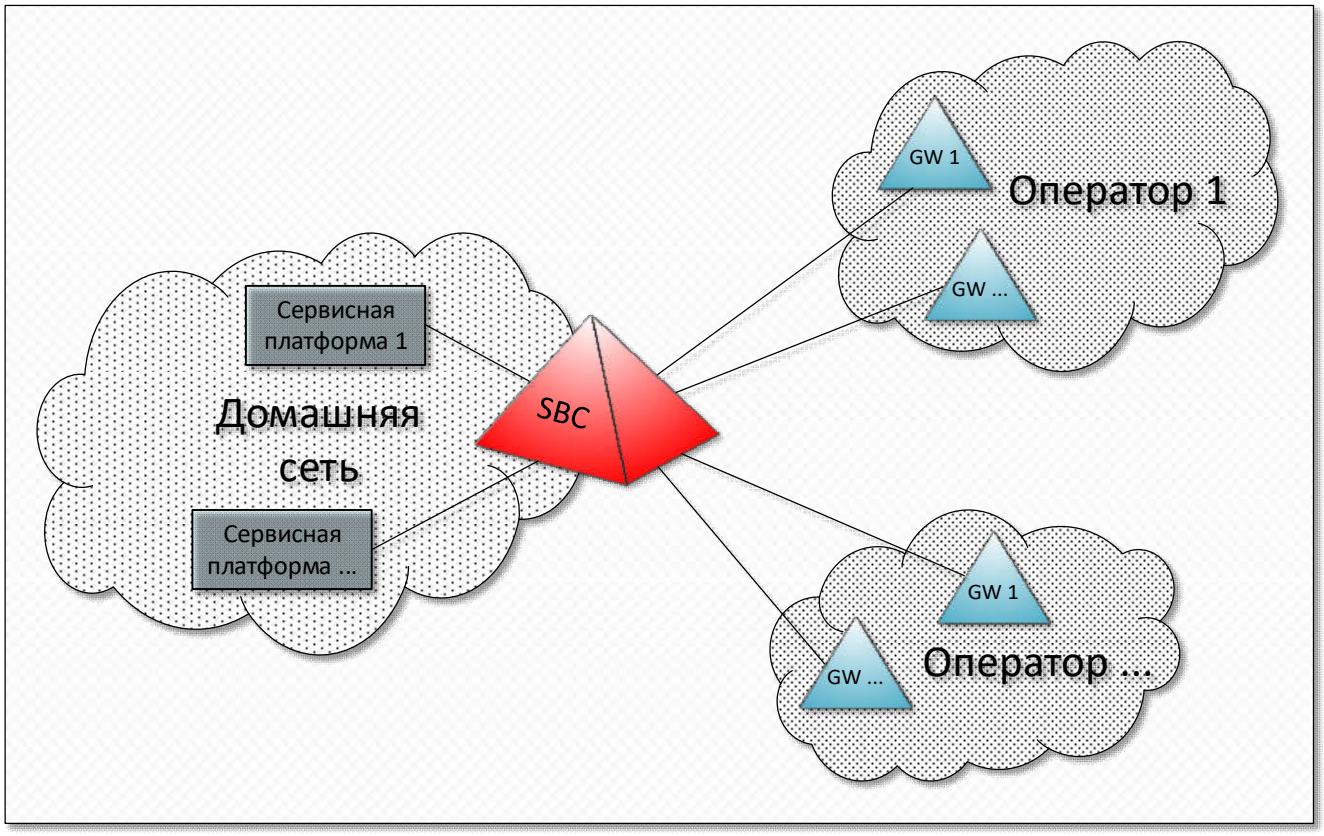

Рисунок 1 — Условная схема построения сети связи с участием SBC

В рамках продукта SBC можно выделить следующие сущности и понятия:

1. Сервисная платформа — оборудование внутренней сети оператора, взаимодействующее с сетями внешних операторов через пограничный контроллер сессий.

2. Оператор — сущность SBC, определяющая взаимодействие доверенной сети оператора (сервисной платформы) с сетью конкретного оператора.

3. Маршрут и медиашлюз — сущности, описывающие оборудование конкретного оператора, с которым осуществляется взаимодействие.

4. MCU — задает и контролирует взаимодействия с медиапотоками.

5. Медиапрофиль — набор списков разрешенных, запрещенных, обязательных и поддерживаемых кодеков.

Медиапрофиль применим как к сервисной платформе, так и к оператору или маршруту оператора. По умолчанию для всех операторов и сервисных платформ используется медиапрофиль default, в котором нет ограничений.

Примечание: данный профиль изменять не рекомендуется.

6. Шлюз — именованная пара ip:port, однозначно определяющая точки входа-выхода в «домашнюю» сеть оператора.

На рисунке 2 показано установление соединения между сервисной платформой и внешним оператором через I-SBC.

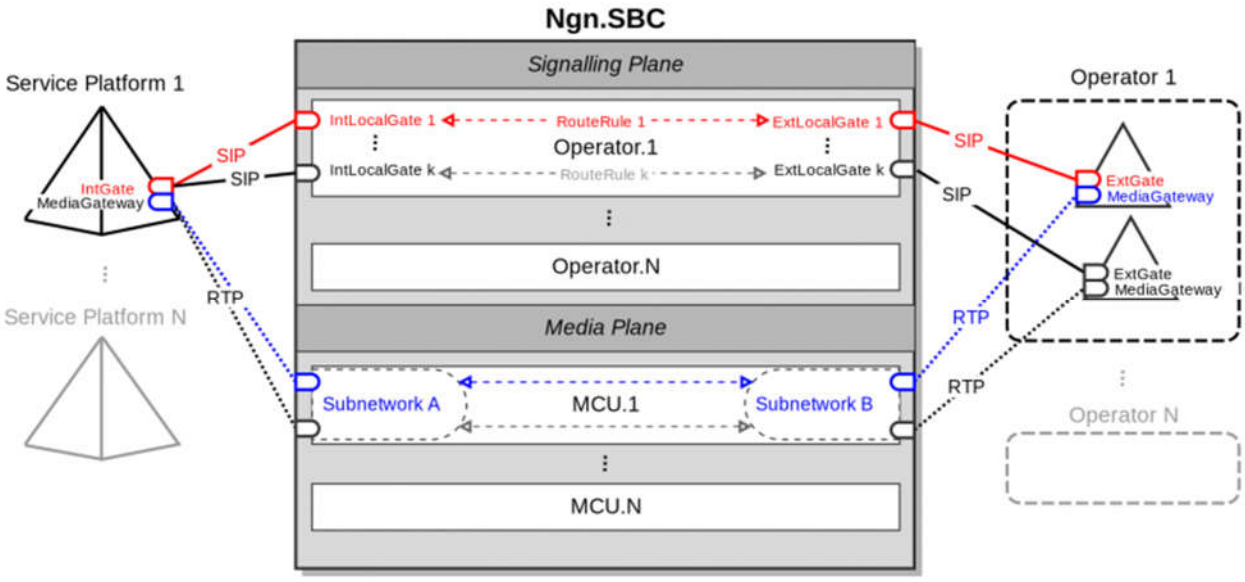

Рисунок 2 — Структурная схема SBC

Конкретный маршрут устанавливает однозначное соответствие между SIP-транком на шлюз оператора (ExtLocalGate ↔ ExtGate) и обслуживающей платформой.

Гарантируется, что все вызовы, пришедшие на IntLocalGate, будут отправлены во внешнюю сеть оператора с ExtLocalGate на ExtGate. Правило действует так же в обратную сторону: вызовы, пришедшие с ExtGate на ExtLocalGate, будут отправлены во внутреннюю сеть оператора на конкретную сервисную платформу.

Входящие и исходящие вызовы внутренней сети оператора имеют ограничения по количеству попыток вызовов в секунду и одновременному их количеству. Доступность шлюзов оператора для входящих вызовов определяется с помощью SIP-запросов OPTIONS. Маршрутизация вызовов определяется набором маршрутов, заданных в конфигурации I-SBC.

Медиашлюз конкретного оператора содержит список подсетей, конкретных адресов и портов медиашлюзов оператора, с которых разрешено принимать

медиатрафик. Если источник медиатрафика не указан в настройках медиашлюза как разрешенный, то сервисная платформа отбросит все такие RTP-пакеты.

Сервисная платформа описывает конфигурацию внутренней сети (или ее конкретного сегмента) и содержит набор шлюзов для сигнализации, определяемых приоритетами и весами.

MCU определяет настройки профилей для приема и отправки медиа от или к оператору, позволяет управлять медиаресурсами и распределять их, указывая количество одновременных аудио и видео-сессий.

4 Пользовательский интерфейс приложения cli

Приложение cli (Command Line Interface) предназначено для управления пограничным контроллером сессий и просмотра текущего состояния отдельных элементов.

Как следует из названия, cli имеет интерфейс командной строки и работает с данными иерархической структуры. Строка приглашения состоит из имени текущего узла иерархии и символа «>».

Далее в тексте будут использоваться следующие обозначения:

1. [someObject] — необязательный элемент команды, задание значения или указывание параметра по желанию или необходимости в некоторых случаях.
2. <Button> — клавиша, расположенная на клавиатуре, или кнопка на экранной клавиатуре.
3. «symbol» — выделение важного символа.
4. $exampleText$ — отображает поле, в котором пользователю необходимо указать свое значение, заполнитель.
5. # — отображает поле, в котором пользователю необходимо указать индекс объекта или число.

4.1 Вызов приложения cli

Внешний компьютер, выполняющий роль терминала, и оборудование доступа должны быть подключены к общей локальной сети.

Для запуска приложения cli необходимо выполнить следующие действия:

1. Авторизоваться в системе под учетной записью администратора.
2. В командной строке ввести имя приложения cli и нажать клавишу <Enter>.

На экране появится приглашение:

```
SBC>
```

Примечание: время старта программного обеспечения, установленного на оборудование доступа, занимает 3 5 минут после включения питания или перезапуска.

3. Если приглашение не появляется, то следует повторить попытку запуска. Для этого требуется нажать комбинацию клавиш <Ctrl>+<C> и повторно ввести команду cli.

Примечание: если после 5 минут приглашение снова не появляется, то следует перезапустить cli, выполнив команду _restart_cli в командной строке операционной системы (cmd.exe, bash и аналоги).

4.2 Работа с приложением cli

Для редактирования ввода используются элементы клавиатуры:

1. Буквенно-цифровые клавиши.
2. Клавиша <Delete> для удаления символа после курсора.
3. Клавиша <Backspace> для удаления символа перед курсором.

Примечание: если введена некорректная команда $badCmd$, то на экране появится сообщение:

```
Unexpected word: $badCmd$
```

Для аварийного выхода из приложения без сохранения конфигурации используется комбинацию клавиш <Ctrl>+<C>.

Примечание: аварийный выход используется в крайних случаях, когда приложение не посылает отклики на вводимые команды. Необоснованное применение аварийного выхода может привести к непредсказуемым последствиям.

В приложении реализован постраничный вывод информации. Если объем выводимой информации превышает размеры экрана, то вывод останавливается, в последней строке экрана отображается подсказка с клавишами, которые управляют выводом:

```
Press END/c, DOWN/ENTER or PAGE_DOWN/SPACE key for scroll
```

Клавиши, управляющие выводом:

1. <End> или <C> — переход в конец вывода.
2. <Стрелка вниз> или <Enter> — построчная прокрутка.
3. <Page Down> или <Пробел/Space> — пооконная прокрутка.

Для выхода из приложения применяется команда exit.

Пример выхода:

```
SBC> exit
Exiting
```

4.3 Управляющие клавиши

В таблице 2 приведен перечень управляющих клавиш и сочетания клавиш, которые используются при работе с приложением.

Таблица 2 — Управляющие клавиши

Клавиша	Значение
<Tab>	Вывод информации о текущем узле при пустом строке вывода: – список дочерних узлов; – список параметров; – список допустимых операций. Вывод возможных вариантов команды при вводе команды
<Home>	Перемещение курсора в начало командной строки
<End>	Перемещение курсора в конец командной строки
<Enter>	Ввод команды
<Backspace>	Удаление символа перед курсором
<Delete>	Удаление символа после курсора
<Ctrl>+<C>	Аварийный выход из приложения
<Стрелка влево>	Перемещение курсора влево
<Стрелка вправо>	Перемещение курсора вправо
<Стрелка вверх>	Перебор ранее введенных команд
<Стрелка вниз>	Перебор ранее введенных команд

4.4 Управляющие команды

В приложении cli реализованы следующие типы управляющих команд:

1. Навигация.
2. Создание и удаление объектов.
3. Настройка параметров.
4. Блокировка и разблокировка объектов.
5. Работа с массивами.
6. Вывод информации о конфигурации и состоянии объектов.
7. Применение ранее введенных данных, восстановление конфигурации.

Формат ввода команд в приложении cli: [action] object [key value] [param pValue [param pValue]…].

1. [action] — имя команды или директива.
2. object — имя объекта, которому изменяют значения параметров.
3. [key] — ключ, идентификатор.
4. [value] — значение ключа.
5. [param] — параметр, необходимый для выполнения [action].
6. [pValue] — значение параметра.
7. [subparam] — подпараметр.
8. [subpValue] — значение подпараметра.

В таблице 3 приведен перечень управляющих команд, которые используются при работе с приложением.

Таблица 3 — Управляющие команды

Команда	**Значение команды**
0-#	Выбор элемента массива с указанным индексом
block	Сервисная блокировка объекта
commit	Применение новых значений к параметрам конфигурации
default	Задание параметру значения по умолчанию
delete	Удаление объекта
end	Переход от текущего узла к родительскому, на уровень выше
insert	Добавление нового элемента в массив перед заданным элементом. Все последующие элементы сдвигаются вниз
moveto	Перемещение элемента на позицию перед указанным
remove	Удаление указанного элемента массива со сдвигом всех последующих вверх
resize	Изменение количества элементов в массиве: добавление новых элементов в конец или удаление последних элементов
rollback	Возвращение состояния на момент последнего ввода команды commit
show	Вывод на экран имен дочерних узлов и параметров текущего узла. **Примечание:** при выполнении команды не отображаются параметры со значениями по умолчанию
show-recursive	Вывод на экран имен всех вложенных узлов, параметры текущего узла и всех вложенных
show-state	Вывод состояний объекта

Команда	Значение команды
show-config	Вывод на экран конфигурации текущего узла и всех вложенных узлов в структурированном виде. Используется для переноса конфигурации. **Примечание:** команда возвращает список команд, необходимых для создания текущей конфигурации
unblock	Сервисная разблокировка объекта

4.4.1 Навигация

В приложении cli параметры конфигурации объединены в иерархию, которую можно представить в виде дерева. В любой момент времени пользователь находится в каком-нибудь конкретном узле дерева.

Все вводимые команды применяются к текущему узлу.

Формат команды для перемещения по дереву: object [key value].

Несколько таких команд можно объединить в одну строку, отделив команды пробелом. Переход от текущего узла к дочернему можно выполнять двумя способами:

1. Вводить последовательно по одной команде:

– object;

– object key value.

Пример перемещения по одной команде:

```
SBC> sbc
sbc> service-platform id 0
sbc service-platform id 0>
```

2. Вводить команды через пробел: object object key value.

Пример перемещения по командам через пробел:

```
SBC> sbc service-platform id 0
sbc service-platform id 0>
```

Переход от текущего узла на уровень выше осуществляется командой end.

Переход к корневому узлу из вложенных узлов происходит последовательным выполнением команды end. При выполнении команды в корневом файле появляется оповещение:

```
SBC> end
```

```
Current object has no parent.
SBC>
```

Вывод текущей конфигурации осуществляется командой show. На экран выводится список настроенных параметров и дочерних узлов.

Формат команды навигации для входа в раздел: object key value.

Примечание: если введена команда перехода в несуществующий узел или объект, то эта команда создаст новый узел или объект.

4.4.2 Создание и удаление объектов

Создание объекта требует задания обязательных параметров в одной строке или отдельной командой из текущего узла.

Команды создания новых объектов, например, физических или логических ресурсов. могут иметь два формата. Это зависит от того, индексируется или нет создаваемый объект в пределах текущего раздела.

Если создаваемый объект object индексируется по ключу key со значением value, то формат команды создания будет следующим: object key value.

Если создаваемый объект object не индексируется, то такой объект может существовать только в единственном экземпляре в данном разделе. Команда создания сводится к вводу имени этого объекта в командной строке.

Объект создается автоматически, если произошел переход к несуществующему объекту.

Формат команды создания: object [key value].

Пример создания:

```
SBC> sbc
sbc> mcu id 1
sbc mcu id 1> commit
transaction result: success
```

Примечание: указание идентификаторов объекта key value опционально. Он не указывается для единичных объектов. Если указать key value для такого объекта, то система выдаст сообщение об ошибке.

Форматы команды удаления: delete object, delete object key value.

Формат команды зависит от того, индексируется ли удаляемый объект. Вложенные в удаляемый узел объекты удаляются автоматически.

Примечание: команда удаления доступна не для всех объектов.

Пример удаления:

```
SBC> sbc
sbc> delete mcu id 1
sbc> commit
```

Удаление объекта в некоторых случаях предполагает его обязательную предварительную блокировку. Блокировка объекта осуществляется командой block. См. п. 4.4.4 «Операции над объектами».

Пример блокировки:

```
SBC> sbc operator id 100
sbc operator id 100> block
sbc operator id 100> end
sbc> delete operator id 100
sbc> commit
```

4.4.3 Настройка параметров

Настройка параметров объекта осуществляется из текущего узла и может включать в себя несколько действий: задание, просмотр и изменение параметра.

Параметр можно задать одновременно с созданием объекта или же отдельной командой после создания.

Формат команды для задания параметра: param value.

Пример задания значения:

```
SBC> sbc
sbc> service-platform id 1
sbc service-platform id 1> polling-timeout 60000
sbc service-platform id 1> show
  mcu-network          'SN-SP'
  media-profile             '1'
  polling-timeout            6000
  gates [size=2]
    gates 0
```

```
    internal-host          '192.168.125.10:5060'
    priority        1
    weight          100
  gates 1
    internal-host          '192.168.125.11:5061'
    priority        0
    weight          50

sbc service-platform id 1> commit
```

В приведенном примере параметр polling timeout задан отдельной командой. Для просмотра заданных параметров служит команда show.

Просмотр полного списка доступных для настройки параметров осуществляется нажатием клавиши <Tab> в пустой строке.

