Мультипротокольные маршрутизаторы NSG

Программное обеспечение NSG Linux

Руководство пользователя

Часть 4

Маршрутизация и службы IP

Версия программного обеспечения 1.0 build 6

Обновлено 31.10.2011

Москва 2011

АННОТАЦИЯ

Данный документ содержит руководство по настройке и применению мультипротокольных маршрутизаторов NSG, оснащенных программным обеспечением NSG Linux. Руководства по применению других продуктов NSG, а также базового программного обеспечения NSG для серий NPS–7e, NSG–500, NX–300 и NSG–800 содержатся в отдельных документах.

Документ состоит из следующих разделов:

Часть 1. Общесистемная конфигурация.
Часть 2. Физические порты.
Часть 3. Протоколы канального уровня. Коммутация пакетов.
Часть 4. Маршрутизация и службы IP.
Часть 5. Туннелирование и виртуальные частные сети (VPN).
Часть 6. Основные команды и утилиты NSG Linux.

Четвёртая часть документа посвящена настройке IP-маршрутизации, а также современных механизмов управления трафиком и обеспечения гарантированного качества услуг (QoS). Также рассматриваются прикладные службы IP, предназначенные для автоматической настройки сети.

Общее описание системы, описание общесистемных параметров и командного языка системы приведены в Части 1. Настройка физических интерфейсов различного типа представлена в Части 2. Настройка протоколов канального уровня (в т.ч. VLAN, организация сеансового доступа средствами PPP и доступа к асинхронным портам средствами Reverse Telnet), коммутация пакетов на втором уровне (Ethernet bridging, Frame Relay) и коммутация пакетов X.25 рассмотрены в Части 3. Часть 5 посвящена построению туннелей и виртуальных частных сетей различных типов. В Части 6 изложены начала работы с ОС Linux в объёме, желательном для администрирования и отладки сетей на основе оборудования NSG с использованием расширенных возможностей системы.

ВНИМАНИЕ Продукция компании непрерывно совершенствуется, в связи с чем возможны изменения отдельных аппаратных и программных характеристик по сравнению с настоящим описанием. Сведения о последних изменениях приведены в файлах README.TXT, CHANGES, а также в документации на отдельные устройства.

Замечания и комментарии по документации NSG принимаются по адресу: doc@nsg.net.ru.

© ООО «Эн–Эс–Джи» 2003–2011

ООО «Эн–Эс–Джи»
Россия 105187 Москва
ул. Вольная, д.35
Тел./факс: (+7–495) 727–19–59 (многоканальный)

http://www.nsg.ru/
mailto:info@nsg.net.ru
mailto:sales@nsg.net.ru
mailto:support@nsg.net.ru

§ СОДЕРЖАНИЕ §

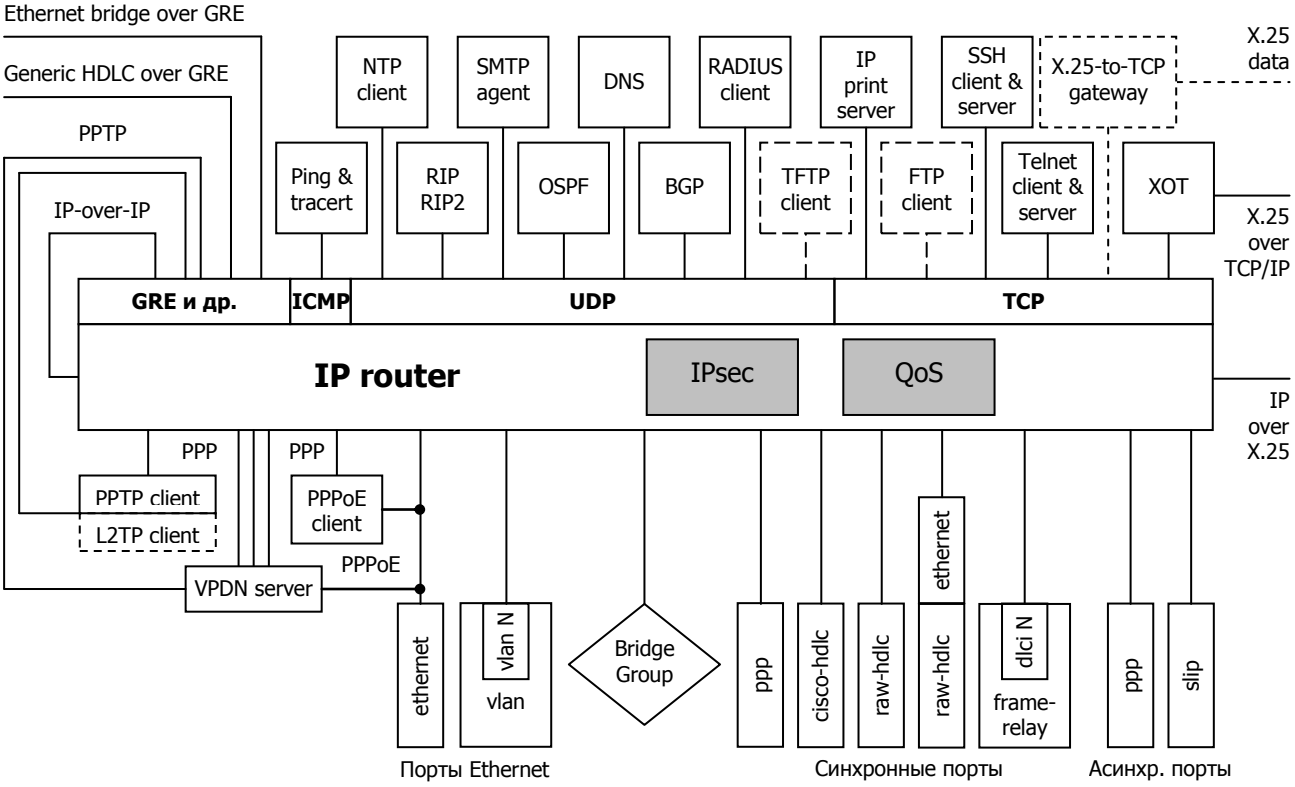
Часть 4. Маршрутизация и службы IP

§4.1. IP-интерфейсы

§4.1.1. Объекты IP-маршрутизации

IP-маршрутизатор в узком смысле слова — это программный модуль, осуществляющий передачу IP-пакетов на третьем уровне протокольной архитектуры (уровне межсетевого взаимодействия). Пакеты IP могут входить в маршрутизатор и выходить из него через следующие типы объектов:

— Физические и виртуальные синхронные порты с инкапсуляцией cisco-hdlc, ppp или raw-hdlc, обозначаемые как sN.0, xN.0 или tN.0, в зависимости от типа физического интерфейса или туннеля.
— Виртуальные каналы (DLC) портов Frame Relay, обозначаемые как sN.<dlci>, xN.<dlci> или tN.<dlci>, соответственно, где <dlci> — номер DLCI.
— Физические порты Ethernet с инкапсуляцией ethernet, обозначаемые как eth0 или sN, соответственно.
— Физические и виртуальные синхронные порты с инкапсуляцией Ethernet-over-HDLC, обозначаемые как sN.0, xN.0 или tN.0, в зависимости от типа физического интерфейса или туннеля.
— Виртуальные локальные сети (VLAN), находящиеся на физических портах Ethernet с инкапсуляцией vlan. Обозначаются как eth0.<vlan> или sN.<vlan>, соответственно, где <vlan> — номер VLAN.
— Физические асинхронные порты с инкапсуляцией PPP и SLIP, обозначаемые s0 ... s1 (универсальные порты) либо a1 ... a16 (встроенные асинхронные порты).
— Туннели IP-over-IP (GRE), обозначаемые как tuniN, где N — номер туннеля, указанный при его создании.
— Туннели IP-over-X.25, обозначаемые как tunxN.
— Мосты (*bridge groups*), обозначаемые как brN.
— Интерфейсы клиента PPPoE. Обозначаются как eth0.0, sN.0, eth0.<vlan>.0 или sN.<vlan>.0 в зависимости от имени физического порта Ethernet или VLAN, на которых работает данный клиент.
— Интерфейсы клиента PPTP. Обозначаются как pptpN.
— Интерфейсы сервера виртуальных сетей коммутируемого доступа (VPDN). Обозначаются как pppNNN.



IP-маршрутизатор и связанные с ним объекты канального и прикладного уровней
(компоненты, показанные длинным пунктиром, доступны только средствами командной оболочки Linux, коротким пунктиром — находятся в разработке)

Строго говоря, физические порты, виртуальные порты (DLC, VLAN) и туннели подключаются к маршрутизатору через соответствующие IP-интерфейсы или суб-интерфейсы; именно к ним относятся вышеприведенные символические имена. Однако поскольку IP-интерфейс скрыт внутри структуры порта и однозначно привязан к нижележащим объектам протокольной иерархии, такая детализация не представляет практического интереса. *В дальнейшем под IP-интерфейсами будут пониматься все вышеперечисленные объекты.*

IP-интерфейсы VPDN (в данной версии это только интерфейсы PPPoE) — динамические, они автоматически создаются системой при установлении соединения и удаляются при его разрыве. Для создания интерфейса VPDN, а также для интерфейсов Async PPP и SLIP используется заранее заданный *шаблон* (*virtual template*), в котором содержится расширенная информация о свойствах IP-интерфейсов (IP-адреса, описание процедуры аутентификации и т.п.). Остальные типы IP-интерфейсов являются статическими, т.е. создаются и удаляются администратором вручную в процессе конфигурирования устройства.

Подробное описание настройки объектов и служб канального уровня приведено в Части 3 данного руководства.

Со стороны четвертого (прикладного) уровня протокольной архитектуры к IP-маршрутизатору подключены многочисленные локальные приложения: обработчики протоколов маршрутизации, серверы и клиенты различных служб и др.

§4.1.2. Параметры IP-интерфейсов

Для IP-интерфейсов, представляющих собой синхронные физические порты, порты Ethernet, DLCI, VLAN, туннели IP-over-*xxx* и мосты, параметры IP назначаются с помощью следующих команд в меню данного объекта:

ip address <ip-префикс> [peer <ip-адрес>] [broadcast <ip-адрес>] [anycast <ip-адрес>]
no ip address <ip-префикс>
: Статическая установка и удаление параметров IP-интерфейса. Обязательным параметром является IP-префикс (адрес и длина маски, например, 123.134.145.156/24).
При необходимости можно отдельно назначить адреса удаленной стороны (peer, в команде show interface показывается как pointopoint) в соединении "точка-точка", широковещательной (broadcast) и групповой (anycast) рассылки.

ip address dhcp
no ip address dhcp
: Включение и выключение клиента DHCP для настройки ключевых параметров IP-интерфейса и маршрутизации.

mtu {<64...18000> | auto }
: Размер MTU, в байтах, для IP-пакетов, отсылаемых через данный интерфейс. При указании auto (значение по умолчанию) размер MTU устанавливается драйвером автоматически в зависимости от типа интерфейса.

Одному IP-интерфейсу может быть назначено одновременно несколько префиксов. В некоторых других системах такие IP-адреса называются *вторичными* (*aliases*). В данном случае, однако, этот термин не вполне точен, поскольку все IP-адреса, назначенные интерфейсу, полностью равноправны, среди них нет главного адреса и псевдонимов. Диапазоны IP-адресов, описываемые этими префиксами, могут перекрываться.

ПРИМЕЧАНИЕ Статическое и динамическое назначение IP-адресов не могут использоваться одновременно. При включении клиента DHCP удаляются все IP-адреса, ранее назначенные данному интерфейсу, и наоборот, при назначении статического IP-адреса выключается клиент DHCP.

ПРИМЕЧАНИЕ Если сервер DHCP сообщает адреса серверов DNS, то для того, чтобы устройство NSG использовало их, необходимо указать в настройках клиента/ретранслятора DNS параметр update-from с указанием на данный интерфейс (см. п.4.12.6).

Отдельный случай представляют собой физические асинхронные порты с инкапсуляцией PPP, SLIP, а также программные интерфейсы PPTP, PPPoE. В отличие от вышеперечисленных, эти типы интерфейсов ориентированы на сеансовые соединения. Для них предусмотрен специальный объект — *шаблон интерфейса* (*virtual template*), в котором указывается статический IP-адрес либо опция получения динамического IP-адреса от удаленной стороны при установлении соединения. Указатель на шаблон содержится в меню IP-интерфейса:

```
!
nsg
    virtual-template 1
        ip address 123.145.167.189
        peer ip address 123.145.167.190
        mtu 1492
        .....
        exit
    port a1
        virtual-template 1
        .....
        exit
    exit
!
```

Настройка шаблонов интерфейсов подробно рассмотрена в Части 3 данного руководства.

ПРИМЕЧАНИЕ Параметры IP-интерфейсов могут также настраиваться в меню конфигурации Zebra (config)# (см. п.4.1.6). При этом, если IP-адрес назначен каким-либо одним из двух вышеназваных способов, то и удалять/изменять его следует этим же способом. Рекомендуется изменять IP-адрес только в меню (config-nsg-xxx)#.

ip proxy-arp { enable | disable }
: Включить/выключить режим ARP Proxy на данном интерфейсе. Подробнее см. п.4.1.4.

ip netflow {<номер_шаблона> | disable }
: Включить сбор и отсылку статистики NetFlow. Подробно о мониторинге IP-трафика см. п.4.11.