У некоторых объектов есть обязательные параметры конфигурации. Таким параметрам обязательно требуется задать значения, некоторые обязательные параметры имеют значения по умолчанию.

В списке обязательные параметры помечены символом «*» в строке комментария.

Пример отображения обязательных параметров:

```
SBC> sbc service-platform id 13
sbc service-platform id 13>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  description       Set description
  mcu-ip            * Set MCU ip-address
  mcu-port          * Set MCU port
  mcu-subnet        * Configure MCU interfaces
  default           set parameter to default value
  end               return to parent
  block             Block the object
  show-state        Show current state of the object
  unblock           Unblock the object
sbc service-platform id 13>
```

Набор параметров объекта может меняться в зависимости от установленных значений других параметров.

Формат команды для изменения значения: object key value param pValue.

Сохранение изменений осуществляется командой commit.

Задание параметру param значения по умолчанию приводит к удалению этого параметра из конфигурации. При этом значение определяется логикой работы программного обеспечения.

Формат команды для задания значения по умолчанию: default param.

Примечание: не все параметры можно удалять из конфигурации.

Определены два вида параметров:

1. Простые параметры. Формат команды настройки: param pValue.

Пример настройки простого параметра:

```
SBC> sip
sip> local-port 8080
sbc> commit
```

2. Сложные параметры. Формат команды настройки: complex_param param1 pValue1 [param2 pValue2].

Пример настройки сложного параметра:

```
SBC> sip
sip> timer t1 500 t2 4000 t4 5000 end
sip> commit
transaction result: success
```

Для редактирования сложного параметра необходимо в командной строке ввести имя параметра. Вложенные параметры редактируются как простые, формат: subparam subpValue.

Пример настройки вложенного параметра:

```
SBC> sip
sip> timer
sip timer> t1 500
sip timer> t2 4000
sip timer> t4 5000
sip timer> end
sip> commit
transaction result: success
```

4.4.4 Операции над объектами

В некоторых разделах доступны команды, позволяющие блокировать и снимать блокировку с физических и логических ресурсов.

Объект блокируется командой block.

Пример блокировки:

```
SBC> sbc operator id 1
sbc operator id 1> block
```

Блокировка объекта снимается командой unblock.

Пример снятия блокировки:

```
SBC> sbc operator id 1
sbc operator id 1> unblock
```

4.4.5 Операции с массивами

В контексте приложения cli массив рассматривается как некоторая совокупность однотипных элементов. Положение элемента в массиве определяется его индексом.

Созданный массив является пустым. Массив можно создать двумя способами: изменяя количество элементов или вставив новые элементы.

Формат вывода массива: array [size = s].

1. Array — имя массива.
2. S — длина массива, т.е. текущее количество элементов.

С массивами возможно выполнить следующие операции:

1. Изменить длину.
2. Удалить элемент.
3. Переместить элемент.
4. Вставить новый элемент.
5. Изменить элемент.

Чтобы вывести список допустимых действий, необходимо нажать на клавишу <Tab>.

Чтобы просмотреть все элементы массива, необходимо вызвать команду show.

Пример пустого массива:

```
SBC> sip
sip> show
...
  gate [size=0]
...
sip>
```

В приведенном примере массиву задано имя gate, а количество элементов в массиве равно 0, size=0.

Чтобы выполнить операции над массивом, необходимо войти в секцию редактирования командой $arrayName$, имя массива.

Пример секции редактирования:

```
sip> gate
sip gate>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  resize            resize vector and if need, append it by default values.
  remove            remove element from vector by index.
  moveto            move element to another position
  <0-13>            select element by index for editing
  insert            insert element before 'idx' and select it for editing.
  end               return to parent
sip gate>
```

В таблице 4 приведены команды взаимодействия с элементами массива.

Таблица 4 — Команды для элементов массива

Команда	**Значение и формат команды**
<0-#lastIndex#>	Выбор элемента с указанным индексом для редактирования. Формат команды: #index
resize	Изменение длины массива: добавление новых элементов в конец или удаление последних элементов. Формат команды: resize #size #size — необходимое количество элементов
remove	Удаление указанного элемента со сдвигом последующих вверх. Формат команды: remove #index

Команда	Значение и формат команды
	#index# — индекс удаляемого элемента
moveto	Перемещение элемента на новую позицию. Формат команды: moveto #indexNow #indexNext #indexNow — текущий индекс элемента. #indexNext — новый индекс элемента
insert	Добавление нового элемента на позицию со сдвигом последующих элементов вниз. Формат команды: insert #index #index — индекс нового элемента

Создание массива, изменяя количество элементов в массиве.

Команда resize изменяет количество элементов, добавляя пустые элементы в конец массива и удаляя элементы, начиная с конца.

Пример создания массива:

```
sip> gate
sip gate> resize 2
sip gate> show
[size=2]
  0
  1

sip gate>
```

Создание массива, вставляя новые элементы.

Команда insert добавляет пустой элемент в массив и автоматически переходит в раздел редактирования нового элемента.

Пример создания массива:

```
sip> gate
sip gate> insert 0
sip gate 0> ip 1.1.1.1 name IntGate port 5061
sip gate 0> end
sip gate> show
[size=1]
  0
    name              'IntGate'
    ip          '1.1.1.1'
```

```
  port            5061

sip gate>
```

Редактирование элементов значений.

Настройка элементов массива осуществляется для каждого элемента отдельно. Чтобы изменить значение элемента, необходимо ввести индекс элемента и новое значение.

Формат команды для изменения элемента: index value.

Пример задания значений:

```
sip gate> 0
sip gate 0> ip 1.1.1.1 name All5060 port 5060
sip gate 0> end
sip gate> 1
sip gate 1> ip 1.1.1.2 name Int5080 port 5080
sip gate 1> end
sip gate> show
[size=2]
 0
  name            'All5060'
  ip        '1.1.1.1'
  port            5060
 1
  name            'Int5080'
  ip        '1.1.1.2'
  port            5080

sip gate>
```

4.4.6 Отображение конфигурации и состояния объектов

Команды отображения конфигурации показывают текущую настройки устройства с учетом внесенных изменений. Действующая конфигурация может отличаться от отображаемой; например, если изменения не были применены командой commit.

Команда show отображает все заданные параметры в данном узле и все дочерние узлы. Для некоторых узлов отображаются все параметры и текущего, и дочерних узлов.

Команда show recursive отображает конфигурацию текущего узла, а затем применяется для всех вложенных узлов, т.е. выполняется рекурсивно. Вывод отформатирован отступами в соответствии со вложенностью узлов.

Значения параметров отображаются в формате: param value.

В одной строке могут отображаться названия нескольких вложенных друг в друга узлов, имя параметра и его значение в формате: object1 object2…param value.

1. Команда show config возвращает список команд cli, необходимых для воссоздания текущей конфигурации. Используется для создания сходных конфигураций.

2. Команда show state отображает текущее состояние ресурса, связанного с этим узлом. Состояние отображается в виде списка переменных со значениями в формате: param = value.

4.4.7 Применение и восстановление конфигурации

Команда commit завершает ряд действий:

1. Сохранение изменений в конфигурации.
2. Создание объекта.
3. Изменение значений параметров.
4. Удаление объекта.

Пример успешного выполнения:

```
transaction result: success
```

Пример неуспешного выполнение:

```
transaction result: fail
```

Если какой-либо обязательный параметр не инициализирован, то система отправит оповещение, что применить текущие невозможно. В появившемся сообщении будет указано имя раздела, в котором отсутствует обязательный параметр.

Пример системного оповещения:

```
SBC> sbc operator id 404
sbc operator id 404> commit
can't commit data
```

not all mandatory fields set in object: sbc /operator id 404/
sbc operator id 404>

Примечание: для облегчения поиска возможных ошибок рекомендуется выполнять команду commit после каждого изменения.

5 Настройка SBC

Для доступа к настройкам конфигурации необходимо войти в систему и запустить приложение (см. п. 4.1 «Вызов приложения cli»).

Справочная информация о текущем узле иерархии отображается при нажатии на клавишу <Tab> в пустой командной строке.

Пример использования <Tab>:

```
SBC>
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  sbc              Configure Session Border Controller
  sip              Configure SIP params
```

Система SBC содержит разделы sip и sbc, в которых настраиваются параметры SIP и основного ядра sbc.

В таблице 5 приведены все типы данных, которые используются при работе с приложением, вместе с описанием.

Таблица 5 — Используемые типы данных

Команда	Значение команды
array	Массив: набор однотипных объектов, или кортеж: ограниченный набор разнородных объектов
bool	Логическая величина. Принимает значения 0 или 1, False или True, соответственно
enum	Выбор варианта из списка предложенных значений
string	Строковая величина. Для ввода значений могут использоваться латинские буквы любого регистра, цифры 0-9 и специальные символы
regex	Регулярное выражение, маска
uint	Числовая величина. Для ввода значений могут использоваться цифры 0-9, а также латинские буквы A-F и x для чисел с основанием 16. Принимает целые положительные значения или 0
ip	Строка типа string, имеет формат IPv4: xxx.xxx.xxx.xxx Используется вместе с портом как тип ip:port. Используется вместе с маской подсети как тип ip/mask

5.1 Раздел sip

Для настройки доступны параметры SIP и таймеры SIP-сигнализации.

Чтобы войти в настройки, в корневом разделе необходимо ввести имя раздела sip.

Пример входа и просмотра раздела sip:

```
SBC> sip
sip>
  show                    show current object
  show recursive          recursive show current object
  show config             show CLI command list for object
  allow-update            Allow UPDATE method
  description             Set description
  dns-srv                 Enable DNS SRV
  gate                    Additional local SIP endpoints
  ip-tos                  Set IP ToS byte value for SIP signalling
  listen-ip               Set local IP-address for SIP signalling. SIP port will be listened
on this address
  local-ip                * Set local IP-address for SIP signalling. It will be used in Via and
Contact headers
  local-port              * Set local UDP-port for SIP signalling. It will be used in Via and
Contact headers
  no-response-reject-cause    Setup reject cause for unreachable SIP trunks
  timer                   Set SIP timers
  default                 set parameter to default value
  end                     return to parent
sip>
```

В таблице 6 приведен перечень управляющих команд.

Таблица 6 — Управляющие команды в разделе sip и его секциях

Команда	Значение команды
show	Вывод на экран имен дочерних узлов и параметров текущего узла. **Примечание:** при выполнении команды не отображаются параметры со значениями по умолчанию
show-config	Вывод на экран конфигурации текущего и всех вложенных узлов в структурированном виде. Возвращает список команд, необходимых для создания текущей конфигурации
show-recursive	Вывод на экран имен всех вложенных узлов, параметры текущего узла и всех вложенных
default	Задание параметру значения по умолчанию
end	Переход от текущего узла к родительскому, на уровень выше

5.2 Основные параметры SIP

В таблице 7 приведен перечень конфигурируемых параметров в разделе sip.

Таблица 7 — Параметры sip

Параметр	Значение параметра
allow-update	Флаг поддержки метода UPDATE. Метод добавляется в заголовок SIP-сообщений Allow. Тип переменной — bool. Значение по умолчанию — 1
description	Текстовое описание. Тип переменной — string
dns-srv	Флаг использования поддержки DNS SRV-записей. Тип переменной — bool
gate	Настройка дополнительных точек доступа и виртуальных шлюзов для SIP-сигнализации. См. п. 5.3 «**Ошибка! Источник ссылки не найден.**». Тип переменной — array
ip-tos	Значение байта ToS в заголовке сообщения SIP. Тип переменной — uint, допустимы шестнадцатеричные числа 0x. Диапазон значений: 0-255 или 0x00-0xFF. Значение по умолчанию — 0
listen-ip	IP-адрес, на котором прослушивается порт UDP. Тип переменной — ip. **Примечание:** если значение не задано, то подходящим будет считаться любой адрес
local-ip (*)	IP-адрес, подставляемый в заголовки Via и Contact сообщений SIP. Обычно соответствует IP-адресу устройства. Обязательный параметр. Тип переменной — ip. **Примечание:** после изменения необходимо перезапустить приложение
local-port (*)	UDP-порт для приема и отправки SIP-сообщений. Обязательный параметр. Тип переменной — uint. Диапазон значений: 1024-65535. Стандартный порт SIP — 5060. **Примечание:** после изменения необходимо перезапустить приложение
no-response-reject-cause	Причина отбоя в случае недоступности SIP-направления.

Параметр	Значение параметра
	Тип переменной — uint. Диапазон значений: 0-255
timer	Настройки таймеров SIP. См. п. 5.4 «**Ошибка! Источник ссылки не найден.**». Тип переменной — array

5.3 Виртуальные шлюзы

Для приема и отправки пакетов сигнализации SIP в пограничном контроллере сессий применяются виртуальные шлюзы (гейты). Шлюзы создаются и для внешней, так и для внутренней сети, в дальнейшем используются при настройке маршрутизации.

Базовым свойством при создании виртуального шлюза является gate.

Параметр gate является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Пример действий с параметром gate:

```
sip gate>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  resize            resize vector and if need, append it by default values.
  remove            remove element from vector by index.
  moveto            move element to another position.
  <0-2>             select element by index for editing.
  insert            insert element before 'idx' and select it for editing.
  end               return to parent
```

Чтобы войти в секцию настройки шлюза, необходимо в разделе sip ввести имя массива gate и выбрать существующий шлюз или создать новый:

```
SBC> sip
sip> gate
sip gate> show
[size=2]
  0
    name              'IntGate'
    ip          '192.168.70.201'
    port              5100
    ip-tos            '0xeF'
  1
```

```
  name              'ExtGate'
  ip          '10.0.0.10'
  port              5200
sip gate> 0
sip gate 0>
 ip        * Set additional local IP-address for SIP signalling.
 ip-tos    Set IP TOS byte value for SIP signalling
 name       * Set GATE name
 port      * Set additional local UDP-port for SIP signalling.
```

В таблице 8 приведен перечень параметров виртуальных шлюзов.

Таблица 8 — Параметры виртуальных шлюзов

Параметр	Значение параметра
allow-update	Флаг поддержки метода UPDATE. Метод добавляется в заголовок SIP-сообщений Allow. Тип переменной — bool. Значение по умолчанию — 1
description	Текстовое описание. Тип переменной — string
dns-srv	Флаг использования поддержки DNS SRV-записей. Тип переменной — bool
gate	Настройка дополнительных точек доступа и виртуальных шлюзов для SIP-сигнализации. См. п. 5.3 «**Ошибка! Источник ссылки не найден.**». Тип переменной — array

5.4 Таймеры SIP-сигнализации

В разделе timer содержатся значения таймеров SIP-сигнализации. Настройка таймеров SIP не является обязательной.

Чтобы вывести на экран список таймеров, необходимо нажать на клавишу <Tab> в пустой командной строке, находясь в разделе sip timer.

```
SBC> sip
sip> timer
 a        INVITE request retransmit interval, for UDP only
 b        INVITE transaction timeout timer
 c        Proxy INVITE transaction timeout
 d        Wait time for response retransmits
 e        non-INVITE request retransmit interval, UDP only
 f        non-INVITE transaction timeout timer
```

g INVITE response retransmit interval
h Wait time for ACK receipt
i Wait time for ACK retransmits
j Wait time for non-INVITE request
k Wait time for response retransmits
t0 Initial INVITE timeout timer (non-standart)
t1 RTT Estimate
t2 The maximum retransmit interval for non-INVITE requests and INVITE responses
t4 Maximum duration a message will remain in the network

В таблице 9 приведено описание таймеров SIP-сигнализации.

Все параметры имеют тип uint, значения задаются в миллисекундах, мс.

Для всех параметров предусмотрены значения по умолчанию.

Таблица 9 — Таймеры SIP-сигнализации

Таймер	**Описание**
t0	Максимальное время ожидания INVITE при исходящем вызове. Значение по умолчанию — 10000 мс
t1	RTT (время двойного оборота по сети). Значение по умолчанию — 500 мс
t2	Максимальный интервал между повторными не INVITE-запросами и ответами на INVITE. Значение по умолчанию — 4000 мс
t4	Максимальное время, в течение которого сообщение может оставаться в сети. Значение по умолчанию — 5000 мс
Таймер A	Время передачи повторного запроса INVITE. Только при использовании UDP. Значение по умолчанию — 500 мс
Таймер B	Время ожидания окончательного ответа на INVITE-сообщение. Значение по умолчанию — 32000 мс
Таймер C	Время ожидания INVITE-сообщения при использовании прокси Значение по умолчанию — 180000 мс
Таймер D	Время ожидания повторных ответов. Значение по умолчанию для UDP — 32000 мс. Значение по умолчанию для TCP/SCTP — 0
Таймер E	Время передачи повторного не INVITE-запроса. Только при использовании UDP. Значение по умолчанию — 500 мс

Таймер	Описание
Таймер F	Время ожидания окончательного ответа на не INVITE-сообщение. Значение по умолчанию — 32000 мс
Таймер G	Время передачи повторного ответа на запрос INVITE. Значение по умолчанию — 500 мс
Таймер H	Время ожидания ACK-сообщения. Значение по умолчанию — 32000 мс
Таймер I	Время ожидания повторных ACK-сообщений. Значение по умолчанию для UDP — 5000 мс. Значение по умолчанию для TCP/SCTP — 0
Таймер J	Время ожидания повторных не INVITE-запросов. Значение по умолчанию для UDP — 32000 мс. Значение по умолчанию для TCP/SCTP — 0
Таймер K	Время ожидания повторных ответов. Значение по умолчанию для UDP — 5000 мс. Значение по умолчанию для TCP/SCTP — 0

Пример инициализации таймера J:

```
SBC> sip
sip> timer
sip timer> J 1000
```

6 Раздел sbc

Для настройки доступны все ключевые параметры MCU, медиапрофилей, операторов, сервисных платформ, маршрутизации и кодеков.

Для входа в раздел настройки требуется ввести имя раздела sbc:

```
sbc>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  audio-codec       Configure audio codec params
  common            Configure common params
  mcu               Configure Multipoint Control Unit params
  media-profile     Configure media profile params
  operator          Configure operator params
  routing           Configure routing
  service-platform  Configure service platform params
  video-codec       Configure video codec params
  end               return to parent
  delete            delete object
```

В таблице 10 приведен перечень управляющих команд.