[no] access-group ...
: Настройка фильтрации IP-пакетов на данном интерфейсе. Подробнее см. п.4.9.

[no] service-policy ...
: Выбор и настройка политики управления IP-трафиком для данного интерфейса. Подробно см. п.4.10.

nat ...
: Настройка трансляции сетевых IP-адресов (NAT) для данного интерфейса. Подробнее см. п.4.8.

[no] crypto ...
: Настройка защищенных туннелей VPN, создаваемых на данном IP-интерфейсе. Подробно о настройке VPN см. Часть 5.

state-script {<0...1000> | start }
: Указатель на номер сценария, вызываемого при возникновении какого-либо события на интерфейсе. Вместо номера сценария 0 может быть использовано ключевое слово start для обозначения скрипта, выполняемого при старте системы. Два или более сценариев могут быть подключены последовательно при помощи параметра next N.
Подробнее о сценариях Linux и NSG см. Часть 6.

Кроме того, в меню IP-интерфейсов имеются также команды:

adm-state { up | down }
: Административное состояние интерфейса. Данная команда отсутствует в меню интерфейсов VLAN и DLC, представляющих собой суб-интерфейсы портов Ethernet и Frame Relay, соответственно. В данной версии NSG Linux предусмотрено только административное включение/выключение порта целиком.

description "<комментарий>"
: Текстовое описание данного интерфейса для удобства администрирования. Если строка содержит пробелы, она должна быть заключена в кавычки. Максимальная длина описания — 255 символов.

show ...
: Просмотр состояния и статистики интерфейса. Подробно см. п.4.1.5.

§4.1.3. Соединения "точка-точка" и ненумерованные интерфейсы

Для соединений WAN "точка-точка" IP-адреса интерфейса и удаленной стороны могут быть заданы в формате с длиной маски 32 бит. Такой формат полностью определяет адреса в данном соединении и фиксирует тот факт, что других адресов (адреса сети, широковещательного адреса) в нем не имеется:

```
port s1 ip address 10.0.0.1/32 peer 10.0.0.2
```

При этом указанные два адреса, в общем случае, никак не связаны друг с другом. В частности, интерфейсу с длиной маски /32 может быть назначен IP-адрес, уже принадлежащий другому интерфейсу данного устройства. Фактически такой интерфейс является ненумерованным (*unnumbered*); адрес, "позаимствованный" у другого интерфейса, используется только в качестве исходного (*source address*) при отправке пакетов локальными службами устройства (ping, XOT и др.) через данный интерфейс.

§4.1.4. ARP Proxy

На любом из IP-интерфейсов устройства может быть включен режим ARP Proxy. В этом случае все другие интерфейсы устройства будут отвечать на запросы ARP с адресами из сети, подключенной к данному интерфейсу (т.е. подпадающими под его адрес и маску). Для этой цели в меню физического порта, Frame Relay DLCI или Ethernet VLAN — например, (config-port-sN)# — имеются команда:

ip proxy-arp { enable | disable }
: Включить/выключить режим ARP Proxy.

Пример. Устройство NSG работает в качестве псевдомоста между двумя сетями Ethernet. Запросы ARP из одной сети на адреса, расположенные в другой сети, разрешаются на MAC-адреса интерфейса NSG, расположенного в исходной сети. В результате пакеты, адресованные из одной сети в другую, попадают на устройство NSG и маршрутизируются им в другую сеть.

```
!
nsg
    card s1 im-et10
    port s1
        mac-address 00:09:56:11:22:33
        ip address 10.0.0.1/8
        ip proxy-arp enable
        exit
    port eth0
        ip address 192.168.0.1/24
        ip proxy-arp enable
        exit
    exit
!
```

§4.1.5. Просмотр состояния и статистики IP-интерфейсов

Команда отображения статистики имеется в меню большинства сетевых устройств (в терминах Linux), присутствующих в системе. Исключением в данной версии являются динамически создаваемые интерфейсы сервера PPPoE. Формат команды:

show
show statistics checkpoint 0
: Вывод состояния, числа изменений состояния UP/DOWN, и текущей статистики объекта, начиная от момента старта устройства или данного объекта. Две вышеуказанные команды являются синонимами. Формат вывода частично варьируется в зависимости от типа объекта.
Для объектов, представляющих собой IP-интерфейсы, в статистике учитывается только IP-трафик. Для портов учитывается трафик канального уровня (в т.ч. заголовки канального уровня, пакеты *keepalive*, пакеты LCP и др.). Дополнительные байты физического уровня (бит-стаффинг и др.) в статистике не учитываются.

show statistics checkpoint <1..15> [set | unset]
: Вывод статистики относительно указанной контрольной точки. Опциональные команды set и unset устанавливают и удаляют контрольную точку, соответственно.
Нулевой точкой для сбора статистики, которая присутствует всегда, является момент старта устройства (или объекта). Помимо нее, для каждого объекта может быть установлено до 15 дополнительных контрольных точек.

show statistics checkpoint all unset
: Удаление всех установленных контрольных точек.

Кроме того, просмотреть характеристики интерфейса возможно при помощи команды show interface <имя> в меню обычного или привилегированного режима, например:

```
nsg# show interface eth0

Interface eth0
    index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,ALLMULTI,MULTICAST>
    HWaddr: 00:09:56:10:03:27
    inet 10.0.0.81/8
        input packets 3001, bytes 201992, dropped 0, multicast packets 0
        input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
        output packets 1940, bytes 126264, dropped 0
        output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
        collisions 0
```

§4.1.6. Другие параметры IP-интерфейсов

Параметры IP-интерфейса, связанные с динамической маршрутизацией, настраиваются в меню конфигурирования Zebra (config)#. Для этой цели имеются команды:

interface <имя>
: Вход в меню настройки IP-интерфейса. Здесь <имя> — символическое имя интерфейса, например, s1.0 или eth0. (Возможные форматы имен интерфейсов перечислены в предыдущем параграфе).

После входа в меню IP-интерфейса системное приглашение принимает вид (config-if)#. В этом меню имеются следующие команды:

ip address <ip-префикс>
no ip address <ip-префикс>
: Установка и удаление IP-префикса. При этом, если длина маски 30 бит или менее, автоматически генерируется еще один адрес, в котором все свободные биты маски установлены в единицу (т.е. самый старший адрес из данного диапазона). Для интерфейсов Ethernet и VLAN это будет широковещательный адрес (broadcast), для интерфейсов "точка-точка" (PPP, Cisco-HDLC и Frame Relay) — адрес удаленной стороны (pointopoint). IP-адрес групповой рассылки данной командой не настраивается.

ВНИМАНИЕ Если IP-префикс назначен средствами Zebra либо средствами NSG, то удалять/изменять его необходимо этим же способом. Во избежание возникающей неоднозначности, для назначения IP-адресов рекомендуется всегда назначать IP-префикс средствами NSG (см. п.4.1.2). Использовать назначение IP-префикса средствами Zebra рекомендуется только для специфических интерфейсов, не настраиваемых в меню NSG — например, для фиктивного интерфейса dummy0.

shutdown
no shutdown
: Выключение и включение интерфейса. В отличие от команды { adm-state up | down } в меню настройки портов (см. Часть 2), данные команды позволяют управлять состоянием отдельных суб-интерфейсов (DLCI, VLAN) на порту, не рестартуя весь порт.

multicast
no multicast
: Разрешение и запрещение широковещательной рассылки на интерфейсе.

ospf ...
no ospf ...
: Группа команд, управляющих работой протокола маршрутизации OSPF на данном интерфейсе. Подробнее о данных командах см. п.4.6.

bandwidth <1...10000000>
no bandwidth <1...10000000>
: Определение и удаление доступной полосы пропускания для вычисления стоимости маршрутов, проходящих через данный интерфейс, в протоколе OSPF. Данный параметр является справочным, не влияет на фактическую скорость работы данного интерфейса и никак не связан с параметром baudrate в конфигурации физического порта (см. Часть 2).

§4.1.7. Псевдо-интерфейсы

Помимо IP-интерфейсов, реально существующих в системе всё время, NSG Linus предусматривает назначение всего набора параметров IP для временных интерфейсов, которые могут создаваться и уничтожаться динамически. Примером такого объекта является виртуальный IP-интерфейс (сокет типа net) системы *ui*TCP. Для его описания используется следующая команда в меню (config-nsg)#:

pseudo-interface <имя>
no pseudo-interface <имя>
: Создание псевдо-интерфейса с указанным именем и вход в меню его редактирования и удаление псевдо-интерфейса, соответственно. Меню псевдо-интерфейса содержит команду ip address ... и все другие параметры, относящиеся к уровню IP.

Как только в системе создаётся IP-интерфейс, имя которого совпадает с именем одного из псевдо-интерфейсов, к нему немедленно применяются все настройки, сделанные для этого псевдо-интерфейса.

Помимо динамических интерфейсов, команда может быть использована для назначения IP-адресов и других параметров IP специальным интерфейсам lo, dummy и т.п., не привязанным непосредственно к объектам 1–2 уровней.

§4.2. Статическая маршрутизация

§4.2.1. Создание, удаление и просмотр маршрутов

Настройка статической маршрутизации производится в главном меню конфигурации Zebra (config)# следующими командами:

ip route <ip-префикс> <шлюз> <расстояние>
no ip route <ip-префикс> <шлюз> <расстояние>

Создать/удалить маршрут к некоторой сети. Сеть описывается IP-префиксом, т.е. совокупностью адреса и длины маски, например, 10.1.1.123/8. Шлюз, через который следует отправлять пакеты в данную сеть, может быть указан либо IP-адресом, либо символическим именем. Если этот параметр имеет формат IP-адреса в десятичной дотовой нотации, то он рассматривается как адрес, в противном случае — как имя. Примеры:

```
ip route 10.0.0.0/8 11.0.0.2
ip route 10.0.0.0/8 s1.1
```

Первый пример описывает маршрут в сеть 10.0.0.0/8 через шлюз с адресом 11.0.0.2; этот адрес может как принадлежать одному из IP-интерфейсов устройства, так и не принадлежать ему (в этом случае производится новый поиск в маршрутной таблице, чтобы определить маршрут к указанному шлюзу). Во втором примере пакеты, адресованные в сеть 10.0.0.0/8, отправляются через интерфейс s1/1.

Расстояние, или длина маршрута — целочисленный параметр, назначаемый административно в пределах от 1 до 255. По существу он используется как приоритет в таблице маршрутизации. Чем меньше длина, тем более предпочтителен данный маршрут (наивысший приоритет — 1). Если длина не указана, по умолчанию она считается равной 1.

ПРИМЕЧАНИЕ При указании имени интерфейса маршрут считается ведущим в непосредственно подключённую сеть. В этом случае он имеет приоритет перед всеми маршрутами, полученными с помощью протоколов динамической маршрутизации.

ПРИМЕЧАНИЕ Административная длина маршрута, в общем случае, отличается от *метрики* маршрута. Метрика вычисляется автоматически в результате работы протоколов динамической маршрутизации.

ip route <ip-адрес> <маска> <шлюз> <расстояние>
no ip route <ip-адрес> <маска> <шлюз> <расстояние>

Другой синтаксис той же команды. Вместо указания IP-префикса сеть может быть описана отдельными адресом и маской. Параметры <шлюз> и <расстояние> используются так же, как и в предыдущем случае. Так, два вышеприведенных примера равносильны следующим командам:

```
ip route 10.0.0.0 255.0.0.0 11.0.0.2
ip route 10.0.0.0 255.0.0.0 s1.1
```

ВНИМАНИЕ Указание маски подсети назначения в той или иной форме является обязательным. Если маска (или длина маски) не указана, автоматическое вычисление маски по классу сети не производится, а команда считается введенной неверно.

Маршрут может быть действующим или не действующим (например, если интерфейс, через который он проходит, находится в состоянии DOWN или если не определен маршрут к указанному шлюзу). При выборе маршрута для каждого пакета ищется действующая запись с максимальной длиной маски, т.е. чем более точно описана сеть назначения, тем больший приоритет имеет данный маршрут. Если к одной подсети имеется несколько действующих маршрутов, то при прочих равных условиях из них выбирается маршрут с наименьшим расстоянием.

ПРИМЕЧАНИЕ Если маршрутов в одну сеть назначения с наибольшей длиной маски и наименьшим расстоянием оказывается несколько, то осуществляется балансировка нагрузки между всеми этими маршрутами (см. п.4.10.10).