Таблица 10 — Управляющие команды в разделе sbc и его секций

Команда	Значение команды
block	Сервисная блокировка объекта
commit	Применение новых значений к параметрам конфигурации
default	Задание параметру значения по умолчанию
delete	Удаление объекта
end	Переход от текущего узла к родительскому, на уровень выше
rollback	Возвращение состояния на момент последнего ввода команды commit
show	Вывод на экран имен дочерних узлов и параметров текущего узла. **Примечание:** не отображает параметры со значениями по умолчанию
show-config	Вывод на экран конфигурации текущего и всех вложенных узлов в структурированном виде. Возвращает список команд, необходимых для создания текущей конфигурации
show-recursive	Вывод на экран имен всех вложенных узлов, параметры текущего узла и всех вложенных
show-state	Вывод состояний объекта
unblock	Сервисная разблокировка объекта

6.1 Основные параметры sbc

В таблице 11 приведен перечень параметров sbc.

Таблица 11 — Параметры sbc

Команда	Значение команды
audio-codec	Настройка аудиокодеков. См. п. 6.8 «**Ошибка! Источник ссылки не найден.**»
common	Настройка общих параметров. См. п. 6.11 «**Ошибка! Источник ссылки не найден.**»
mcu	Настройка MCU. См. п. 6.5 «**Ошибка! Источник ссылки не найден.**»
media-profile	Настройка медиапрофилей. См. п. 6.7 «**Ошибка! Источник ссылки не найден.**»
operator	Настройка оператора. См. п. 6.2 «**Ошибка! Источник ссылки не найден.**»
routing	Настройка правил внутренней маршрутизации. См. п. 6.10 «**Ошибка! Источник ссылки не найден.**»
service-platform	Настройка сервисной платформы. См. п. 6.3 «**Ошибка! Источник ссылки не найден.**»
video-codec	Настройка видеокодеков. См. п. 6.9 «**Ошибка! Источник ссылки не найден.**»

6.2 Оператор

Основные настройки, выполняемые в разделе:

1. Ограничение количества вызовов в секунду cps и режим обработки вызовов, если ограничение превышено.

2. Настройка максимального количества одновременных, входящих и исходящих вызовов.

3. Задание номера сервисной платформы, подсети MCU и идентификатора медиапрофиля для вызовов из внешней сети.

Чтобы войти в подраздел настройки оператора, необходимо в узле sbc ввести operator id #.

Пример вывода информации об операторе:

```
sbc> operator id 0
sbc operator id 0>
  show                  show current object
  show recursive        recursive show current object
  show config           show CLI command list for object
  route                 Configure routes for operator
  cps                   Set calls peer seconds
  cps-exceed-mode       Select behavior when exceeding the CPS
```

```
  description              Set description
  hide-display-name        Hide display name in SIP headers
  max-call-duration        Set maximum call duration
  max-calls                Set max calls count
  max-incoming-calls       Set max incoming calls count
  max-outgoing-calls       Set max outgoing calls count
  mcu-network              * Set MCU network ID for external media
  media-gateway            Configure media gateways for operator
  media-profile            Set media profile ID for external network
  no-ping-cause            Setup release cause for unreachable SIP trunks
  no-response-cause        Setup release cause on no response calls
  p-asserted-identity forward Forward P-Asserted-Identity (PAI) header
  prack                    Setup functioning of operator with PRACK
  public-dst-hostname      Set public destination hostname
  public-dst-hostname-mask Set allowed public hostname for destination SIP-URI
  sip-i-enabled            Set SIP-I/SIP-T mode
  sp-number                * Set service platform number
  transport                Select transport type for outgoing calls
  user-phone-to-sip-uri    Add 'user=phone' parameter to SIP URI
  default                  set parameter to default value
  end                      return to parent
  delete                   delete object
  block                    Block the object
  show-state               Show current state of the object
  unblock                  Unblock the object
sbc operator id 0>
```

В таблице 12 приведен перечень основных параметров оператора.

Таблица 12 — Параметры оператора

Параметр	Значение параметра
route	Правила маршрутизации для оператора (см. п. 6.2.1 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — array
cps	Максимальное количество вызовов в секунду. Тип переменной — uint или “–1”. Значение по умолчанию — “–1”, нет ограничений
cps-exceed-mode	Режим обработки вызовов при превышении значения cps. Тип переменной — enum. Возможные значения: – DROP — игнорировать новые запросы INVITE; – RELEASE — отправлять ошибку 4xx/5xx.

Параметр	Значение параметра
	Значение по умолчанию — DROP
description	Текстовое описание оператора. Тип переменной — string. Значение по умолчанию — «», пустая строка
hide-display-name	Флаг сокрытия display–name в заголовках SIP. Тип переменной — bool. Значение по умолчанию — 1
max-call-duration	Максимальная длительность разговора. Тип переменной — uint или “–1”, измеряется в секундах. Значение по умолчанию — “–1”, нет ограничений
max-calls	Максимальное количество одновременных вызовов. Тип переменной — uint или “–1”. Значение по умолчанию — “–1”, нет ограничений
max-incoming-calls	Максимальное количество одновременных вызовов от оператора. Тип переменной — uint или “–1”. Значение по умолчанию — “–1”, нет ограничений
max-outgoing-calls	Максимальное количество одновременных вызовов к оператору. Тип переменной — uint или “–1”. Значение по умолчанию — “–1”, нет ограничений
mcu-network (*)	Имя подсети MCU, на которую ожидается прием медиа с внешней сети (см. п. 6.6 «**Ошибка! Источник ссылки не найден.**»). Обязательный параметр. Тип переменной — string
media-gateway	Настройки медиашлюзов оператора (см. п. 6.2.2 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — array
media-profile	Имя медиапрофиля для внешней сети (см. п. 6.7 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — string
no-ping-cause	Отображаемая причина отбоя в случае потери связи. Тип переменной — uint. **Примечание:** значение должно быть не меньше 400. Значение по умолчанию — 500
no-response-cause	Отображаемая причина отбоя в случае отсутствия ответа. Тип переменной — uint. **Примечание:** значение должно быть не меньше 400. Значение по умолчанию — 500

Параметр	Значение параметра
p-asserted-identity-forward	Флаг использования прокси для PAI. Тип переменной — bool. Значение по умолчанию — 0
public-dst-hostname	Строка замены для заголовка Host в строке запроса, если текущее значение не подходит для public-dst-hostname-mask. Тип переменной — string. Значение по умолчанию — «», пустая строка. **Примечание:** если параметру не задано значение, то произойдет замена на Host назначения
public-dst-hostname-mask	Выражение для проверки заголовка Host в строке запроса. Тип переменной — regex. Значение по умолчанию — «», пустая строка. **Примечание:** если значение не задано, то никакой Host не удовлетворяет регулярному выражению
sip-i-enabled	Режим работы ISUP. Тип переменной — enum. Возможные значения: – Forbidden — запрещено использование ISUP; – Optional — разрешено использование ISUP; – Mandatory — обязательно использование ISUP. Значение по умолчанию — Optional
sp-number (*)	Номер сервисной платформы. Обязательный параметр. Тип переменной — uint
transport	Тип транспортного протокола для исходящих вызовов. Тип переменной — enum. Возможные значения: – UDP; – TCP. Значение по умолчанию — Undefined. **Примечание:** если параметру не задано значение, то будет использоваться тот же протокол, что и во входящем запросе
user-phone-to-sip-uri	Флаг использования заголовка user = phone в SIP-запросах. Тип переменной — bool. Значение по умолчанию — 0

Пример настроек оператора:

```
SBC> sbc operator id 0
```

```
sbc operator id 0> show
  description          'SOA'
  cps          '40'
  max-calls            '400'
  max-incoming-calls           '-1'
  max-outgoing-calls           '-1'
  sp-number            0
  mcu-network          'ExtMCU1'
  media-profile                '0'
  media-gateway [size=1]
    media-gateway 0
      host             '1.1.1.1'
      port-range [size=1]
        port-range 0
          port-start           33000
          port-count           1000
  route rule 0
sbc operator id 0> route rule 0
sbc operator id 0 route id 0> show
  internal-gate                'IntGate1'
  external-host                '1.1.1.1:5060'
  external-gate                'ExtGate1'
  mcu-network          'ExtMCU1'
```

6.2.1 Правила маршрутизации для оператора

Конкретный маршрут устанавливает однозначное соответствие между SIP-транком на шлюз оператора и SIP-транком на обслуживающую платформу.

Чтобы войти в секцию настройки маршрута для оператора, необходимо в разделе sbc operator id # ввести route rule # и выбрать существующий маршрут или создать новый:

```
sbc operator id 0> route rule 0
sbc operator id 0 route id 0>
  show                 show current object
  show recursive         recursive show current object
  show config          show CLI command list for object
  RuriRouting          Enable/Disable Request URI Routing
  description          Set description
  external-gate         * Set external network GateID
  external-host         * Set external network address
  internal-gate         * Set internal network GateID
```

```
max-call-duration      Set maximum call duration
mcu-network            Set MCU network ID for external media
media-profile          Set media profile ID for external network
ping-timeout           Set sip-ping interval
polling-timeout        Set polling timeout
public-dst-hostname    Set public destination hostname
sip-i-enabled          Set SIP-I/SIP-T mode
sp-number              Set service platform number
strong-link            Set strong link
transport              Select transport type for outgoing calls
user-phone-to-sip-uri  Add ‘user=phone’ parameter to SIP URI
default                set parameter to default value
end                    return to parent
block                  Block the object
show-state             Show current state of the object
unblock                Unblock the object
```

В таблице 13 приведены параметры маршрута оператора и их описание.

Таблица 13 — Параметры маршрута оператора

Параметр	**Значение параметра**
RuriRouting	Флаг маршрутизации вызова на Host, указанный в запросе. Тип переменной — bool. Значение по умолчанию — 0
description	Текстовое описание маршрута. Тип переменной — string. Значение по умолчанию — «», пустая строка
external-host (*)	Адрес шлюза, принимающего и отправляющего вызовы, во внешней сети оператора. Обязательный параметр. Тип переменной — ip:port
external-gate (*)	Адрес шлюза, с которого отправляются вызовы во внешнюю сеть. Обязательный параметр. Тип переменной — string. **Примечание:** должен быть указан в SIP.cfg
internal-gate (*)	Адрес шлюза, на который отправляются вызовы из внутренней сети. Обязательный параметр. Тип переменной — string. **Примечание:** должен быть указан в SIP.cfg

Параметр	Значение параметра
max-call-duration	Максимальная длительность разговора. Тип переменной — uint или “–1”, измеряется в секундах. Значение по умолчанию — “–1”, нет ограничений
mcu-network	Идентификатор подсети MCU, на которую ожидается прием медиа с внешней сети. Тип переменной — string. Значение по умолчанию — значение для оператора
media-profile	Идентификатор медиапрофиля для внешней сети. Тип переменной — string. Значение по умолчанию — значение для оператора
ping-timeout	Период пингования. Тип переменной — uint, измеряется в миллисекундах, мс. Значение по умолчанию — 0
polling-timeout	Период поллинга вызовов. Тип переменной — uint, измеряется в миллисекундах, мс. Диапазон значений: 0-3600000 мс. Значение по умолчанию — значение для раздела sbc
public-dst-hostname	Строка замены для заголовка Host в строке запроса, если текущее значение не подходит для public-dst-hostname-mask. Тип переменной — string. Значение по умолчанию — «», пустая строка. **Примечание:** если параметру не задано значение, то произойдет замена на Host назначения
sip-i-enabled	Режим работы ISUP. Тип переменной — enum. Возможные значения: – Forbidden — запрет на использование ISUP; – Optional — принимает, но не отправляет ISUP; – Mandatory — принудительное использование ISUP. Значение по умолчанию — Optional
sp-number	Номер сервисной платформы. Тип переменной — uint. Значение по умолчанию — значение для оператора
strong-link	Флаг приема входящих вызовов с external-host независимо от порта источника. Если значение данного параметра — 0, то вызовы принимаются с любого порта шлюза из внешней сети оператора. Тип переменной — bool.

Параметр	Значение параметра
	Значение по умолчанию — 1. **Примечание:** маршрутизация вызова в сторону оператора всегда осуществляется только на порт, заданный в параметре external-host, и не зависит от значения strong-link
transport	Тип транспортного протокола для исходящих вызовов. Тип переменной — enum. Возможные значения: – UDP; – TCP. Значение по умолчанию — значение для оператора. **Примечание:** если параметру не задано значение, то будет использоваться тот же протокол, что и во входящем запросе

Внимание! Если в маршруте используются шлюзы с одним IP адресом, то нельзя задавать параметру strong link значение 0.

Пример настройки параметров маршрута:

```
SBC> sbc operator id 0
sbc operator id 0> route rule 0
sbc operator id 0 route id 0> external-host 1.1.1.1:5060 external-gate ExtGate1
sbc operator id 0 route id 0> internal-gate IntGate1 mcu-network ExtMCU1
sbc operator id 0 route id 0> ping-timeout 10000 end
sbc operator id 0 route id 0> show
  internal-gate              'IntGate1'
  external-host              '1.1.1.1:5060'
  external-gate              'ExtGate1'
  mcu-network          'ExtMCU1'
  ping-timeout         10000
```

6.2.2 Медиашлюзы оператора

Настройка медиашлюзов определяет набор разрешенных подсетей и портов для приема медиатрафика от оператора.

Параметр media-gateway является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

```
SBC> sbc operator id 0
sbc operator id 0> media-gateway
sbc operator media-gateway>
  commit           apply modifications
```

```
  rollback         cancel modifications
  show             show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  resize           resize vector and if need, append it by default values.
  insert           insert element before 'idx' and select it for editing.
  end              return to parent
sbc operator media-gateway>
```

Чтобы войти в секцию настройки медиашлюза, необходимо в разделе sbc operator id # ввести имя массива media-gateway и выбрать существующий медиашлюз или создать новый.

Пример вывода информации о медиашлюзах:

```
SBC> sbc operator id 0
sbc operator id 0> media-gateway
sbc operator media-gateway> show
[size=0]

sbc operator media-gateway> insert 0
sbc operator media-gateway 0>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  host             * Set IP-network from which media is allowed
  port-range        * Set range of UDP ports from which media is allowed
  end              select parent
sbc operator media-gateway 0>
```

В таблице 14 приведены параметры медиашлюзов оператора и их описания.

Таблица 14 — Параметры медиашлюзов оператора

Параметр	**Значение параметра**
host **(*)**	Подсеть, которой разрешено передавать медиа. Обязательный параметр. Значение — ip/mask
port-range **(*)**	Диапазон разрешенных портов. Обязательный параметр. Тип переменной — array

Массив port range задает диапазон разрешенных медиапортов:

```
sbc operator media-gateway 0> port-range
sbc operator media-gateway port-range> insert 0
sbc operator media-gateway port-range 0>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  port-count       * Set port count
  port-start       * Set start port
  end              select parent
sbc operator media-gateway port-range 0>
sbc operator media-gateway port-range 0> port-start 17000 port-count 1000 end
sbc operator media-gateway port-range> show
[size=1]
  0
    port-start         17000
    port-count          1000

sbc operator media-gateway port-range>
```

В таблице 15 приведены параметры диапазона разрешенных портов.

Таблица 15 — Параметры диапазона разрешенных портов

Параметр	Значение параметра
port-start (*)	Начальный номер порта. Обязательный параметр. Тип переменной — uint. Диапазон значений: 1-65535
port-count (*)	Количество портов в диапазоне. Обязательный параметр. Тип переменной — uint. Диапазон значений: 1-65535

Примечание: значение последнего в диапазоне порта не должно превышать 65535, т.е. сумма port start и port count не должна быть больше 65535.

6.3 Сервисная платформа

Сервисная платформа определяет конфигурацию внутренней сети оператора.

Чтобы войти в подраздел настройки медиашлюза, необходимо в узле sbc ввести service-platform id #.

```
SBC> sbc service-platform id 0
sbc service-platform id 0>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  description       Set description
  gates             * Configure service platform gates params
  mcu-network       * Set MCU network ID for internal media
  media-profile     Set media profile ID for internal network
  no-ping-cause     Setup release cause for unreachable SIP trunks
  no-response-cause Setup release cause on no response calls
  ping-timeout      Set sip-ping interval
  polling-timeout   Set polling timeout
  reroute-attempts  Set default max reroute attempts for service platform
  rerouting-cause   Set rerouting reasons for service platform
  sip-i-enabled     Set SIP-I/SIP-T mode
  transport         Select transport type for outgoing calls
  default           set parameter to default value
  end               return to parent
  show-state        Show current state of the object
sbc service-platform id 0>
```

В таблице 16 приведен перечень параметров сервисной платформы.

Таблица 16 — Параметры сервисной платформы

Параметр	Значение параметра
description	Текстовое описание сервисной платформы. Тип переменной — string. Значение по умолчанию — «», пустая строка
gates (*)	Список шлюзов сервисной платформы (см. п. **Ошибка! Источник ссылки не найден.** «**Ошибка! Источник ссылки не найден.**»). Обязательный параметр. Тип переменной — array
mcu-network (*)	Идентификатор подсети MCU, на которую ожидается прием медиа из внутренней сети. Обязательный параметр. Тип переменной — string

Параметр	Значение параметра
media-profile	Имя медиапрофиля для внешней сети (см. п. 6.7 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — string. Значение по умолчанию — 0
no-ping-cause	Причина отбоя при потере связи. Тип переменной — uint. **Примечание:** значение должно быть не меньше 400. Значение по умолчанию — 500
no-response-cause	Причина отбоя при отсутствии ответа. Тип переменной — uint. **Примечание:** значение должно быть не меньше 400. Значение по умолчанию — 500
ping-timeout	Период пингования шлюзов на сервисной платформе. Тип переменной — uint, измеряется в миллисекундах, мс. Диапазон значений: 0-3600000 мс. Значение по умолчанию — 0, пингование отключено
polling-timeout	Период поллинга вызовов. Тип переменной — uint, измеряется в миллисекундах, мс. Диапазон значений: 0-3600000 мс. Значение по умолчанию — значение параметра, заданное для sbc
reroute-attempts	Максимальное количество попыток перемаршрутизации для сервисной платформы. Тип переменной — uint. Значение по умолчанию — значение параметра, заданное для sbc
rerouting-cause	Список причин для перемаршрутизации вызова. Тип переменной — array
sip-i-enabled	Режим работы ISUP. Тип переменной — enum. Возможные значения: – Forbidden — запрет на использование ISUP; – Optional — решение по ISUP перекладывается на вторую сторону; – Mandatory — обязательное использование ISUP. Значение по умолчанию — Optional
transport	Тип транспортного протокола для исходящих вызовов. Тип переменной — enum. Возможные значения:

Параметр	Значение параметра
	– UDP; – TCP. Значение по умолчанию — Undefined. **Примечание:** если параметру не задано значение, то будет использоваться тот же протокол, что и во входящем плече

Пингование шлюзов сервисной платформы реализовано путем периодической посылки SIP-сообщений OPTIONS. По умолчанию все шлюзы доступны, если задан ненулевой период пингования, то доступны только те, которые отвечают на SIP-запросы.