Чтобы просмотреть таблицу маршрутизации, используется команда show ip route в меню обычного или привилегированного режима:

```
nsg# show ip route

Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
B - BGP, > - selected route, * - FIB route

C>* 10.0.0.0/8 is directly connected, eth0
S>  11.0.0.0/8 [3/0] via 15.0.0.2, s1.17 inactive
  *                  via 10.0.0.3, eth0
S>* 12.0.0.0/8 [1/0] via 14.0.0.2, s4.0
S   12.0.0.0/8 [2/0] via 10.0.0.3, eth0
C>* 127.0.0.0/8 is directly connected, lo
```

В таблице показываются все маршруты, определенные в системе — как действующие в данный момент, так и не действующие по какой-либо причине. Записи упорядочены по адресам назначения и по расстояниям (причем этот порядок не совпадает с фактическим порядком следования записей во внутреннем представлении таблицы). Обозначения в таблице:

K	Маршрут сконфигурирован средствами ОС Linux; работоспособность таких маршрутов не контролируется демоном Zebra
C	Маршрут в сеть, подключенную непосредственно к одному из IP-интерфейсов маршрутизатора
S, R, O, B	Происхождение маршрута (статический или созданный одним из протоколов динамической маршрутизации — RIP, OSPF, BGP, соответственно)
>	Маршрут имеет наивысший приоритет среди всех действующих маршрутов с данной сетью назначения
*	Маршрут, используемый для данной сети назначения в данное время
[m/n]	Административная длина маршрута и его метрика (для статических маршрутов второй параметр равен нулю)

В вышеприведенном примере к сети 11.0.0.0/8 определено два маршрута с одинаковой длиной, но первый из них (через интерфейс s1) неактивен. Поэтому, хотя он и указан как предпочтительный, используется маршрут через интерфейс eth0. Для сети 12.0.0.0/8 также указано два маршрута, однако длина у этих маршрутов разная, поэтому пакеты отправляются через шлюз 14.0.0.2.

§4.2.2. Маршруты по умолчанию

Специальной команды для настройки шлюза по умолчанию (*default gateway*) в NSG Linux не предусмотрено. Шлюз по умолчанию конфигурируется как маршрут общего вида с адресом сети назначения 0.0.0.0/0. Поскольку длина маски в данной записи нулевая, она имеет наименьший приоритет и рассматривается в том случае, если для пакета не найден более специфический маршрут. Пример:

```
ip  route  0.0.0.0/0  eth0
```

Как и других маршрутов, маршрутов по умолчанию может быть несколько. В этом случае применяются общие правила выбора: из числа действующих маршрутов выбирается маршрут с наименьшей длиной, а если таковых несколько — то первый по порядку. Пример:

```
nsg# show ip route

Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] is directly connected, eth0
S                   is directly connected, s1
S   0.0.0.0/0 [2/0] is directly connected, s4
```

§4.2.3. Множественные таблицы маршрутизации

Ядро NSG Linux позволяет хранить и использовать несколько таблиц маршрутизации, обозначаемых номерами от 1 до 254. В командной оболочке vtysh такая возможность в настоящее время полностью не реализована. Команды

```
table <1...254>   — в главном меню конфигурации Zebra (config)#

show table        — в меню обычного и привилегированного режимов
```

зарезервированы для последующего использования. Расширенные возможности маршрутизации с использованием нескольких таблиц (маршрутизация на основе установленных правил и др.) доступны в данной версии только средствами командной оболочки ОС Linux.

§4.3. Списки доступа (*access lists, prefix lists*)

Списки доступа позволяют сортировать входящий и исходящий трафик, как пользовательский, так и служебный. Такая сортировка необходима для работы многих служб — фильтрации пакетов, NAT, протоколов динамической маршрутизации, VPN и др.

Каждый список состоит из одной или нескольких записей, по которым проверяются все пакеты, обрабатываемые той или иной службой маршрутизатора. В настройках соответствующих служб указывается список доступа, по которому следует проверять пакеты, и действия, которые следует предпринять, если пакет *соответствует* этому списку; если пакет не соответствует списку, эти действия не выполняется.

В программном обеспечении NSG Linux используются списки доступа в нескольких форматах, а именно:

— простые *access lists* NSG с номерами от 1...99 и 1300...1999
— расширенные *access lists* NSG с номерами от 100...199 и 2000...2699
— простые нумерованные *access lists* пакета Zebra с номерами от 1...99 и 1300...1999
— расширенные нумерованные *access lists* пакета Zebra с номерами от 100...199 и 2000...2699
— именованные *access lists* пакета Zebra
— именованные *prefix lists* пакета Zebra

Простые нумерованные и именованные *access lists* используют наиболее простые и употребительные критерии; расширенные *access lists* и *prefix lists* содержат обширный набор критериев для гибкой сортировки трафика. Разделение списков доступа на NSG и Zebra является временным; эти списки используются для настройки функции, доступных через меню конфигурации NSG (config-nsg)# и через основное меню конфигурации Zebra (config)#, соответственно. В будущем такое дублирование будет устранено по мере переноса команд Zebra в единый узел меню NSG.

ПРИМЕЧАНИЕ В отличие от командного языка других производителей, access list 0 рассматривается как именованный список доступа с текстовым именем 0.

Каждая запись, входящая в список, содержит некоторый критерий, которому должен удовлетворять пакет, и ключевое слово permit (разрешить) либо deny (запретить). Такой синтаксис сложился в Cisco-подобных командных языках исторически. На практике эти ключевые слова следует рассматривать как утверждения "пакет соответствует списку" и "пакет не соответствует списку", соответственно. Записи рассматриваются последовательно, и после срабатывания одной из записей проверка прекращается.

Любой список заканчивается неявной записью deny any, т.е. если пакет не соответствует ни одной записи из данного списка, то он отвергается. Если список, на который ссылается некоторая служба, отсутствует, то по умолчанию принимается правило permit any, т.е. под действие данной службы подпадают все пакеты.

§4.3.1. Именованные *access lists* пакета Zebra

Именованные списки доступа позволяют сортировать пакеты по принадлежности к некоторой IP-сети, которая может трактоваться как сеть источника пакета, или как сеть назначения пакета, или как-либо иначе в зависимости от того, какая служба использует данный *access list*. Для управления именованными списками доступа используются следующие команды в главном меню конфигурации Zebra (config)# :

access-list <имя> [<комментарий>] { permit | deny } {<ip-префикс> | any }
Создать запись в *access list*. Комментарий является необязательным и предназначен исключительно для удобства администрирования. Критерием для отбора пакетов является либо префикс подсети, содержащий ее IP-адрес и длину маски, либо ключевое слово any, означающее любую сеть.

no access-list <имя> [<комментарий>] { permit | deny } {<ip-префикс> | any }
Удалить данную запись из *access list*.

no access-list <имя> [<комментарий>]
Удалить данный *access list* целиком.

Записи в именованных списках типа *access list* следуют в том же порядке, в каком они были введены пользователем.

Пример. Следующему списку доступа соответствуют все пакеты с адресами от 10.128.0.0 до 10.255.255.255:

```
access-list filter deny 10.0.0.0/9
access-list filter permit 10.0.0.0/8
```

§4.3.2. Простые нумерованные *access lists* пакета Zebra

Простые нумерованные списки доступа позволяют сортировать пакеты по IP-адресам, которые могут трактоваться как источник пакета, или как назначение пакета, или как-либо иначе в зависимости от того, какая служба использует данный *access list*. Номер списка должен находиться в диапазоне 1...99 или 1300...1999. Для управления простыми нумерованными списками доступа используются следующие команды в главном меню конфигурации Zebra (config)# :

access-list <номер> [<комментарий>] { permit | deny } <ip-адрес> <шаблон>
no access-list <номер> [<комментарий>] { permit | deny } <ip-адрес> <шаблон>
Создать/удалить запись в *access list*. Комментарий является необязательным и предназначен исключительно для удобства администрирования. Критерием для отбора пакетов является принадлежность к некоторому множеству IP-адресов, которое определяется указанным адресом и шаблоном.

ПРИМЕЧАНИЕ Шаблон адреса (*wildcard bits*), как и IP-адрес, представляет собой число длиной 4 байта, представленное в десятичной ботовой нотации. Оно содержит двоичные единицы в тех битах, значения которых могут варьироваться. Иногда шаблон называется также *инверсией маски*; однако в общем случае это более широкое понятие, поскольку нулевые/ненулевые биты могут чередоваться в нем произвольным образом. Примеры:
— шаблон 0.0.0.7 соответствует маске длиной 29 бит (255.255.255.248);
— шаблон 0.0.0.3 соответствует маске длиной 30 бит (255.255.255.252);
— шаблон 0.0.0.5 не может быть описан никакой маской (в последнем байте допускается изменение 0 и 2 битов, но не допускается изменение битов 1 и 3…7).

access-list <номер> [<комментарий>] { permit | deny } host <ip-адрес>
no access-list <номер> [<комментарий>] { permit | deny } host <ip-адрес>
Создать/удалить запись в *access list*. Критерием для отбора пакетов является один фиксированный IP-адрес.

access-list <номер> [<комментарий>] { permit | deny } any
no access-list <номер> [<комментарий>] { permit | deny } any
Создать/удалить запись, под которую подпадают пакеты с любыми IP-адресами.

no access-list <номер> [<комментарий>]
Удалить данный *access list* целиком.

Записи в нумерованных *access lists* следуют в том же порядке, в каком они были введены пользователем.

Пример. Следующему списку доступа соответствуют все пакеты с нечетными адресами вида 10.x.x.x:

```
access-list 51 permit 10.0.0.1 0.255.255.254
```

§4.3.3. Расширенные нумерованные *access lists* пакета Zebra

Расширенные нумерованные списки доступа позволяют сортировать пакеты по IP-адресам источника и назначения одновременно. Номер списка должен находиться в диапазоне 100...199 или 2000...2699. Для управления расширенными нумерованными списками доступа используются следующие команды в главном меню конфигурации Zebra (config)# :

access-list <номер> [<комментарий>] { permit | deny } ip <источник> <назначение>
no access-list <номер> [<комментарий>] { permit | deny } ip <источник> <назначение>
Создать/удалить запись в *access list*. Комментарий является необязательным и предназначен исключительно для удобства администрирования. Источник и назначение пакета могут быть заданы в любом из трех форматов:

<ip-адрес> <шаблон>	совокупность базового IP-адреса и шаблона, которым должен удовлетворять адрес источника или назначения, соответственно
host <ip-адрес>	одиночный IP-адрес
any	любой IP-адрес

Пакет соответствует данной записи, если и адрес источника, и адрес назначения попадают в указанные множества.

no access-list <номер> [<комментарий>]
Удалить данный *access list* целиком.

Записи в нумерованных *access lists* следуют в том же порядке, в каком они были введены пользователем.

Пример. Под нижеприведенное правило (номер 151) подпадает любой трафик, посылаемый из сети 11.0.0.0 с маской 255.0.0.0 в сеть 12.0.0.0 с маской 255.0.0.0:

```
access-list 151 permit ip 11.0.0.0 0.255.255.255 12.0.0.0 0.255.255.255
```

§4.3.4. Простые *access lists* NSG

Простые нумерованные *access lists* NSG доступны через узел меню (config-nsg)# и предназначены для настройки объектов, находящихся внутри данного узла. По своим возможностям они аналогичны простым нумерованным *access lists* пакета Zebra, но снабжены более гибким и удобным набором команд для их настройки.

Для управления *access lists* данного типа используются следующие команды меню (config-nsg)#:

access-list std-ip <номер>
no access-list std-ip <номер>
Создать/редактировать и удалить *access list*. Первая команда создает *access list* с заданным номером, если он не существует, и входит в меню его конфигурации. Номера списков могут быть в диапазонах 1...99 и 1300...1999.

Записи в *access list* нумеруются и могут вводиться в произвольном порядке. Редактирование списка осуществляется в меню (config-access-...-NN)# при помощи следующих команд:

description <комментарий>
Ввести текстовое описание (строка) для данного *access list*. Если строка содержит пробелы, она должна быть заключена в кавычки. Максимальная длина описания — 255 символов.

add <приоритет> { permit | deny } <IP-адреса>
Добавить/редактировать запись в *access list*. Приоритет, он же порядковый номер записи, определяет порядок следования записей в списке и может принимать значения от 1 до 10000. Параметр < IP-адреса > определяет множество IP-адресов, на которые должна действовать данная запись, и может быть записан в одном из следующих форматов:

<ip-адрес> <шаблон>	совокупность базового IP-адреса и шаблона (*wildcard bits)*
host <ip-адрес>	одиночный IP-адрес
any	любой IP-адрес

Данный адрес может трактоваться как источник пакета, или как назначение пакета, или как-либо иначе в зависимости от того, какая служба использует данный *access list.*

delete < приоритет>
Удалить запись с заданным номером из *access list.*

ПРИМЕЧАНИЕ При изменении *access list* в данной версии NSG Linux требуется рестарт всех объектов, которые его используют.

Примеры. Нижеприведенная последовательность команд создает *access list* с номером 37 и описанием "My access list":

```
access-list std-ip 37 remark "My access list"
access-list std-ip 37 add 10 permit 10.0.0.1 0.0.255.255
access-list std-ip 37 add 20 deny 12.0.0.5 0.0.0.0
access-list std-ip 37 add 13 deny 10.1.0.1
access-list std-ip 37 add 25 permit 10.0.0.2 0.0.0.255
access-list std-ip 37 add 17 permit 10.0.0.1 0.255.0.255
access-list std-ip 37 add 19 permit 13.0.0.1 0.0.0.255
access-list std-ip 37 add 13 deny 10.1.0.2
access-list std-ip 37 delete 19
```

Список содержит пять записей, которые делают следующее, в соответствии с их приоритетами:

запись 10 — допускает все адреса вида 10.0.x.x
запись 13 — блокирует адрес 10.1.0.2; запись с приоритетом 13 введена 2 раза, поэтому применяется последний вариант
запись 17 — допускает адреса вида 10.x.0.x, за исключением запрещенного ранее (запись 13) 10.1.0.2
запись 19 — введена, а затем удалена, т.е. более в списке не присутствует
запись 20 — блокирует адрес 12.0.0.5
запись 25 — ничего не делает, так как адрес 10.0.0.2 уже допущен в записи 10

Все остальные варианты блокируются по умолчанию.