Пример настройки:

```
SBC> sbc service-platform id 0
sbc service-platform id 0>
sbc service-platform id 0> ping-timeout 5000
sbc service-platform id 0> polling-timeout 300000
sbc service-platform id 0>
```

6.4 Шлюзы сервисной платформы

Сервисная платформа содержит набор шлюзов для сигнализации, определяемых приоритетами и весами.

Параметр gates является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в секцию настройки шлюза сервисной платформы, необходимо в разделе sbc service-platform id # ввести имя массива gates и выбрать существующий медиашлюз или создать новый.

```
SBC> sbc service-platform id 0
sbc service-platform id 0>
sbc service-platform gates>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  resize           resize vector and if need, append it by default values.
  remove           remove element from vector by index.
```

```
  moveto          move element to another position.
  <0-0>           select element by index for editing.
  insert          insert element before 'idx' and select it for editing.
  end             return to parent
sbc service-platform gates> 0
sbc service-platform gates 0>
  commit          apply modifications
  rollback        cancel modifications
  show            show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  internal-host     * Set IP:port internal Gateway
  mcu-network       Set MCU network ID for external media
  media-profile     Set media profile ID for external network
  ping-timeout      Set sip-ping interval
  polling-timeout   Set polling timeout
  priority          * Set gateway priority
  transport         Select transport type for outgoing calls
  weight            * Set gateway weight
  default         set parameter to default value
  end             select parent
sbc service-platform gates 0>
```

В таблице 17 приведен перечень параметров шлюзов сервисной платформы.

Таблица 17 — Параметры шлюзов сервисной платформы

Параметр	Значение параметра
internal-host	Адрес шлюза сервисной платформы, который будет принимать и отправлять вызовы. Обязательный параметр. Тип переменной — ip:port
mcu-network	Имя подсети MCU, на которую ожидается прием медиа из внутренней сети. Тип переменной — string. Значение по умолчанию — значение для сервисной платформы
media-profile	Имя медиапрофиля для внешней сети. Тип переменной — string. Значение по умолчанию — значение для сервисной платформы
ping-timeout	Период пингования шлюзов сервисной платформы. Тип переменной — uint, измеряется в миллисекундах, мс. Диапазон значений: 0-3600000 мс.

Параметр	Значение параметра
	Значение по умолчанию — значение для сервисной платформы **Примечание:** если значение не 0, то доступны только те гейты, которые отвечают на запрос OPTIONS
polling-timeout	Период поллинга вызовов. Тип переменной — uint, измеряется в миллисекундах, мс. Диапазон значений: 0-3600000 мс. Значение по умолчанию — значение для сервисной платформы
priority (*)	Приоритет шлюза. Обязательный параметр. Тип переменной — uint. Диапазон значений: 0-100. **Примечание:** 0 — наивысший приоритет, 100 — самый низкий
transport	Тип транспортного протокола для исходящих вызовов. Тип переменной — enum. Возможные значения: – UDP; – TCP. Значение по умолчанию — Undefined. **Примечание:** если значение не задано, то используется тот же протокол, что и во входящем плече
weight (*)	Вес шлюза. Используется, когда у шлюзов равный приоритет. Обязательный параметр. Тип переменной — uint. **Примечание:** все шлюзы с весом 0 считаются недоступными

Примечание: по умолчанию значения общих для сервисной платформы и шлюзов параметров совпадают, если при настройке гейтов не указать иное.

В рамках сервисной платформы выбор шлюза, на который будут отправляться вызовы, производится в соответствии с приоритетами. Если имеется несколько шлюзов с одинаковым приоритетом, то случайным образом выбирается конкретный шлюз.

Вероятность выбора каждого доступного шлюза рассчитывается по формуле: P = gateWeight / sumWeightsAvailableGatesSamePrior.

1. GateWeight — вес конкретного шлюза/

2. SumWeightsAvailableGatesSamePrior — сумма весов всех доступных шлюзов с тем же приоритетом.

Пример настройки:

```
SBC> sbc service-platform id 0
sbc service-platform id 0> gates
sbc service-platform gates> show
[size=0]

sbc service-platform gates> resize 1
sbc service-platform gates> 0
sbc service-platform gates 0> internal-host 1.1.1.1:5060
sbc service-platform gates 0> priority 0
sbc service-platform gates 0> weight 50
sbc service-platform gates 0> end
sbc service-platform gates> show
[size=1]
 0
   internal-host            '1.1.1.1:5060'
   priority          0
   weight             50

sbc service-platform gates> insert 1 internal-host 1.1.1.2:5061 priority 1 weight 100
mcu-network ExtMCU1 media-profile ExtMedia1
sbc service-platform gates 1> end
sbc service-platform gates> show
[size=2]
 0
   internal-host            '1.1.1.1:5060'
   priority          0
   weight             50
 1
   internal-host            '1.1.1.2:5061'
   priority          1
   weight            100
   mcu-network               'ExtMCU1'
   media-profile             'ExtMedia1'

sbc service-platform gates> commit
```

6.5 MCU

Multipoint Control Unit (MCU) отвечает за контроль медиапотоков.

Чтобы войти в подраздел параметров MCU, необходимо в узле sbc ввести mcu id #.

```
SBC> sbc mcu id 0
sbc mcu id 0>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  description       Set description
  mcu-ip            * Set MCU host
  mcu-port          * Set MCU port
  mcu-subnet        * Configure MCU interfaces
  default           set parameter to default value
  end               return to parent
  block             Block the object
  show-state        Show current state of the object
  unblock           Unblock the object
sbc mcu id 0>
```

В таблице 18 приведено описание параметров MCU.

Таблица 18 — Параметры MCU

Параметр	**Значение параметра**
description	Текстовое описание MCU. Тип переменной — string. Значение по умолчанию — «», пустая строка
mcu-ip **(*)**	IP-адрес канала управления MCU. Обязательный параметр. Тип переменной — ip
mcu-port **(*)**	Порт канала управления MCU. Обязательный параметр. Тип переменной — uint
mcu-subnet	Настройка интерфейсов MCU (см. п. 6.6 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — array

Пример настройки:

```
SBC> sbc mcu id 0
sbc mcu id 0> description TestMCU
sbc mcu id 0> mcu-host
sbc mcu id 0> 10.0.0.1
sbc mcu id 0> mcu-port 1951
sbc mcu id 0> commit
```

6.6 Подсети MCU

Подсеть MCU определяет настройки интерфейсов MCU для приема медиа от оператора и отправки медиа к оператору, позволяет ограничивать количество одновременных аудио и видеосессий.

Параметр mcu-subnet является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в подраздел настройки подсети MCU, необходимо в подразделе sbc mcu id # ввести имя массива mcu-subnet и выбрать существующую подсеть или создать новую.

```
SBC> sbc mcu id 0
sbc mcu id 0> mcu-subnet
sbc mcu mcu-subnet>
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  resize           resize vector and if need, append it by default values.
  remove           remove element from vector by index.
  moveto           move element to another position.
  <0-0>            select element by index for editing.
  insert           insert element before 'idx' and select it for editing.
  end              return to parent
sbc mcu mcu-subnet> 0
sbc mcu mcu-subnet 0>
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  audio-count      * Set audio sessions max count
  mcu-media-ip     * Set MCU interface address for media
  name             * Set network name
  video-count      * Set video sessions max count
  end              select parent
```

В таблице 19 приведены параметры подсети MCU.

Таблица 19 — Параметры подсети MCU

Параметр	Значение параметра
audio-count (*)	Количество аудиосессий на MCU. Обязательный параметр.

Параметр	Значение параметра
	Тип переменной — uint. Диапазон значений: 0-32767
mcu-media-ip (*)	IP-адрес интерфейса, на котором MCU будет занимать порты для приема и отправки медиа. Обязательный параметр. Тип переменной — ip
name (*)	Имя подсети. Обязательный параметр. Тип переменной — string
video-count (*)	Количество видеосессий на MCU. Обязательный параметр. Тип переменной — uint. Диапазон значений: 0-32767

Внимание! Суммарное количество сессий audio count + video count не должно превышать число 32767.

Пример настройки:

```
SBC> sbc mcu id 0
sbc mcu id 0> mcu-subnet
sbc mcu mcu-subnet> resize 1
sbc mcu mcu-subnet> show
[size=1]
  0

sbc mcu mcu-subnet> 0
sbc mcu mcu-subnet 0> name MCU-Net1
sbc mcu mcu-subnet 0> mcu-media-host 1.1.1.1
sbc mcu mcu-subnet 0> audio-count 1000
sbc mcu mcu-subnet 0> video-count 555
sbc mcu mcu-subnet 0> end
sbc mcu mcu-subnet> show
[size=1]
  0
    name              'MCU-Net1'
    mcu-media-host          '1.1.1.1'
    audio-count             1000
    video-count             555
sbc mcu mcu-subnet> commit
```

6.7 Медиапрофили

Медиапрофиль определяет набор кодеков для приема медиа от оператора и передачи медиа к оператору.

Чтобы войти в подраздел медиапрофиля, необходимо в узле sbc ввести media profile id #.

```
SBC> sbc media-profile id 0
sbc media-profile id 0>
  commit          apply modifications
  rollback        cancel modifications
  show            show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  audio           Configure audio codecs list
  dscp            Set DSCP value
  dtmf            DTMF transmission type
  forced-proxy     Enable force media proxy
  video           Configure video codecs list
  default         set parameter to default value
  end             return to parent
sbc media-profile id 0>
```

В таблице 20 приведены параметры медиапрофилей.

Таблица 20 — Параметры медиапрофилей

Параметр	Значение параметра
forced-proxy	Флаг принудительного проксирования медиатрафика. Тип переменной — bool. Значение по умолчанию — 1, проксирование включено
dtmf	Тип передачи DTMF. Тип переменной — enum. Возможные значения: – TRANSPARENT — передача в исходном виде; – SIP_INFO — передача в SIP-запросе INFO; – RFC2833 — передача в RTP в формате RFC 2833; – RFC2833_OR_SIP_INFO — передача в RTP в формате RFC 2833, если вторая сторона не поддерживает такой формат, то в SIPзапросе INFO. Значение по умолчанию — TRANSPARENT
dscp	Формат DiffServ Code Point.

Параметр	Значение параметра
	Тип переменной — uint. Диапазон значений: 0-63
audio	Списки аудиокодеков (см. п. 6.7.1 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — array
video	Списки видеокодеков (см. п. 6.7.2 «**Ошибка! Источник ссылки не найден.**»). Тип переменной — array

Пример настройки:

```
SBC> sbc media-profile id 0
sbc media-profile id 0>
sbc media-profile id 0> forced-proxy 1
sbc media-profile id 0> commit
```

6.7.1 Списки аудиокодеков

В данной секции настраивается список аудиокодеков для медиапрофиля.

Параметр audio является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в секцию настройки списка аудиокодеков, необходимо в подразделе sbc media-profile id # ввести имя массива audio.

```
SBC> sbc media-profile id 0
sbc media-profile id 0> audio
sbc media-profile audio>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  allowed           Set allowed audio codecs list
  disable-codecs    Disable audio codecs
  mandatory         Set mandatory audio codecs list
  prohibited        Set prohibited audio codecs list
  supported         Set supported audio codecs list
  default           set parameter to default value
  end               select parent
sbc media-profile audio>
```

В таблице 21 приведен перечень параметров для списков аудиокодеков и их описания.

Таблица 21 — Параметры списка аудиокодеков

Параметр	Значение параметра
disable-codecs	Флаг запрета всех кодеков. Тип переменной — bool. Значение по умолчанию — 0
allowed	Список разрешенных кодеков. Кодеки, которые можно передавать удаленной стороне и которые могут присутствовать при входящем вызове от оператора. Если список пустой, то разрешены любые кодеки. Тип переменной — array, элементы — названия кодеков string
mandatory	Список обязательных кодеков. Список кодеков, заявляемый при исходящем вызове (в сторону оператора). В случае входящего вызова хотя бы один из обязательных кодеков должен присутствовать в списке кодеков от удаленной стороны, и должен находиться в списке разрешенных. Если список пустой, то обязательных кодеков нет. Тип переменной — array, элементы — названия кодеков string
prohibited	Список запрещенных кодеков. Кодеки, которые нельзя передавать удаленной стороне и которые не могут присутствовать в списке при входящем вызове от оператора. Если список пустой, то разрешены любые кодеки. Тип переменной — array, элементы — названия кодеков string
supported	Список поддерживаемых кодеков. Кодеки, которые удаленная сторона точно поддерживает. При входящем вызове эти кодеки являются разрешенными. Если список пустой, то о поддерживаемых кодеках ничего не известно. Тип переменной — array, элементы — названия кодеков string

Примечание: при формировании того или иного списка аудиокодеков используются кодеки раздела sbc audio-codec id # (см. п. 6.8 «**Ошибка! Источник ссылки не найден.**»).

Пример настройки:

```
SBC> sbc media-profile id 0
```

```
sbc media-profile id 0> audio
sbc media-profile audio> allowed insert 0 1 end
sbc media-profile audio> allowed insert 1 2 end
sbc media-profile audio> allowed insert 2 3 end
sbc media-profile audio> mandatory insert 0 1 end
sbc media-profile audio> prohibited insert 0 9 end
sbc media-profile audio> show
mandatory [size=1] 1
supported [size=0]
allowed [size=3] 1, 2, 3
prohibited [size=1] 9
sbc media-profile audio> commit
```

6.7.2 Списки видеокодеков

В данной секции настраивается список видеокодеков для медиапрофиля.

Параметр video является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами»).

Чтобы войти в секцию настройки списка видеокодеков, необходимо в подразделе sbc media profile id # ввести имя массива video:

```
sbc media-profile id 0> video
sbc media-profile video>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  allowed          Set allowed video codecs list
  disable-codecs   Disable video codecs
  mandatory        Set mandatory video codecs list
  prohibited       Set prohibited video codecs list
  supported        Set supported video codecs list
  default          set parameter to default value
  end              select parent
sbc media-profile video>
```

В таблице 22 приведен перечень параметров для списков видеокодеков и их описания.

Таблица 22 — Параметры списка видеокодеков

Параметр	Значение параметра
disable-codecs	Флаг запрета всех кодеков. Тип переменной — bool. Значение по умолчанию — 0
allowed	Список разрешенных кодеков. Такие кодеки можно передавать удаленной стороне, и они могут присутствовать при входящем вызове от оператора. Тип переменной — array, элементы — названия кодеков string. **Примечание:** при пустом списке разрешены все кодеки
mandatory	Список обязательных кодеков. Такие кодеки заявляются при исходящем вызове в сторону оператора. При входящем вызове хотя бы один из обязательных кодеков должен присутствовать в списке от удаленной стороны и находиться в списке разрешенных. Тип переменной — array, элементы — названия кодеков string. **Примечание:** при пустом списке обязательных кодеков нет
prohibited	Список запрещенных кодеков. Такие кодеки нельзя передавать удаленной стороне, и они не могут находиться в списке при входящем вызове от оператора. Тип переменной — array, элементы — названия кодеков string. **Примечание:** при пустом списке разрешены все кодеки
supported	Список поддерживаемых кодеков. Кодеки, которые удаленная сторона точно поддерживает. При входящем вызове эти кодеки являются разрешенными. Тип переменной — array, элементы — названия кодеков string. **Примечание:** при пустом списке о поддерживаемых кодеках нет данных

Примечание: при формировании того или иного списка видеокодеков используются кодеки раздела sbc video-codec id # (см. п. 6.9 «**Ошибка! Источник ссылки не найден.**»).

Пример настройки:

```
SBC> sbc media-profile id 0
sbc media-profile id 0> video
sbc media-profile video> allowed insert 0 1 end
sbc media-profile video> allowed insert 1 2 end
sbc media-profile video> allowed insert 2 3 end
sbc media-profile video> mandatory insert 0 1 end
sbc media-profile video> prohibited insert 0 9 end
```

```
sbc media-profile video> show
mandatory [size=1] 1
supported [size=0]
allowed [size=3] 1, 2, 3
prohibited [size=1] 9
sbc media-profile video> commit
```

6.8 Аудиокодеки

Настроенные кодеки используются для списков медиапрофиля. См. п. 6.7.1 «Списки аудиокодеков».

Чтобы войти в подраздел аудиокодеков, необходимо в узле sbc ввести audio codec id #.

```
SBC> sbc audio-codec id 0
sbc audio-codec id 0>
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  channels-count    Set audio channels count
  codec             * Set codec name
  description       Set description
  rx-bitrate        Set rx bitrate
  sample-rate       Set sample rate
  tx-bitrate        Set tx bitrate
  default           set parameter to default value
  end               return to parent
sbc audio-codec id 0>
```

В таблице 23 приведен перечень параметров аудиокодеков и их описание.

Таблица 23 — Параметры аудиокодека

Параметр	Значение параметра
description	Текстового описание аудиокодека. Тип переменной — string
channels-count	Количество выделенных каналов. Тип переменной — uint. Значение по умолчанию — 1
codec (*)	Официальное название кодека согласно стандартам RFC. Обязательный параметр. Тип переменной — string
rx-bitrate	Скорость на прием.

Параметр	Значение параметра
	Тип переменной — uint, измеряется в битах в секунду, бит/с. Диапазон значений: 0-1300000 бит/с. Значение по умолчанию — 0
sample-rate	Частота дискретизации. Тип переменной — uint, измеряется в герцах, Гц. Диапазон значений: 0-96000 Гц. Значение по умолчанию — 8000 Гц
tx-bitrate	Скорость на передачу. Тип переменной — uint, измеряется в битах в секунду, бит/с. Диапазон значений: 0-1300000 бит/с. Значение по умолчанию — 0

Пример настройки:

```
SBC> sbc audio-codec id 0
sbc audio-codec id 0> codec G729
sbc audio-codec id 0> channels-count 1000
sbc audio-codec id 0> rx-bitrate 8000
sbc audio-codec id 0> tx-bitrate 8000
sbc audio-codec id 0> show
  codec              'G729'
  rx-bitrate         8000
  tx-bitrate         8000
  channels-count             1000
sbc audio-codec id 0> commit
transaction result: success
```

6.9 Видеокодеки

Настроенные кодеки используются для списков медиапрофиля. См. п. 6.7.2 «Списки видеокодеков».