Следующая команда удаляет весь *access list* с номером 37:

```
no access-list std-ip 37
```

§4.3.5. Расширенные *access lists* NSG

Расширенные нумерованные *access lists* NSG доступны через узел меню (config-nsg)# и предназначены для настройки объектов, находящихся внутри данного узла. По своим возможностям они аналогичны расширенным нумерованным *access lists* пакета Zebra и предоставляют дополнительные критерии для сортировки пакетов.

Для управления *access lists* данного типа используются следующие команды меню (config-nsg)#:

access-list ext-ip <номер>
no access-list ext-ip <номер>
Создать/редактировать и удалить *access list*. Первая команда создает *access list* с заданным номером, если он не существует, и входит в меню его конфигурации. Номера списков могут быть в диапазонах 100...199 и 2000...2699.

Записи в *access list* нумеруются и могут вводиться в произвольном порядке. Редактирование списка осуществляется в меню (config-access-...-NN)# при помощи следующих команд:

description <комментарий>
Ввести текстовое описание (строка) для данного *access list*. Если строка содержит пробелы, она должна быть заключена в кавычки. Максимальная длина описания — 255 символов.

add <приоритет> { permit | deny } <протокол> [<параметры_протокола>] [tos <тип_сервиса>]
Добавить/редактировать запись в *access list*. Приоритет, он же порядковый номер записи, определяет порядок следования записей в списке и может принимать значения от 1 до 10000. Из остальных параметров обязательным является только протокол, который указывается либо номером, либо ключевым словом (ip, tcp, udp, icmp, и т.п.) За ним следует набор параметров, состав которого зависит от выбранного протокола. Частные случаи для различных протоколов см. ниже.
Последний параметр tos позволяет сортировать пакеты по значениям поля ToS (DiffServ) в заголовке IP-пакета. В данной версии NSG Linux параметр может принимать только фиксированные значения из следующего списка:

minimize-delay	Сервис Minimize Delay	(ToS=8)
maximize-throughput	Сервис Maximize Throughput	(ToS=4)
maximize-reliability	Сервис Maximize Reliability	(ToS=2)
minimize-cost	Сервис Minimize Monetary Cost	(ToS=1)
normal-service	Сервис Normal Service	(ToS=0), значение по умолчанию

Смешанные значения запрещены.

delete < приоритет>
Удалить запись с заданным номером из *access list*.

Для различных протоколов команда add принимает следующий вид:

add <приоритет> { permit | deny } ip <источник> <назначение> [tos <тип_сервиса>]
Запись относится ко всем IP-пакетам независимо от протокола, инкапсулируемого в IP. При этом параметры <источник> и <назначение> определяют два множества IP-адресов (*source addresses* и *destination addresses*, соответственно), на которые должна действовать данная запись. Оба эти параметра могут быть записаны в одном из следующих форматов:

<ip-адрес> <шаблон>	совокупность базового IP-адреса и шаблона (*wildcard bits)*
host <ip-адрес>	одиночный IP-адрес
any	любой IP-адрес

Эти два параметра являются обязательными и могут быть заданы только парой.

add <приоритет> { permit | deny } icmp <источник> <назначение> [icmp-type <0...255> [icmp-code <0...255>]] [tos <тип_сервиса>]
Запись относится только к пакетам ICMP. В списке параметров, после обязательных IP-адресов, можно указать также тип и код пакетов ICMP.

add <приоритет> { permit | deny } udp <источник> [<порт>] <назначение> [<порт>] [tos <тип_сервиса>]
Запись относится только к пакетам UDP. В списке параметров, помимо обязательных IP-адресов, можно указать также номера портов UDP источника и назначения, в одном из следующих форматов:

eq <0...65535>	один указанный номер порта
neq <0...65535>	все номера портов, кроме указанного
lt <0...65535>	все номера портов младше указанного (исключительно)
gt <0...65535>	все номера портов старше указанного (исключительно)
range <0...65535> <0...65535>	все номера портов в указанном диапазоне
nrange <0...65535> <0...65535>	все номера портов, кроме указанного диапазона

Если номер порта (источника или назначения) не указан, запись действует на пакеты с любым номером порта.

add <приоритет> { permit | deny } tcp <источник> [<порт>] <назначение> [<порт>] ↵
[syn {1|0}] [ack {1|0}] [fin {1|0}] [rst {1|0}] [urg {1|0}] [psh {1|0}] [tos <тип_сервиса>]
Запись относится только к пакетам TCP. В списке параметров, помимо обязательных IP-адресов, можно указать также номера портов TCP источника и назначения, как и для протокола UDP. Помимо этого, для протокола TCP можно указать также наличие флагов, определяющих состояние TCP-соединения. Единица соответствует установленному флагу, ноль — снятому. При отсутствии указания на какой-либо флаг он не проверяется,

add <приоритет> { permit | deny } <0...255> <источник> <назначение> [tos <тип_сервиса>]
Запись относится к IP-пакетам с указанным номером протокола, инкапсулируемого в IP. В частности, 1=ICMP, 6=TCP, 17=UDP; однако при таком формате команды, в отличие от приведенных выше, не допускаются специфические параметры, относящиеся к данным протоколам. Список протокольно-зависимых параметров в данном случае может содержать только IP-адреса источника и назначения.

Пример использования флагов TCP:

```
nsg
    access-list ext-ip 100
        add 1 deny tcp any any syn 1 ack 0 fin 0
        add 2 permit tcp any any
```

Данное правило выделяет пакеты, инициирующие TCP-соединение — с установленным флагом SYN и не установленными ACK и FIN. Далее, используя это правило, можно заблокировать запросы на установление TCP-соединений в определенном направлении. Предположим, например, что к порту eth0 устройства подключена локальная сеть офиса, а к порту s1 — сеть поставщика услуг. Хосты из сети офиса должны устанавливать TCP-соединения с внешним миром без ограничений. Хостам же из внешнего мира разрешено только отвечать на запросы, посылать пакеты данных по установленному соединению и разрывать соединение; инициировать соединения с хостами внутренней сети они не должны. Для этой цели фильтр подключается следующим образом:

```
nsg
    port s1
        access-group transit input 100
```

(Подробнее о включении IP-фильтров см. §4.9.)