Чтобы войти в подраздел видеокодеков, необходимо в узле sbc ввести video codec id #.

```
SBC> sbc video-codec id 0
sbc video-codec id 0>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
```

```
  channels-count    Set video channels count
  codec           * Set codec name
  description       Set description
  height          Set height of the frame
  rx-bitrate      Set rx bitrate
  sample-rate       Set sample rate
  tx-bitrate      Set tx bitrate
  width           Set width of the frame
  default         set parameter to default value
  end             return to parent
sbc video-codec id 0>
```

В таблице 24 приведен перечень параметров видеокодеков и их описание.

Таблица 24 — Параметры видеокодека

Параметр	**Значение параметра**
channels-count	Количество каналов. Тип переменной — uint. Значение по умолчанию — 1
codec (*)	Официальное название кодека согласно стандартам RFC. Обязательный параметр. Тип переменной — string
description	Текстовое описание видеокодека. Тип переменной — string
height	Высота кадра. Тип переменной — uint, измеряется в пикселях, пк. Диапазон значений: 0-4320 пк. Значение по умолчанию — 0
rx-bitrate	Скорость на прием. Тип переменной — uint, измеряется в битах в секунду, бит/с. Диапазон значений: 0-1300000 бит/с. Значение по умолчанию — 0
sample-rate	Частота дискретизации. Тип переменной — uint, измеряется в герцах, Гц. Диапазон значений: 0-96000 Гц. Значение по умолчанию — 90000 Гц
tx-bitrate	Скорость на передачу. Тип переменной — uint, измеряется в битах в секунду, бит/с. Диапазон значений: 0-1300000 бит/с. Значение по умолчанию — 0

Параметр	Значение параметра
width	Ширина кадра. Тип переменной — uint, измеряется в пикселях, пк. Диапазон значений: 0-7680 пк. Значение по умолчанию — 0

Пример настройки:

```
SBC> sbc video-codec id 0
sbc video-codec id 0> codec H264
sbc video-codec id 0> channels-count 1000
sbc video-codec id 0> rx-bitrate 240000
sbc video-codec id 0> height 240
sbc video-codec id 0> width 320
sbc video-codec id 0> commit
```

6.10 Маршрутизация и модификация контекста

Внутренняя маршрутизация позволяет задавать различные варианты направления вызовов с настраиваемой модификацией контекста вызова.

Общее описание маршрутизации:

1. Определяется источник вызова из полей сообщения SIP. При поступлении вызова проверяется заголовок Host источника вызова, чтобы определить на принадлежность сервисной платформе или оператору. Если вызов исходит от оператора, то далее определяется медиашлюз. После этого проверяются ограничения оператора, не заблокирован ли маршрут.

2. Выполняется маршрутизация и модификация контекста вызова согласно заданным правилам. Если заданы маски, проверяются условия попадания под маску. Далее контекст вызова модифицируется, следуя настройкам правил внешней маршрутизации.

3. Определяется получатель вызова. После модификации контекста проверяется компонентный адрес получателя вызова: попадает ли под маску маршрута оператора или маску сервисной платформы. Если получателем вызова является оператор, то подтверждение проверяются ограничения, заданные маршрутом оператора. далее ожидается подтверждение, что ли данный оператор не заблокирован. Если

получателем является сервисная платформа, то начинается процесс выбора шлюза согласно заданным приоритетам и весам. После этого вызов направляется по найденному маршруту.

4. Если принят запрос от другой стороны на завершение вызова до получения сообщения 200 OK, то применяется алгоритм перемаршрутизации.

Примечание: упомянутые параметры подробно описаны в следующих секциях:

– Host оператора — параметр host в настройках медиашлюзов оператора, см. п. 6.2.2 «Медиашлюзы оператора»;

– маршруты оператора — параметр route в настройках оператора, см. п. 6.2.1 «Правила маршрутизации для оператора»;

– Host сервисной платформы — параметр gates в настройках сервисной платформы, см. п. 6.4 «Шлюзы сервисной платформы».

Настройка правил маршрутизации с модификацией контекста выполняется в разделе sbc routing.

Пример вывода информации о правилах маршрутизации:

```
sbc> routing
sbc routing> show
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  flexible-direction Configure flexible direction params
  action            Configure action templates
  mask              Configure mask templates for incoming parameters
  rerouting-cause   Set default rerouting reasons list
  rule              Configure routing rules
  default           set parameter to default value
  end               return to parent
sbc routing> show recursive
  rerouting-cause [size=0]
  mask [size=0]
  action [size=0]
  rule [size=2]
    rule 0
      name            'SP.1->Op.0'
      source-address          'Ngn.SBC.SP.1'
```

```
    called-party-number            '5029'
    calling-party-number           '4000[13456]'
    destination-address            'Ngn.SBC.Op.0.Route.0'
    reroute-rules [size=0]
  rule 1
    name            'Op.0->SP.1'
    source-address          'Ngn.SBC.Op.0.Route.0'
    called-party-number            '40001|40003'
    calling-party-number           '5029'
    destination-address            'Ngn.SBC.SP.1'
    reroute-rules [size=0]
sbc routing>
```

В таблице 25 приведен перечень параметров маршрутизации, а также их описания.

Таблица 25 — Параметры маршрутизации

Параметр	Значение параметра
flexible-direction	Настройка компоненты Flexible Direction (см. п. 6.10.6 «**Ошибка! Источник ссылки не найден.**»)
action	Настройка шаблонов действий для модификации контекста вызова (см. п. 6.10.3 «**Ошибка! Источник ссылки не найден.**»)
mask	Настройка шаблонов масок для правил маршрутизации (см. п. 6.10.2 «**Ошибка! Источник ссылки не найден.**»)
rerouting-cause	Настройка списка причин для перемаршрутизации (см. п. 6.10.1 «**Ошибка! Источник ссылки не найден.**»)
rule	Настройка правил маршрутизации (см. п. 6.10.5 «**Ошибка! Источник ссылки не найден.**»)

6.10.1 Общие причины для перемаршрутизации

В данной секции настраивается общий список причин для перемаршрутизации. Эти причины используются, если в правиле в явном виде не заданы причины для перемаршрутизации.

Чтобы создать причину для перемаршрутизации, необходимо в узле sbc routing ввести имя массива rerouting-cause и выбрать существующую причину или создать новую.

```
sbc routing> rerouting-cause
  init      vector initialization.
  resize    resize vector and if need, append it by default values.
  remove    remove element from vector by index.
```

```
  moveto    move element to another position.
  <0-0>     select element by index for editing.
  insert    insert element before 'idx' and select it for editing.
  end       return to parent
sbc routing> rerouting-cause
vector<int>> resize 1
vector<int>> 0 34
vector<int>> show
[size=1] 34
```

Примечание: указанные здесь причины перемаршрутизации используются, если не заданы в параметрах самого правила (см. п. 6.10.4 «**Ошибка! Источник ссылки не найден.**»).

6.10.2 Шаблоны масок для правил маршрутизации

В системе различают два типа шаблонов масок:

1. Predefined — предварительно определенный шаблон маски. Задается только в массиве mask. Определен глобально для всего раздела sbc.

2. Raw — шаблон маски, определенный в секции. Задается в секциях по необходимости. Определен локально, только для секции создания.

Параметр mask является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в секцию шаблонов масок, необходимо в разделе sbc routing ввести имя массива mask и выбрать существующий шаблон или создать новый.

```
ssbc routing mask 0>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
  label            Set label for mask template
  mask             Set mask template
  default          set parameter to default value
  end              select parent
sbc routing mask 0>
```

В таблице 26 приведены параметры шаблона маски и их описание.

Таблица 26 — Параметры шаблона маски

Параметр	Значение параметра
label (*)	Имя шаблона маски. Тип переменной — string. **Примечание:** вызов шаблона происходит по его имени
mask (*)	Шаблон маски. Тип переменной — regex

Пример создания шаблона маски:

```
sbc routing mask> resize 1
sbc routing mask> 0
sbc routing mask 0> label Office_2 mask ^8124494727$
sbc routing mask 0> show
label          'Office_2'
mask           '^8124494727$'
sbc routing mask 0> end
```

6.10.3 Шаблоны действий

В системе различают два типа шаблона действий:

1. Predefined — предварительно определенный шаблон действия. Задается только в массиве action. Определен глобально для всего раздела sbc.

2. Raw — шаблон маски, определенный в секции. Задается в секциях по необходимости. Определен локально, только для секции, где был создан.

Параметр action является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами»).

Чтобы войти в секцию шаблонов действий, необходимо в разделе sbc routing ввести имя массива action и выбрать существующий шаблон или создать новый.

```
sbc routing> action
  resize    resize vector and if need, append it by default values.
  insert    insert element before 'idx' and select it for editing.
  end       return to parent
sbc routing action 0>
  commit           apply modifications
  rollback         cancel modifications
  show             show current object
  show recursive   recursive show current object
  show config      show CLI command list for object
```

```
add-prefix       Add prefix
del-digits       Delete digits
label            Set label for action template
set-number       Set new number
default          set parameter to default value
end              select parent
```

В таблице 27 приведены параметры шаблона действий и их описание.

Таблица 27 — Параметры шаблона действий

Параметр	Значение параметра
add-prefix	Символы, которые будут добавлены в начало номера. Тип переменной — string
del-digits	Количество символов, которые будут удалены из начала номера. Тип переменной — uint
label **(*)**	Имя шаблона действия. Обязательный параметр. Тип переменной — string
set-number	Номер, который будет отображаться в качестве CdPN или CgPN. Тип переменной — string

Пример создания шаблона действий:

```
sbc routing action> resize 1
sbc routing action> 0
sbc routing action 0> del-digits 3
sbc routing action 0> add-prefix 812
sbc routing action 0> label A1
sbc routing action 0> show
label          'A1'
del-digits             3
add-prefix             '812'
sbc routing action 0>
```

В шаблоне из полученного номера удаляются первые три цифры, затем в начало номера добавляется префикс 812.

6.10.4 Правила перемаршрутизации

В этой секции задаются причины и последующие действия при перемаршрутизации, применимые к конкретному правилу. Если причины заданы, то

глобальные причины перемаршрутизации не учитываются (см. п. 6.10.1 «Общие причины для перемаршрутизации»).

Параметр reroute-rules является массивом, для него доступны стандартные команды , описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в секцию правил перемаршрутизации, необходимо в разделе sbc routing rule # ввести имя массива reroute-rules и выбрать существующее правило или создать новое:

```
sbc routing rule 1> reroute-rules
sbc routing rule reroute-rules> 1
reroute_rules 1>
  commit            apply modifications
  rollback          cancel modifications
  show              show current object
  show recursive    recursive show current object
  show config       show CLI command list for object
  rerouting-cause   Set rerouting reasons
  rerouting-rule    Set reroute rule name
  default           set parameter to default value
  end               select parent
reroute_rules 1>
```

В таблице 28 приведено описание параметров для правил перемаршрутизации.

Таблица 28 — Параметры правила перемаршрутизации

Параметр	Значение параметра
rerouting-cause	Список причин Q.850, при наличии которых осуществляется перемаршрутизация. Тип переменной — array, элементы — причины из Q.850 типа uint
rerouting-rule	Название правила перемаршрутизации. Тип переменной — string

Пример настройки правила:

```
reroute_rules 1> rerouting-rule second
reroute_rules 1> rerouting-cause insert 0 34
vector<int>> end
reroute_rules 0> show
rerouting-rule        'second'
rerouting-cause [size=1] 34
reroute_rules 1>
```

6.10.5 Правила маршрутизации

Правила выполняются по порядку сверху вниз, пока не найдется подходящее, закончатся правила, либо не найдется подходящее.

Параметр rule является массивом, для него доступны стандартные команды, описанные в п. 4.4.5 «Операции с массивами».

Чтобы войти в секцию правил маршрутизации, необходимо в разделе sbc routing ввести имя массива rule и выбрать существующие настройки или создать новые.

Пример вывода информации о правилах маршрутизации.

```
sbc routing> rule
  resize    resize vector and if need, append it by default values.
  remove    remove element from vector by index.
  moveto    move element to another position.
  <0-7>     select element by index for editing.
  insert    insert element before 'idx' and select it for editing.
  end       return to parent

sbc routing> rule 0
sbc routing rule 0>
  show                      show current object
  show recursive            recursive show current object
  show config               show CLI command list for object
  called-party-number       Set CdPN mask
  calling-party-category    Set CgPC mask
  calling-party-number      Set CgPN mask
  cdpn-template             Set CdPN mask template
  cgpc-template             Set CgPC mask template
  cgpn-template             Set CgPN mask template
  destination-address       Set destination direction
  gateid                    Set SBC LocalGate for outgoing traffic generate (Mandatory for
SP to SP call)
  local-gate                Set SBC LocalGate for incoming calls
  modify-cdpn--add-prefix   Add CdPN prefix
  modify-cdpn--del-digits   CdPN prefix-length to delete
  modify-cdpn--set-action   Set CdPN modify template
  modify-cdpn--set-number   Set new CdPN
  modify-cgpc--add-prefix   Add CgPC prefix
  modify-cgpc--del-digits   CgPC prefix-length to delete
  modify-cgpc--set-action   Set CgPC modify template
  modify-cgpc--set-category Set new CgPC, ISUP-R code
```

```
  modify-cgpn--add-prefix      Add CgPN prefix
  modify-cgpn--del-digits      CgPN prefix-length to delete
  modify-cgpn--set-action      Set CgPN modify template
  modify-cgpn--set-number      Set new CgPN
  name                         Set route rule name
  reroute-attempts             Set max reroute attempts
  reroute-label                Set reroute label
  reroute-rules                Configure reroute rules
  source-address               Set source component address mask
  source-address-template      Set source component address mask template
  default                      set parameter to default value
  end                          select parent
sbc routing rule 0>
```

В таблице 29 приведены параметры правил маршрутизации и их описание.

Таблица 29 — Параметры правила маршрутизации

Параметр	Значение параметра
name	Имя правила маршрутизации. Тип переменной — string
reroute-attempts	Количество попыток перемаршрутизации. Тип переменной — uint. **Примечание:** является значением по умолчанию для всех правил перемаршрутизации данного правила
reroute-label	Имя правила перемаршрутизации. Тип переменной — string
reroute-rules	Настройки параметров правил перемаршрутизации (см. п. 6.10.4 «**Ошибка! Источник ссылки не найден.**»)
source-address	Маска для компонентного адреса источника вызова. Тип переменной — regex. Возможные источники вызова: – NGN.SBC.Op.#.Route.# — вызов от оператора; – Op.# — идентификатор оператора; – Route.# — номер маршрута оператора; – NGN.SBC.SP.# — вызов от сервисной платформы; – SP.# — номер платформы
source-address-template	Шаблон для маски компонентного адреса источника вызова. Тип переменной — regex
gateId	Адрес шлюза для исходящего трафика. Тип переменной — ip:port.

Параметр	Значение параметра
	Примечание: поле обязательно требуется заполнить для вызовов от сервисной платформы на сервисную платформу
local-gate	Адрес шлюза для входящего трафика. Тип переменной — ip:port
called-party-number	Маска для номера вызываемого абонента. Тип переменной — string
calling-party-category	Маска для категории вызывающего абонента. Тип переменной — string
calling-party-number	Маска для номера вызываемого абонента. Тип переменной — regex
cdpn-template	Шаблон масок для номера вызываемого абонента. Тип переменной — regex
cgpc-template	Шаблон масок для категории вызывающего абонента. Тип переменной — regex
cgpn-template	Шаблон масок для номера вызывающего абонента. Тип переменной — regex
destination-address	Маска для компонентного адреса назначения вызова. Возможные адреса назначения вызова: – NGN.SBC.Op.#.Route.# — вызов от Оператора; – Op.# — идентификатор оператора; – Route# — номер маршрута оператора; – NGN.SBC.SP.# — вызов от сервисной платформы; – SP.# — номер сервисной платформы
modify-cdpn--add-prefix	Символы, которые будут добавлены в начало CdPN. Тип переменной — string
modify-cdpn--del-digits	Количество символов, которые будут удалены из начала CdPN. Тип переменной — uint
modify-cdpn--set-action	Шаблон модификации CdPN. Задается с помощью имени действия, которое осуществляется над CdPN. Тип переменной — string
modify-cdpn--set-number	Номер, который будет отображаться в качестве CdPN. Тип переменной — string
modify-cgpc--add-prefix	Символы, которые будут добавлены в начало CgPC. Тип переменной — string
modify-cgpc--del-digits	Количество символов, которые будут удалены из начала CgPC. Тип переменной — uint

Параметр	Значение параметра
modify-cgpc--set-action	Шаблон модификации CgPC. Задается с помощью имени действия, которое осуществляется над CgPC. Тип переменной — string
modify-cgpc--set-category	Категория, которая будет отображаться в качестве CgPC, код ISUP-R. Тип переменной — string
modify-cgpn--add-prefix	Символы, которые будут добавлены в начало CgPN. Тип переменной — string
modify-cgpn--del-digits	Количество символов, которые будут удалены из начала CgPN. Тип переменной — uint
modify-cgpn--set-action	Шаблон модификации CgPN. Задается с помощью имени действия, которое осуществляется над CgPN. Тип переменной — string
modify-cgpn--set-number	Номер, который будет отображаться в качестве CgPN. Тип переменной — string

Пример настройки правил маршрутизации:

```
sbc routing> show
 rerouting-cause [size=0]
 mask [size=0]

 action [size=0]

 rule [size=8]
  rule 0
   name              'SP.1->Op.0'
   source-address          'Ngn.SBC.SP.1'
   called-party-number             '5029'
   calling-party-number            '4000[13456]'
   destination-address            'Ngn.SBC.Op.0.Route.0'
   reroute-rules [size=0]

  rule 1
   name              'Op.0->SP.1'
   source-address          'Ngn.SBC.Op.0.Route.0'
   called-party-number             '40001|40003'
   calling-party-number            '5029'
   destination-address            'Ngn.SBC.SP.1'
   reroute-rules [size=0]
```

```
rule 2
  name              'SP.1->Op.1'
  source-address          'Ngn.SBC.SP.1'
  called-party-number             '40001|5029'
  calling-party-number            '400'
  modify-cdpn--add-prefix           '8'
  destination-address             'Ngn.SBC.Op.1.Route.0'
  reroute-rules [size=0]

rule 3
  name              'Op.1->SP.1'
  source-address          'Ngn.SBC.Op.1.Route.0'
  called-party-number             '40001'
  calling-party-number            '40000'
  destination-address             'Ngn.SBC.SP.1'
  reroute-rules [size=0]

rule 4
  source-address          'Ngn.SBC.Op.1.Route.0'
  called-party-number             '85029'
  calling-party-number            '40000'
  modify-cdpn--del-digits           1
  destination-address             'Ngn.SBC.Op.0.Route.0'
  reroute-rules [size=1]
    reroute-rules 0
      rerouting-rule              'To_82021'
      rerouting-cause [size=0]

rule 5
  source-address          'Ngn.SBC.Op.2.Route.0'
  destination-address             'Ngn.SBC.SP.1'
  reroute-rules [size=0]

rule 6
  source-address          'Ngn.SBC.SP.1'
  called-party-number             '50001'
  destination-address             'Ngn.SBC.Op.2.Route.0'
  reroute-rules [size=0]

rule 7
  name              'To_82021'
  reroute-label           'To_82021'
```

```
sbc routing> rule
```

6.10.6 Подсистема Flexible Direction

Внимание! Для корректной работы компоненты необходимы следующие настройки общих параметров (см. п. 6.11 «Общие глобальные параметры»):

1. Параметр routing enabled — значение 1.
2. Параметр allow unknown calls — значение 1.

Flexible Direction — подсистема, которая обрабатывает запросы пользователей и производит адресацию вызова при неверном форматировании. Если IP-адрес или порт ресурса указан некорректно: пропущены точки «.» или двоеточие «:», то подсистема обработает такую строку и маршрутизирует ее, если в запросе отсутствуют другие ошибки.

В поле username должна находиться строка формата: xxxxxxxxxxxxyyyy или xxxxxxxxxxxx.

1. x — формируют IP-адрес.
2. y — формируют номер порта.

Пример — строка 1921681001008080 преобразуется в 192.168.100.100:8080.

Чтобы войти в секцию подсистемы Flexible Direction, необходимо в узле sbc ввести routing flexible direction id #.

```
sbc routing> flexible-direction id 0
sbc routing flexible-direction id 0>
  commit              apply modifications
  rollback            cancel modifications
  show                show current object
  show-recursive      recursive show current object
  show-config         show CLI command list for object
  description         Set description
  dst-media-network   * Set media network name for call destination
  dst-media-profile   Set media profile ID for call destination
  local-gate          Set virtual SIP EP ID for calls
  src-media-network   * Set media network name for call source
  src-media-profile   Set media profile ID for call source
  default             set parameter to default value
  end                 return to parent
```

```
  block              Block the object
  show-state         Show current state of the object
  unblock            Unblock the object
sbc routing flexible-direction id 0>
```

В таблице 30 приведено описание параметров Flexible Direction.

Таблица 30 — Параметры Flexible Direction

Параметр	Значение параметра
description	Текстовое описание. Тип переменной — string
dst-media-network **(*)**	Имя подсети MCU для получателя вызова B. Обязательный параметр. Тип переменной — string
dst-media-profile	Имя медиапрофиля для получателя вызова B. Тип переменной — string
local-gate	Имя шлюза, с которого необходимо отправить вызов абоненту B. Тип переменной — string
src-media-network **(*)**	Имя подсети MCU для источника вызова A. Обязательный параметр. Тип переменной — string
src-media-profile	Имя медиапрофиля для источника вызова A. Тип переменной — string

6.11 Общие глобальные параметры

В разделе определяются общие параметры, которые действуют во всех сущностях sbc в качестве значений по умолчанию. Здесь также задаются параметры, необходимые для работы подсистемы Flexible Direction (см. п. 6.10.6 «Подсистема Flexible Direction»).

Чтобы войти в подраздел общих параметров, необходимо в узле sbc ввести common.

```
sbc common>
  show                 show current object
  show recursive       recursive show current object
  show config          show CLI command list for object
  allow-unknown-calls  Allow calls from unknow source (only for Flexible Direction)
  diversion-proxy      Enable Diversion header proxy
```

```
  dump-preanswer        Enable dump recording in preanswer
  force-add-q850        Enable forced addition of reason Q.850
  polling-timeout       Set polling timeout
  reroute-attempts      Set default max reroute attempts
  rerouting-cause       Set rerouting reasons
  routing-enable        Enable routing system
  sort-dumps            Enable distributed dump storage mode (modify param not allowed)
  stat-interval         Set statistics update period
  use-sp-mcu            Enable SP MCU support (modify param not allowed)
  default               set parameter to default value
  end                   return to parent
sbc common> show
  stat-interval               10
  polling-timeout             30000
  rerouting-cause [size=1] 18
sbc common>
```

В таблице 31 приведено описание общих параметров **common**.

Таблица 31 — Глобальные параметры sbc

Параметр	Значение параметра
allow-unknown-calls	Флаг приема вызовов от неизвестных источников. Влияет только на работу подсистемы Flexible Direction. Тип переменной — bool. Значение по умолчанию — 0. **Примечание:** для Flexible Direction требуется значение 1
diversion-proxy	Флаг использования прокси для заголовка Diversion. Тип переменной — bool. Значение по умолчанию — 0
dump-preanswer	Флаг записи действий, происходящих до получения ответа от другой стороны. Тип переменной — bool. Значение по умолчанию — 0
force-add-q850	Флаг принудительного использования причин Q.850. Тип переменной — bool. Значение по умолчанию — 0
polling-timeout	Период поллинга вызовов. Тип переменной — uint, измеряется в миллисекундах, мс. Значение по умолчанию — 0
reroute-attempts	Максимальное количество попыток перемаршрутизации в рамках сервисной платформы.

Параметр	Значение параметра
	Тип переменной — uint. Значение по умолчанию — 1. **Примечание:** если для сервисной платформы не задано значение для аналогичного поля, то будет использовано данное значение
rerouting-cause	Список причин, при наличии которых осуществляется перемаршрутизация. Тип переменной — array. Значение по умолчанию — [], пустой список
routing-enable	Флаг включения внутренней маршрутизации. Тип переменной — bool. Значение по умолчанию — 0. **Примечание:** для Flexible Direction требуется значение 1
sort-dumps	Флаг использования распределенного хранения дампов и поддержка SIP_Dumper. Тип переменной — bool. Значение по умолчанию — 0
stat-interval	Период обновления статистики. Тип переменной — uint, измеряется в секундах, с. Значение по умолчанию — 5 с
use-sp-mcu	Флаг поддержки сервисной платформы MCU. Влияет только на работу подсистемы FEC Proxy, которая обеспечивает помехоустойчивое кодирование RTP-пакетов. Тип переменной — bool. Значение по умолчанию — 0. **Примечание:** для FEC Proxy требуется значение 1

6.12 Просмотр состояния

Чтобы посмотреть состояния объекта, необходимо ввести команду show-state в разделе настройки этого объекта. В результате будет отображена информация, созданная подсистемой аварийной индикации.

Формат вывода информации: param = value.

Каждому полю состояния также соответствует переменная param.DT, которая хранит дату и время последнего изменения param.

Формат переменной: #Year #Mon #Day #Hour:#Min:#Sec.#Ms.

1. #Year — год, г., содержит четырьмя цифрами.

2. #Mon — месяц, мес., содержит двумя цифрами.
3. #Day — день, д., содержит двумя цифрами.
4. #Hour — час, ч, содержит двумя цифрами.
5. #Min — минута, мин, содержит двумя цифрами.
6. #Sec — секунда, с, содержит двумя цифрами.
7. #Ms — миллисекунда, мс, содержит тремя цифрами.

Просмотр состояний доступен для следующих объектов:

1. Оператор.
2. Маршрут оператора.
3. Подсеть MCU.

6.12.1 Состояние маршрута

Команду show state требуется выполнить в разделе sbc operator id # route id #.

В таблице 32 приведено описание состояний маршрута.

Таблица 32 — Состояния маршрута оператора

Переменная	Описание
OSTATE	Оперативное состояние. Тип переменной — bool
ASTATE	Административное состояние. Тип переменной — bool
Descrip	Текстовое описание оператора. Тип переменной — string
Addr	Адрес внешней подсети, который будет принимать и отправлять вызовы. Тип переменной — ip:port
inCPS	Текущее количество вызовов в секунду от оператора. Тип переменной — uint
inMCPS	Максимальное количество вызовов в секунду от оператора за период statT. Тип переменной — uint
noTr	Количество SIP-сообщений, для которых не найдена транзакция за период statT. Тип переменной — uint
uReq	Количество не известных SIP-сообщений за период statT. Тип переменной — uint

Переменная	Описание
statT	Период обновления статистики. Тип переменной — uint, измеряется в миллисекундах, мс

Пример использования:

```
sbc operator id 0 route id 0> show-state
ASTATE = 1
ASTATE.DT = 2020-02-13 12:48:07
Addr = 192.168.99.151:5060
Addr.DT = 2020-02-13 12:48:07
OSTATE = 1
OSTATE.DT = 2020-02-13 12:48:07
PSTATE = 3
PSTATE.DT = 2020-02-13 12:48:07
inCPS = 0
inCPS.DT = 2020-02-21 14:44:31
inMCPS = 0
inMCPS.DT = 2020-02-21 14:44:31
noTr = 903
noTr.DT = 2020-02-21 14:44:31
parsEr = 0
parsEr.DT = 2020-02-21 14:44:31
statT = 15
statT.DT = 2020-02-13 12:48:07
uReq = 0
uReq.DT = 2020-02-21 14:44:31
sbc operator id 0 route id 0>
```

6.12.2 Состояния MCU

Команду show state требуется выполнить в разделе sbc mcu id #.

В таблице 33 приведено описание состояний MCU.

Таблица 33 — Состояния MCU

Переменная	Описание
ASTATE	Административное состояние. Тип переменной — bool
OSTATE	Оперативное состояние. Тип переменной — bool
Descrip	Текстовое описание оператора. Тип переменной — string
Congess	Уровень текущей нагрузки.

Переменная	Описание
	Тип переменной — enum. Возможные значения: – 0 — NORMAL, нормальный уровень; – 1 — MINOR, нижний уровень; – 2 — MAJOR, верхний уровень; – 3 — CRITICAL, критический уровень
ExMxUsr	Флаг, проверяет, что в MCU количество аудиоканалов больше, чем суммарное количество пользователей на MCU. Тип переменной — bool

Пример использования:

```
sbc mcu id 0> show-state
ASTATE = 1
ASTATE.DT = 2020-02-13 12:48:07
Congess = 0
Congess.DT = 2020-02-21 15:56:36
ExMxUsr = 0
ExMxUsr.DT = 2020-02-13 12:48:07
OSTATE = 1
OSTATE.DT = 2020-02-13 12:48:07
sbc mcu id 0>
```

6.12.3 Состояние оператора

Команду show state требуется выполнить в разделе sbc operator id #.

В таблице 34 приведено описание состояний оператора.

Таблица 34 — Состояние оператора

Переменная	Описание
ASTATE	Административное состояние. Тип переменной — bool
OSTATE	Оперативное состояние. Тип переменной — bool
Descrip	Текстовое описание оператора. Тип переменной — string
lInCPS	Максимальное количество вызовов в секунду от оператора. Тип переменной — uint
lCurCal	Максимальное количество одновременных вызовов. Тип переменной — uint
lICrCl	Максимальное количество одновременных вызовов от оператора.

Переменная	Описание
	Тип переменной — uint
lOCrCl	Максимальное количество одновременных вызовов к оператору. Тип переменной — uint
inCPS	Текущее количество вызовов в секунду от оператора. Тип переменной — uint. Период перепосылки — по требованию
cur	Общее количество текущих вызовов. Тип переменной — uint. Период перепосылки — по требованию
in_cur	Количество текущих вызовов от оператора. Тип переменной — uint. Период перепосылки — по требованию
out_cur	Количество текущих вызовов к оператору. Тип переменной — uint. Период перепосылки — по требованию
exCPS	Превышение максимального количества вызовов в секунду от оператора. Тип переменной — bool. Период перепосылки — через 15 с после изменения состояния
exCur	Превышение максимального количества вызовов. Тип переменной — bool. Период перепосылки — через 15 с после изменения состояния
exICur	Превышение максимального количества вызовов от оператора. Тип переменной — bool. Период перепосылки — через 15 с после изменения состояния
exOCur	Превышение максимального количества вызовов к оператору. Тип переменной — bool. Период перепосылки — через 15 с после изменения состояния
uPT	Количество вызовов с неизвестным типом медиа за последние 15 сек. Тип переменной — uint. Период перепосылки — 15 с
uMedSrc	Количество вызовов от неизвестных источников медиа за последние 15 сек. Тип переменной — uint. Период перепосылки — 15 с

Переменная	Описание
ovBand	Количество вызовов с превышением битрейта за последние 15 сек. Тип переменной — uint. Период перепосылки — 15 с

Пример:

```
sbc operator id 0> show-state
ASTATE = 1
ASTATE.DT = 2020-02-13 12:48:07
OSTATE = 1
OSTATE.DT = 2020-02-13 12:48:07
exCPS = 0
exCPS.DT = 2020-02-13 12:48:07
exCur = 0
exCur.DT = 2020-02-13 12:48:07
exICur = 0
exICur.DT = 2020-02-13 12:48:07
exOCur = 0
exOCur.DT = 2020-02-13 12:48:07
inCPS = 0
inCPS.DT = 2020-02-21 14:18:11
lCurCal = -1
lCurCal.DT = 2020-02-13 12:48:07
lICrCl = -1
lICrCl.DT = 2020-02-13 12:48:07
lInCPS = -1
lInCPS.DT = 2020-02-13 12:48:07
lOCrCl = -1
lOCrCl.DT = 2020-02-13 12:48:07
ovBand = 1
ovBand.DT = 2020-02-21 14:19:41
uMedSrc = 0
uMedSrc.DT = 2020-02-21 14:19:41
uPT = 0
uPT.DT = 2020-02-21 14:19:41
cur = 1842
in_cur = 921
out_cur = 921
sbc operator id 0>
```

7 Описание журналов

7.1 Журнал CDR

CDR — это запись, содержащая сведения о работе оборудования.

В течение всего времени работы SBC на носитель данных пишется различная информация, в основном, это файлы журналов. Чтобы не допустить переполнения дискового пространства, на оборудовании SBC в фоновом режиме работает утилита очистки. Эта программа удаляет файлы, если время их существования превысило допустимое значение.

В связи с этим информация о вызовах пишется не в один большой файл, а разбивается на несколько. Таким образом, файлы, существующие дольше необходимого, могут быть удалены утилитой очистки, не повлияв на важные данные.

Правила, по которым формируется имя файла и длительность записи информации в него, определяются в файле конфигурации /usr/protei/Protei-SBC/SBC/config/Trace.cfg.

Внимание! Конфигурационный файл Trace.cfg редактировать не рекомендуется.

На момент поставки в SBC определен следующий формат журнала для записи CDR файлов (Trace.cfg):

```
sbc_cdr = {
    file = "cdr/%Y_%m_%d_%H_%M_%S.cdr";
    period = "1hour";
    type = "cdr";
    level=10;
    local_level=10;
};
```

В данном формате присутствуют следующие параметры:

1. File — правило формирования имени, # отмечаются числовые значения параметра.

2. Period — период времени, в течении которого в CDR файл будет производиться запись информации о поступающих событиях, по истечении этого

времени текущий CDR файл закрывается и открывается следующий CDR файл. Формат записи:

– #day — создание нового CDR файла каждый # день, в 00:00:00;

– #hour — создание нового CDR файла каждый # час, в nn:00:00.

nn = (# + 1) mod 23, mod — остаток от деления.

Пример: 1hour — длительность записи в файл 1 час.

3. type — тип журнала, для CDR файлов имеет значение cdr.

4. level — уровень детализации вывода информации. Возможные значения:

– 0 — файл не пишется;

– 1 10 — файл пишется.

Значение по умолчанию — 1.

5. local_level – уровень детализации вывода информации в локальный файл. Возможные значения:

– 0 — локальная запись отключена;

– 1 10 — файл пишется.

Значение по умолчанию — 1.

Директория, где будут располагаться CDR-файлы, создается в общем разделе для журналов. Имя и расположение общего раздела определяются в глобальных параметрах файла Trace.cfg. Как правило, общий раздел именуется logs.

Имя CDR файла формируется динамически согласно формату в конфигурационном файле Trace.cfg. В таблице 35 приведено описание подставляемых полей.

Таблица 35 — Описания подставляемых полей в имени CDR файла

Имя поля	**Описание**	**Примечание**
%Y	год	подставляемое поле — текущий год
%m	месяц	подставляемое поле — текущий месяц
%d	день	подставляемое поле — текущий день
%H	час	подставляемое поле — текущий час
%M	минута	подставляемое поле — текущая минута
%S	секунда	подставляемое поле — текущая секунда

Символы «.CDR» в примере — произвольное дополнение к имени, задаваемое пользователем.

Подставляемые поля в имени файла могут располагаться в любом месте имени в любом сочетании с произвольными символами, в произвольном порядке.

В итоге, исходя из данных, представленных в примере, будет создан файл с именем, состоящим из даты-времени на момент создания и постоянного значения (в примере — это строка «.CDR»), например, 2015_12_14_16_31_01.CDR.

Таким образом, исходя из примера блока конфигурации CDR файла, приведенного в начале раздела, будет создана папка с именем CDR, где будут формироваться CDR файлы через каждый час.

Параметры, отображающие данные типа «Дата и время …», имеют формат: #Year #Mon #Day #Hour:#Min:#Sec.#Ms.

1. #Year — год, г., записан четырьмя цифрами.
2. #Mon — месяц, мес., записан двумя цифрами.
3. #Day — день, д., записан двумя цифрами.
4. #Hour — час, ч, записан двумя цифрами.
5. #Min — минута, мин, записана двумя цифрами.
6. #Sec — секунда, с, записана двумя цифрами.
7. #Ms — миллисекунда, мс, записана тремя цифрами.

7.1.1 Формат CDR-файлов

CDR файл — это набор структурированных строк записей.

Все записи имеют одинаковый формат, независимо от события, по причине которого запись была сформирована. Символ разделитель полей записи — точка с запятой «;».

Пример:

```
2020 02 27 21:00:00.540;Ngn.SBC.Op.0;SipCall.267;I.Ngn.SBC.Op.0.Route.2.SipCall.267.0009700943;53021027729937128;53021027729937129;53021027729937130;5E5803611C629000B4B40_192.168.99.151:5061;5E5803786243500909D84_192.168.110.154:5070;78128220454;78128223545;78128220454;78128223545;2020_02_27_20:59:20.401;2020_02_27
```

20:59:20.461;40.055;I;1;16;192.168.99.151:23194/0/0/2002/2000/0/59;192.168.110.155:33510/0/0/2000/2003/0/57;;;0;0;0;Ngn.SBC.Op.0.Route.2;Ngn.SBC.SP.0;BYE;40.116

В таблице 36 приведено описание параметров CDR файла.

Таблица 36 — Параметры CDR файла

Имя поля	Описание	Пример
Release_DT	Дата и время окончания вызова	2020-02-27 21:00:00.540
Op_CA	Компонентный адрес оператора. Формат: Ngn.SBC.Op.#	Ngn.SBC.Op.0
Call_CA	Компонентный адрес логики, обработавшей вызов. Формат: SipCall.#	SipCall.267;
CallUniqueID	Уникальный идентификатор вызова на SBC верхнего уровня. Формат: $CallType$.$CallSource$.SipCall.#.#	I.Ngn.SBC.Op.0.Route.2. SipCall.267.0009700943;
GUID.CallID	Уникальный идентификатор вызова на SBC нижнего уровня	53021027729937128
GUID.CallLegID_A	Уникальный идентификатор абонента А	53021027729937129
GUID.CallLegID_B	Уникальный идентификатор абонента В	53021027729937130
CallID_A	SIP CallID для плеча А	5E5803611C629000B4B40 _192.168.99.151:5061
CallID_B	SIP CallID для плеча В	5E5803786243500909D84 _192.168.110.154:5070
CdPN_A	Номер вызывающей стороны в плече А	78128220454
CgPN_A	Номер вызываемой стороны в плече А	78128223545
CdPN_B	Номер вызывающей стороны в плече В	78128220454
CgPN_B	Номер вызываемой стороны в плече В	78128223545
Start_DT	Дата и время начала вызова	2020-02-27 20:59:20.401
Answer_DT	Дата и время ответа	2020-02-27 20:59:20.461
SpeechDuration	Длительность разговора, в секундах	40.055
CallType	Тип вызова. Возможные варианты: – I —входящий, от оператора;	I

Имя поля	Описание	Пример
	– O — исходящий, от сервисной платформы	
ReleaseSource	Источник отбоя. Возможные варианты: – 0 — SBC; – 1 — плечо A; – 2 — плечо B	1
Q850Cause	Причина отбоя согласно Q.850.	16
RTPStat_A	Статистика RTP-соединения плеча A. Содержит ряд параметров. См. п. 7.1.2 «**Ошибка! Источник ссылки не найден.**»	192.168.99.151:23194 /0/0/2002/2000/0/59
RTPStat_B	Статистика RTP-соединения плеча B. Содержит ряд параметров. См. п. 7.1.2 «**Ошибка! Источник ссылки не найден.**»	192.168.110.155:33510 /0/0/2000/2003/0/57
NumRoutingTries	Количество попыток маршрутизации	
RoutingHistory	История маршрутизации вызова	
UnauthSource	Флаг, была ли авария «неразрешенный источник медиа»	0
UnauthPT	Флаг, была ли авария «некорректный тип данных payload type»	0
OverBand	Флаг, была ли авария «превышение полосы пропускания»	0
CgP_CA	Компонентный адрес источника вызова. Формат: Ngn.SBC.Op.#.Route.# или Ngn.SBC.SP.#	Ngn.SBC.Op.0.Route.2
CdP_CA	Компонентный адрес последнего получателя вызова	Ngn.SBC.SP.0
SipReleaseCause	SIP-причина отбоя. Возможные значения: – CANCEL; – BYE; – NO_RESP; – UNKNOWN.	BYE

Имя поля	Описание	Пример
	#cause# в диапазоне 400-699	
SessionDuration	Длительность всего соединения, с момента отправления первого INVITE до разрыва соединения	40.116

Примечание: если вызываемая сторона не ответила на вызов, то поле Answer_DT принимает значение 0000 00 00 00:00:00.000.

7.1.2 RTP-статистика в плече

Формат записи статистики в плечах A и B, символ-разделитель — слэш «/»:

1. Удаленный HostPort для RTP.
2. Тип данных rx кодек.
3. Тип данных tx кодек.
4. Количество переданных пакетов.
5. Количество принятых пакетов.
6. Количество потерянных пакетов.
7. Дисперсия, среднеквадратичное отклонение времени получения пакета от математического ожидания.

Пример:

192.168.99.151:23194/0/0/2002/2000/0/59

В таблице 37 приведено описание параметров CDR файла.

Таблица 37 — Параметры CDR файла

Имя поля	Описание	Пример
ip:port	IP-адрес и порт, обрабатывающие RTP-пакеты	192.168.99.151: 23194
rxPayloadType	Тип данных принимающего кодека, payload type	0
txPayloadType	Тип данных передающего кодека, payload type	0
SentPackages	Количество отправленных пакетов	2002
RecvPackages	Количество принятых пакетов	2000
LostPackages	Количество потерянных и не обработанных пакетов	0

Имя поля	Описание	Пример
Variance	Среднеквадратичное отклонение времени получения пакета от математического ожидания	59

7.2 Диагностика

В журнал диагностики делается запись по причине отбоя (разрушения) вызова в системе.

Правило формирования имени файла и длительность записи информации в него определяются в файле конфигурации /usr/protei/Protei SBC/SBC/config/Trace.cfg.

Внимание! Конфигурационный файл Trace.cfg редактировать не рекомендуется.

На момент поставки в SBC определен следующий формат журнала диагностики (Trace.cfg):

```
sbc_diagnostic = {
   file=diagnostic.log;
   level=10;
   local_level=10;
};
sbc_diagnostic_warning = {
   file=diagnostic.log;
   level=10;
   local_level=10;
};
```

В данном примере присутствуют следующие параметры:

1. File — правило формирования имени файла (символом «%» отмечаются подставляемые поля, в данные поля вместо символов «%» и следующего за ним символа подставляется значение соответствующего параметра).

2. Level — уровень детализации вывода информации. Возможные значения:

– 0 — файл не пишется;

– 1 10 — файл пишется.

Значение по умолчанию — 1.

3. Local_level — уровень детализации вывода информации в локальный файл. Возможные значения:

– 0 — локальная запись отключена;

– 1 10 — файл пишется.

Значение по умолчанию — 1.

Директория, где будут располагаться файлы диагностики, создается в общем разделе для журналов. Имя и расположение общего раздела определяются в глобальных параметрах файла Trace.cfg. Как правило, общий раздел именуется logs.

7.2.1 Формат файла диагностики

Все записи имеют одинаковый формат, независимо от события, по причине которого запись была сформирована. Символ разделитель полей записи — точка с запятой «;».

Пример:

```
2018 05 18 16:32:35.127;53012159799754752;53012159799754753;+74957000000;+74957190000;SBC;0;Ngn.SBC.Route;1;Can't find operator;4;282;1 18679@10.10.0.4;;DefaultOperator;
```

В таблице 38 описаны параметры журнала диагностики sbc_diagnostic.

Таблица 38 — Параметры журнала sbc_diagnostic

Имя поля	Описание	Пример
Release_DT	Дата и время окончания вызова	2018-05-18 16:32:35.127
GUID.CallID	Уникальный идентификатор вызова на SBC нижнего уровня	53012159799754752
GUID.CallLegID	Уникальный идентификатор вызывающего	53012159799754753
CdPN	Номер вызывающей стороны	74957000000
CgPN	Номер вызываемой стороны	74957190000
Subsystem	Подсистема SBC, в которой возникла ошибка. Возможные варианты: – SIP — соответствует разделу sip; – SBC — соответствует разделу sbc	SBC
ReleaseSource	Источник отбоя. Возможные варианты: – 0 — SBC; – 1 — плечо A; – 2 — плечо B	0

Имя поля	Описание	Пример
Call_CA	Компонентный адрес логики, обработавшей вызов	Ngn.SBC.Route
CauseCode	Номер причины отбоя. См. п. 7.2.2 «**Ошибка! Источник ссылки не найден.**»	1
DiagLine	Строка диагностики с описанием ошибки. См. п. 7.2.2 «**Ошибка! Источник ссылки не найден.**»	Can’t find operator
ErrorFile	Код файла, в котором возникла ошибка. Возможные значения: – 4 — SipCall.cpp; – 5 — SbcComponent.cpp	4
ErrorLine	Номер строки в коде, где произошла ошибка	282
CallID_A	SIP CallID в плече A	1-18679@10.10.0.4
CallID_B	SIP CallID в плече B	
GateID	Имя локального шлюза SBC в плече A	DefaultOperator

7.2.2 Коды причин и строки диагностики

В таблице 39 приведены коды причин CauseCode со строками диагностики DiagLine, а также краткое описание. См. п. 7.2.1 «Формат файла диагностики».

Таблица 39 — Коды причин и строки диагностики

Код	Строка диагностики	Описание
1	Can't find operator	Не удалось найти подходящего оператора
2	Operator blocked	Оператор заблокирован
4	Max number of calls	Превышено максимальное число одновременных вызово
5	Empty SIP message pointer	Не заполнены поля SIP-сообщения.
6	ALG reject this call	Подсистема ALG отказалась принимать вызов
7	MCU reject this call	Подсистема MCU отказалась принимать вызов
8	Administrative release	Вызов прекращен по причине изменения состояния оператора или MCU
9	Can't find internal gate	Шлюз сервисной платформы отсутствует или временно недоступен
10	Op.Route blocked	Маршрут оператора заблокирован
16	No free UA handler	Недостаточно свободных логик. Требуется увеличить количество SIP_INIT_Handlers
18	Exceeded CPS	Превышено максимальное значение cps

Код	Строка диагностики	Описание
19	Can’t fill call context.	Контекст вызова не заполнен
20	Can’t resolve call source	Не удалось определить источник вызова
21	Can’t resolve call destination	Не удалось определить получателя
22	Can't register call	Не удалось зарегистрировать вызов
23	Can't get gate	Не удалось выбрать свободный шлюз сервисной платформы
24	Can’t set call type	Не удалось установить тип вызова. Вызов не попал ни в одну из указанных категорий: – Op→Op, вызов от оператора к оператору; – Op→SP, вызов от оператора к платформе; – SP→Op, вызов от платформы к оператору; – SP→SP, вызов от платформы к платформе
25	Routing Failure	Ошибка маршрутизации
26	Can’t find service platform	Не удалось найти сервисную платформу
27	Unsupported call type	Во время внешней маршрутизации используется не поддерживаемый тип вызова
28	Op.Route is unavailable	Маршрут оператора не доступен
29	Reached max call duration	Продолжительность вызова превысила установленные ограничения
30	Remote side doesn't support required PRACK	Одна из сторон требует поддержку PRACK, но вторая сторона не может это выполнить
31	No media from one side	В течение 30 секунд одна из сторон не отправила ни одного RTP-пакета

8 Приложение

В приложении содержится следующая информация:

1. Алгоритмы маршрутизации.
2. Типовые задачи конфигурирования.

8.1 Алгоритм внутренней маршрутизации

В общем случае алгоритм можно разбить на 3 фазы:

1. Определение источника вызова.
2. Маршрутизация.
3. Определение получателя вызова.

На рисунке 3 показаны фазы маршрутизации.

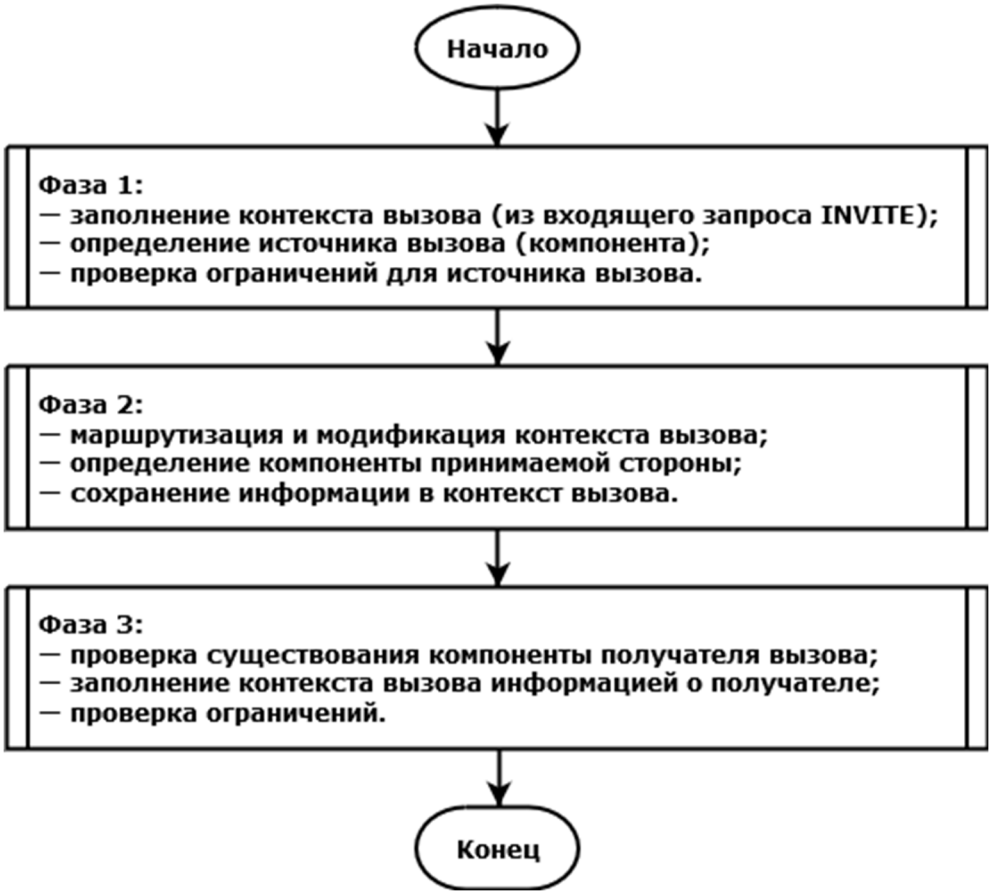

Рисунок 3 — Фазы маршрутизации

При запросе на «отбой» вызова от стороны B до получения 200 OK (от стороны B) используется алгоритм перемаршрутизации.

8.1.1 Фаза 1: Определение источника вызова

На рисунке 4 показана первая фаза алгоритма маршрутизации.

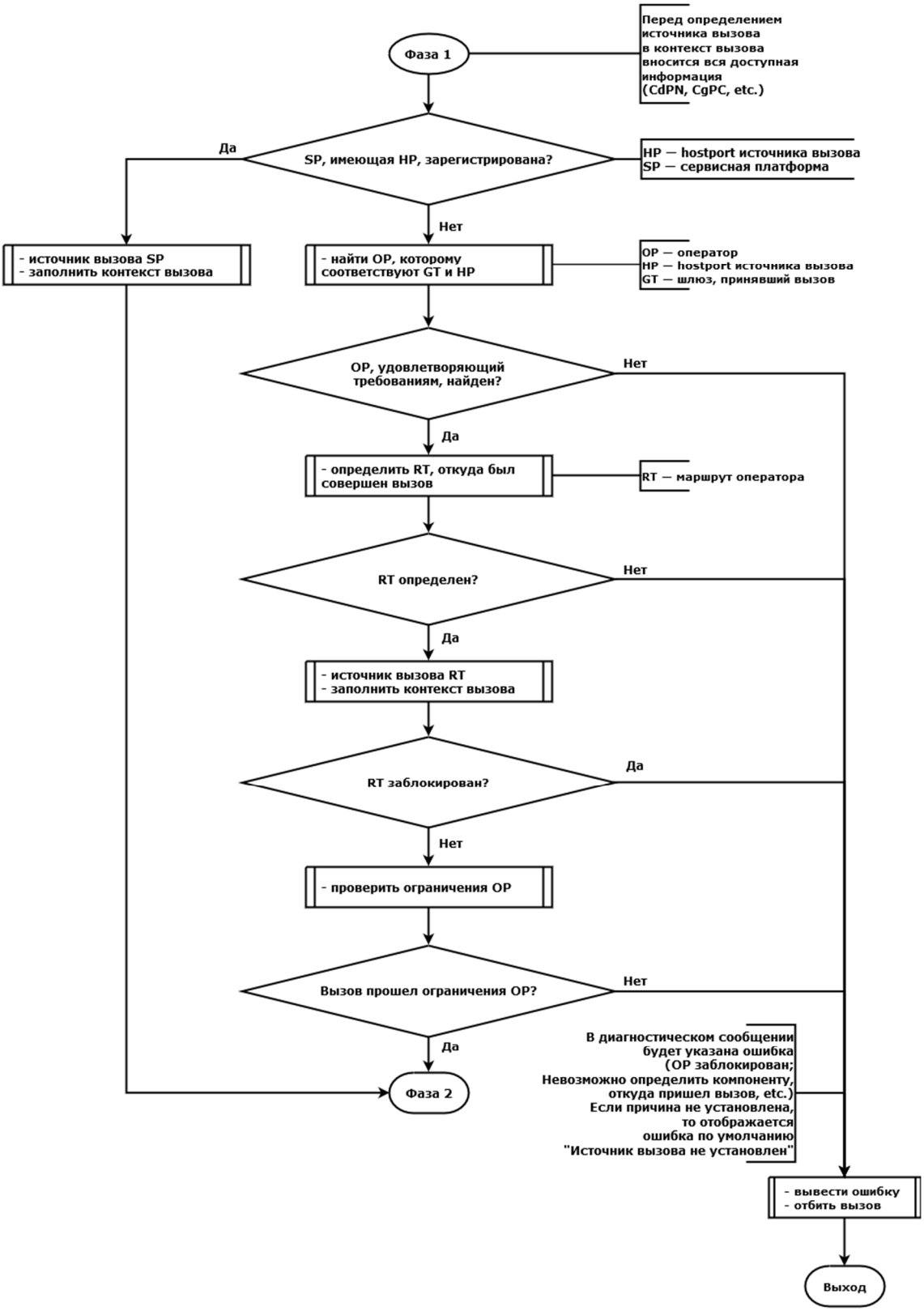

Рисунок 4 — Алгоритм маршрутизации. Фаза 1

8.1.2 Фаза 2: Маршрутизация и модификация контекста

На рисунке 5 показана вторая фаза алгоритма маршрутизации.

Рисунок 5 — Алгоритм маршрутизации. Фаза 2

8.1.3 Фаза 3: Определение получателя вызова

На рисунке 6 показана третья фаза алгоритма маршрутизации.

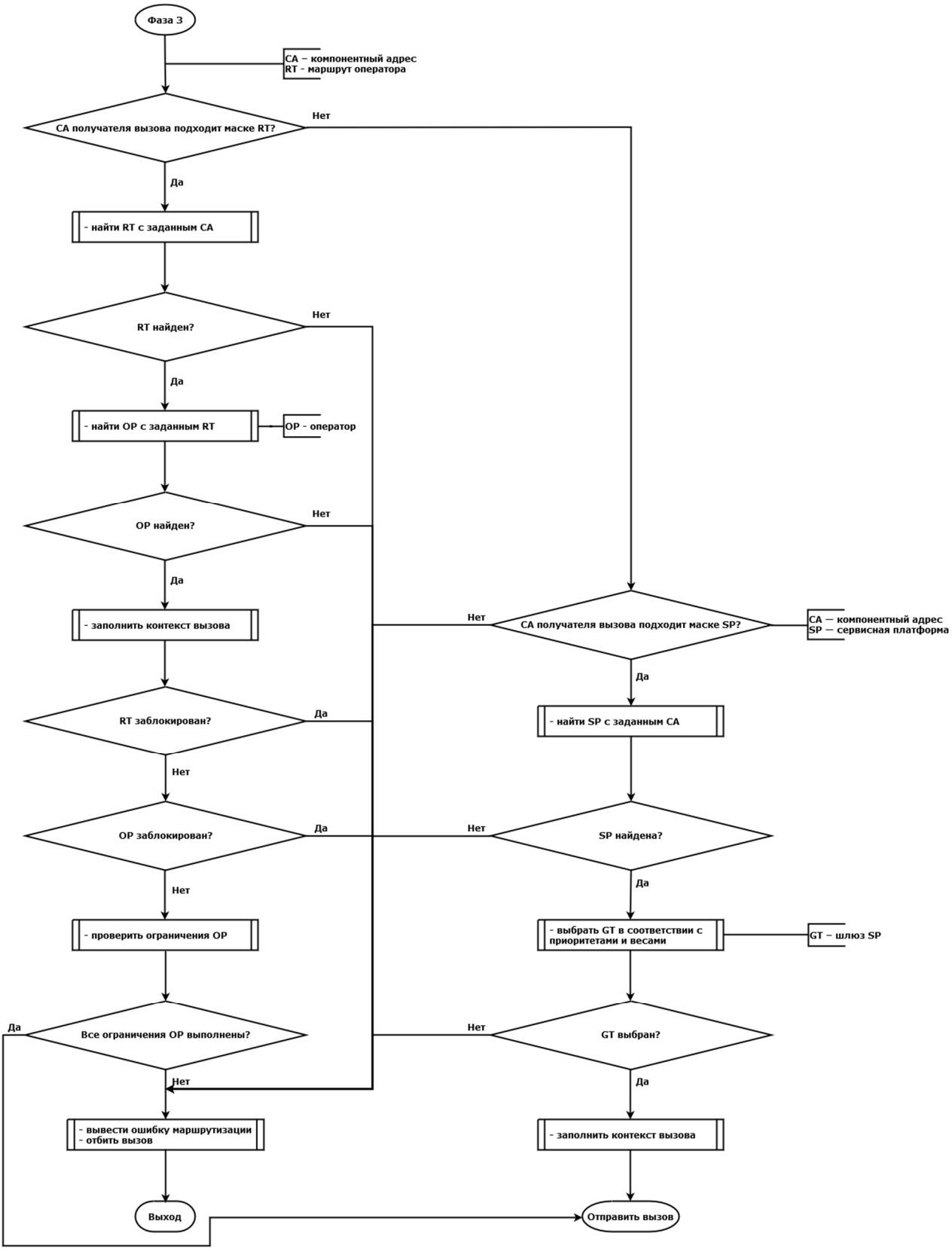

Рисунок 6 — Алгоритм маршрутизации. Фаза 3

8.1.4 Перемаршрутизация

На рисунке 7 показан алгоритм маршрутизации.

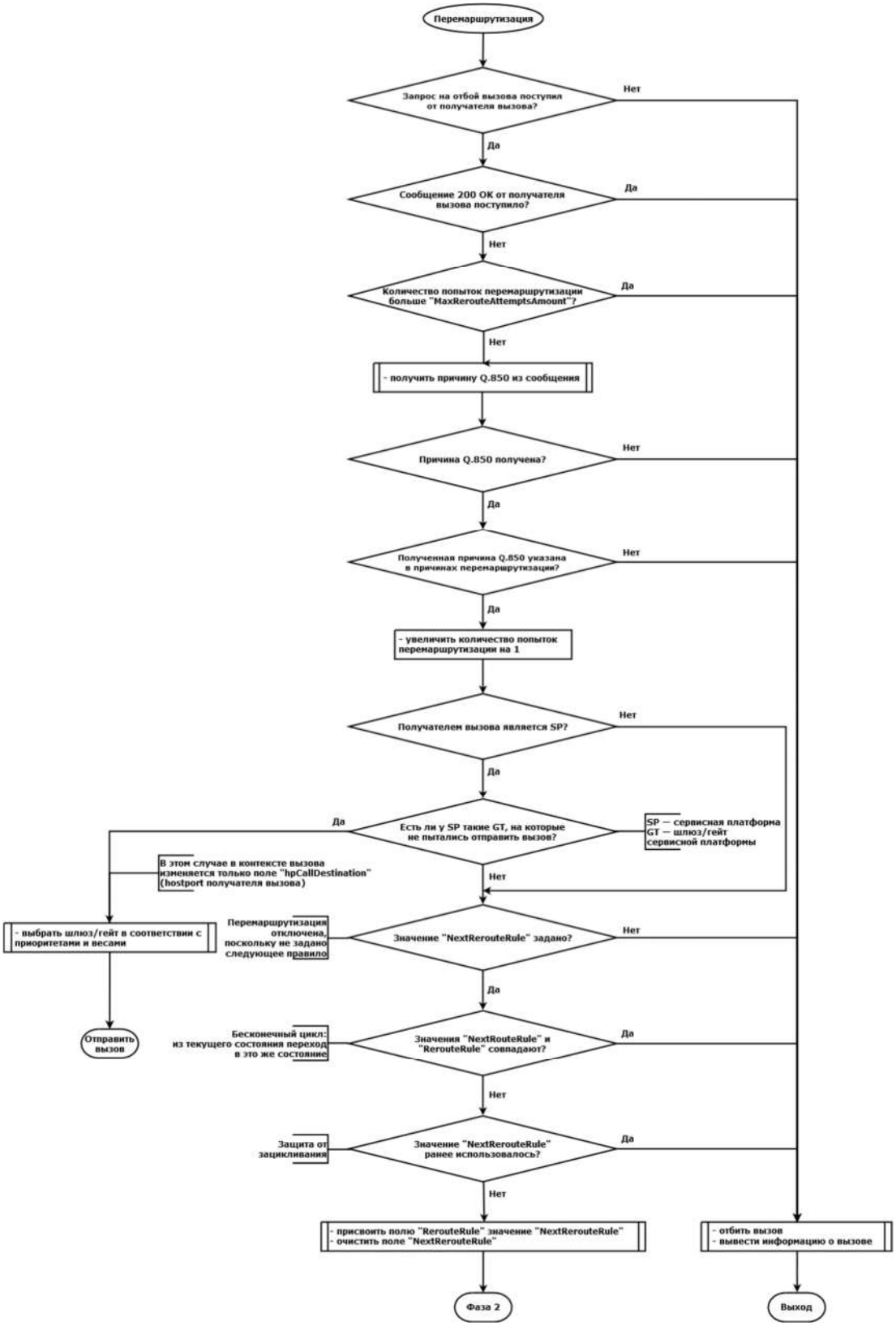

Рисунок 7 — Алгоритм маршрутизации

8.2 Типовые задачи конфигурирования I-SBC

Чтобы настроить базовый SIP-транка между сервисной платформой и оператором при посредничестве I-SBC, необходимо создать и сконфигурировать объекты:

1. Медиашлюзы (гейты).
2. MCU.
3. Сервисная платформа.
4. Оператор.
5. Правила маршрутизации.

8.2.1 Создать и настроить медиашлюз

Медиашлюзы являются необходимым элементом для всех остальных объектов, поэтому именно шлюзы конфигурируются в первую очередь.

Чтобы создать и настроить медиашлюз, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sip.

```
$username$:~$ ./cli
SBC> sip
```

2. Создать новый шлюз, добавив новый элемент в массив gate командой gate insert #. # — индекс элемента в массиве, тип — uint.

```
sip> gate insert 1
```

3. Задать значения параметрам name, ip, port следующими командами:

- name $newGate$: имя шлюза, тип — string, не более 7 символов;
- ip $ipAddr$: IP адрес шлюза, тип — ip;
- port $portLis$: прослушиваемый порт, тип — uint, в диапазоне 1024 65535.

```
sip gate 1> name OP1_SG
sip gate 1> ip 127.0.0.1
sip gate 1> port 5060
```

4. Применить эти настройки к текущей конфигурации командой commit.

```
sip gate 1> commit
```

Пример создания и конфигурирования медиашлюза:

```
SBC> sip gate insert 1
sip gate 1> name OP1-SG
sip gate 1> ip 127.0.0.1
sip gate 1> port 5060
sip gate 1> commit
```

8.2.2 Настроить MCU

Чтобы настроить MCU, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Создать подгруппу MCU командой mcu id #.

– # — номер подгруппы MCU, тип — uint.

```
sbc> mcu id 1
sbc mcu id 1>
```

3. Задать значения параметрам mcu ip, mcu port следующими командами:

– mcu ip $ipAddr$: IP адрес, который занимает MCU; тип — ip;

– mcu port $portLis$: прослушиваемый порт, тип — uint, в диапазоне 1024 65535.

```
sbc mcu id 1> mcu-ip 127.0.0.1
sbc mcu id 1> mcu-port 1960
```

4. Создать новую подсеть, добавив элемент в массив mcu subnet командой mcu subnet insert #.

– # — номер подсети, тип — uint.

```
sbc mcu id 1> mcu-subnet insert 1
sbc mcu mcu-subnet 1>
```

5. Задать значения параметрам audio count, video count, name, mcu media ip следующими командами:

– audio-count # : количество аудиоканалов, тип — uint;

– video-count # : количество видеоканалов, тип — uint;

– name $subName$: имя подсети, тип — string;

– mcu-media-ip $ipAddr$: IP адрес для медиатрафика, тип — ip.

```
sbc mcu mcu-subnet 1> audio-count 50
sbc mcu mcu-subnet 1> video-count 50
sbc mcu mcu-subnet 1> name IntMCU
sbc mcu mcu-subnet 1> mcu-media-ip 192.168.3.124
```

6. Применить эти настройки к текущей конфигурации командой commit.

```
sbc mcu mcu-subnet 1> commit
```

Пример создания и конфигурирования MCU:

```
SBC> sbc
sbc> mcu id 1
sbc mcu id 1> mcu-ip 127.0.0.1
sbc mcu id 1> mcu-port 1960
sbc mcu id 1> mcu-subnet insert 1
sbc mcu mcu-subnet 1> audio-count 50
sbc mcu mcu-subnet 1> video-count 50
sbc mcu mcu-subnet 1> name IntMCU
sbc mcu mcu-subnet 1> mcu-media-ip 192.168.3.124
sbc mcu mcu-subnet 1> commit
```

8.2.3 Создать и настроить сервисную платформу

Чтобы создать и настроить сервисную платформу, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Создать платформу командой service-platform id #.

– # — номер сервисной платформы, тип — uint.

```
sbc> service-platform id 1
sbc service-platform id 1>
```

3. Создать новый медиашлюз, добавив элемент в массив gates командой gates insert #.

– # — номер шлюза, тип — uint.

```
sbc service-platform id 1> gates insert 1
sbc service-platform gates 1>
```

4. Задать значения параметрам internal host, priority, weight, mcu network следующими командами:

- internal host $ipAddr$: IP адрес шлюза, тип — ip;
- priority # : приоритет шлюза, тип — uint, в диапазоне 0 100;
- weight # : вес шлюза, тип — uint, в диапазоне 0 100;
- mcu-network $subName$: имя подсети, тип — string.

```
sbc service-platform gates 1> internal-host 185.253.14.12:5060
sbc service-platform gates 1> priority 1
sbc service-platform gates 1> weight 50
```

5. Применить эти настройки к текущей конфигурации командой commit.

```
sbc service-platform gates 1> commit
```

Пример создания и конфигурирования сервисной платформы с одним шлюзом:

```
SBC> sbc
sbc> service-platform id 1
sbc service-platform id 1> gates insert 1
sbc service-platform gates 1> internal-host 185.253.14.12:5060
sbc service-platform gates 1> priority 1
sbc service-platform gates 1> weight 50
sbc service-platform gates 1> commit
```

8.2.4 Создать и настроить оператора

Чтобы создать оператора и настроить его, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Создать оператора командой operator id #.

- # — номер оператора, тип — uint.

```
sbc> operator id 1
sbc operator id 1>
```

3. Задать значения параметрам mcu network, sp number следующими командами:

– mcu-network $subName$: имя подсети, тип — string;

– sp number # : номер сервисной площадки, тип — uint.

```
sbc operator id 1> mcu-network IntMCU
sbc operator id 1> sp-number 1
```

4. Создать новое правило маршрутизации, добавив элемент в массив route командой route rule #.

– # — номер правила, тип — uint.

```
sbc operator id 1> route rule 1
sbc operator id 1 route id 1>
```

5. Задать значения параметрам external host, external gate, internal gate следующими командами:

– external-host $HostPort$: адрес внешней подсети, тип — ip:port;

– external-gate $extName$: имя шлюза для исходящих, тип — string;

– internal-gate $intName$: имя шлюза для входящих, тип — string.

```
sbc operator id 1 route id 1> external-gate OP1-SG
sbc operator id 1 route id 1> internal-gate SP1-SG
sbc operator id 1 route id 1> external-host 65.34.51.67:5060
```

6. Применить эти настройки к текущей конфигурации командой commit.

```
sbc operator id 1 route id 1> commit
```

Таким образом обеспечивается фиксированный SIP-транк при установлении соединения между оператором и сервисной платформой.

Пример создания и конфигурирования оператора с одним правилом маршрутизации:

```
SBC> sbc
sbc> operator id 1
sbc operator id 1> mcu-network IntMCU
sbc operator id 1> sp-number 1
sbc operator id 1> route rule 1
sbc operator id 1 route id 1> external-gate OP1-SG
sbc operator id 1 route id 1> internal-gate SP1-SG
```

```
sbc operator id 1 route id 1> external-host 65.34.51.67:5060
sbc operator id 1 route id 1> commit
```

8.2.5 Настроить кодеки в медиапрофиле

Медиапрофиль позволяет настраивать четыре списка аудио и видеокодеков: Prohibited, Allowed, Supported, Mandatory. Для конфигурирования медиапрофиля необходимо создать кодеки в системе, а затем расположить в соответствующих списках.

Примечание: приведенные инструкции описывают работу с аудиокодеком. Взаимодействия с видеокодеком полностью аналогичны с точностью до замены audio на video.

Создать кодек.

Чтобы создать кодек, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Создать кодек командой audio-codec id #.

– # — номер кодека, тип — uint.

```
sbc> audio-codec id 1
sbc audio-codec id 1>
```

3. Задать значение параметру codec командой codec $offName$.

– $offName$ — официальное название кодека, тип — string.

```
sbc audio-codec id 1> codec PCMA
```

Пример создания аудиокодека:

```
SBC> sbc
sbc> audio-codec id 1
sbc audio-codec id 1> codec PCMA
```

Создать и настроить медиапрофиль.

Примечание: в настройке используется кодек PCMA из раздела 8.2.5.1.

Чтобы настроить медиапрофиль, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Создать медиапрофиль командой media-profile id #.

– # — номер медиапрофиля, тип — uint.

```
sbc> media-profile id 1
sbc media-profile id 1>
```

3. Указать тип кодека с помощью команд audio и video.

```
sbc media-profile id 1> audio
sbc media-profile audio>
```

4. Добавить кодек в нужный список командой $listType$ insert #list #audio.

– $listType$ — название списка из упомянутых в начале раздела, тип — string;

– #list — номер кодека в списке $listType$, тип — uint;

– #audio — номер кодека в списке кодеков sbc audio-codec, тип — uint.

```
sbc media-profile audio> prohibited insert 1 1
```

5. Применить эти настройки к текущей конфигурации командой commit.

```
prohibited> commit
```

Пример создания и настройки медиапрофиля:

```
SBC> sbc
sbc> media-profile id 1
sbc media-profile id 1> audio
sbc media-profile audio> prohibited insert 1 1
prohibited> commit
```

8.2.6 Переадресовать вызовы к сервисной платформе

Если вызов исходит от оператора к сервисной платформе, то для случаев, когда основной шлюз недоступен, предусмотрена возможность перенаправления вызова на другие шлюзы.

Чтобы задать настройки перемаршрутизации, необходимо выполнить следующие действия:

1. Запустить командную строку cli и перейти в секцию sbc.

```
$username$:~$ ./cli
SBC> sbc
```

2. Зайти в общие параметры common раздела sbc командой sbc common.

```
sbc> common
sbc common>
```

3. Задать значения параметрам rerouting cause, reroute attempts следующими командами:

– reroute attempts #, количество попыток, тип — uint;

– rerouting cause #list #Q850#, причины для осуществления перемаршрутизации:

– #list — номер причины в списке, тип — uint;

– #Q850# — номер причины в спецификации Q.850, тип — uint.

```
sbc common> reroute-attempts 3
sbc common> rerouting cause 0 18
```

4. Применить эти настройки к текущей конфигурации командой commit.

```
sbc common> commit
```

Пример настройки перемаршрутизации:

```
SBC> sbc common
sbc common>
sbc common> reroute-attempts 3
sbc common> rerouting-cause 0 18
sbc common> commit
```

Лист регистрации изменений

Изм	Номера листов (страниц)				Всего листов (страниц) в док.	№ документа	Входящий № сопроводительного докум. И дата	Подпись	Дата
	Заме-ненных	Изме-ненных	Новых	Анну-лиро-ванных					

Изм.	Лист	№ докум.	Подпись	Дата