В результате получается требуемый эффект "полупрозрачного зеркала". Данная ситуация отчасти схожа с тем, что имеет место при IP-маскарадинге и Source NAT: хосты внутренней сети становятся недоступными из Интернет. Однако использование фильтра для этой цели имеет два принципиальных отличия: а) правило фильтрации относится только к TCP-соединениям; б) IP-адреса и номера портов не изменяются.

§4.3.6. Списки доступа типа *prefix lists*

Списки доступа типа *prefix lists* применяются для фильтрации маршрутов, полученных по протоколам динамической маршрутизации, и используют несколько иной набор критериев, чем *access lists*. Они позволяют установить отдельно диапазон IP-адресов и отдельно — требования к длине маски. Можно указать максимальную длину, или минимальную, или оба ограничения одновременно. Таким образом, они позволяют разделять маршруты к подсетям разного размера внутри некоторой заданной сети.

Записи в списке упорядочены по номерам (приоритетам) и рассматриваются последовательно, от наименьших номеров к наибольшим. Кроме того, списки этого типа предоставляют дополнительные возможности, такие как учет статистики и т.п.

Для управления *prefix lists* используются следующие команды в главном меню конфигурации Zebra (config)# :

ip prefix-list <имя> { permit | deny } <ip-префикс> [ge <min>] [le <max>]
ip prefix-list <имя> seq <1...4294967295> { permit | deny } <ip-префикс> [ge <min>] [le <max>]
Создать критерий, входящий в *prefix list*. Параметры команды:

<имя>	Символическое имя *prefix list*.
<1...4294967295>	Порядковый номер данной записи в списке. Если номер не указывается, то записи автоматически присваивается номер, кратный 5 и бóльший последнего существующего номера. Если номер указывается явно, то он может быть любым, в т.ч. новые записи могут вставляться на неиспользуемые номера между существующими.
<ip-префикс>	Базовый IP-префикс подсети (в формате a.b.c.d/n), к которой должен принадлежать/не принадлежать рассматриваемый маршрут.
ge <min>	Минимальная длина маски. Чтобы пакет соответствовал данной записи, длина маски в рассматриваемом маршруте должна быть не менее (*greater or equal*), чем указанная.

le <max> Максимальная длина маски. Чтобы пакет соответствовал данной записи, длина маски в рассматриваемом маршруте должна быть не более (*less or equal*), чем указанная.

Длина маски, указанная в префиксе, и два последних параметра должны удовлетворять соотношению:

$$n < min \leq max$$

Порядок следования параметров ge <min> и le <max> может быть любым. Допускается использовать один или оба параметра, или не использовать их вовсе. Если ни один из параметров не используется, тогда, чтобы пакет соответствовал данной записи, длина маски в рассматриваемом маршруте должна быть в точности равна указанной в префиксе.

Если создается новая запись, в точности совпадающая с существующей, но имеющая другой номер, будет выдано сообщение об ошибке. Однако если и номер, и все параметры новой записи совпадают с существующей, это ошибкой не считается. Если создается новая запись с существующим номером, но другими параметрами, то она заменяет собой существующую.

no ip prefix-list <имя> { permit | deny } <ip-префикс> [le <макс.длина>] [ge <мин.длина>]
no ip prefix-list <имя> seq <1...4294967295> { permit | deny } <ip-префикс> [le <длина>] [ge <длина>]
Удалить существующую запись из *prefix list*. Параметры команды должны в точности совпадать с параметрами, указанными при создании этой записи.

no ip prefix-list <имя>
Удалить данный *prefix list* целиком.

ip prefix-list <имя> description <описание>
no ip prefix-list <имя> description [<описание>]
Создать/удалить описание записи. Описание используется для удобства администрирования и не оказывает влияния на работу системы. При удалении описания повторять его полностью не требуется.

ip prefix-list sequence-number
no ip prefix-list sequence-number
Включить/выключить вывод порядковых номеров при просмотре *prefix lists*. По умолчанию, порядковые номера выводятся.

Для просмотра существующих *prefix lists* и информации об их использовании предусмотрены следующие команды show в меню обычного или привилегированного режима:

show ip prefix-list [<имя>]
Вывести все *prefix lists*, определенные в системе, либо *prefix list* с указанным именем.

show ip prefix-list <имя> seq <1...4294967295>
Вывести указанную запись *prefix list* и статистику ее использования.

show ip prefix-list <имя> <ip-префикс> [longer | first-match]
Вывести все записи с указанным префиксом, входящие в данный *prefix list*. Если дополнительно указан ключ longer, выводятся все записи, в которых использован тот же адрес сети, а длина префикса больше или равна указанной. Если дополнительно указан ключ first-match, то выводится только первая запись с данным префиксом и статистика ее использования.

show ip prefix-list summary [<имя>]
Вывести краткие сведения обо всех *prefix lists*, определенных в системе, либо о *prefix list* с указанным именем.

show ip prefix-list detail [<имя>]
Вывести полностью все *prefix lists*, определенные в системе, либо *prefix list* с указанным именем, а также статистику использования каждой записи.

Для сброса статистики *prefix lists* предусмотрена следующие команды clear в меню привилегированного режима:

clear ip prefix-list [<имя>]
Обнулить счетчики всех *prefix lists*, определенных в системе, либо *prefix lists* с указанным именем.

clear ip prefix-list <имя> <ip-префикс>
Обнулить счетчики всех записей указанного *prefix list*, содержащих данный префикс.

§4.4. Правила преобразования маршрутов (*route maps*)

Правила преобразования маршрутов (*route maps*) — еще один универсальный инструмент, предназначенный для тонкой настройки IP-маршрутизации. *Route map* представляет собой фильтр, позволяющий избирательно применять определенные действия к определенным маршрутам. Он используется при импорте маршрутов из одного протокола маршрутизации в другой; с его помощью возможно установить те атрибуты маршрута, которые являются специфическими для нового протокола.

Правило состоит из одной или нескольких записей, упорядоченных по приоритетам. Для создания *route map* используется следующая команда в главном меню конфигурации Zebra (config)#:

route-map <имя> { permit | deny } [<приоритет>]

Создать в указанном правиле запись с указанным приоритетом, если она не существует, и перейти в меню ее конфигурации.

Второй параметр определяет применение данной записи:

- permit — Если маршрут соответствует критериям match данной записи, обработать его в соответствии с инструкциями set. Если не соответствует, продолжить проверку по записям данного правила с более низким приоритетом.
- deny — Если маршрут соответствует критериям match данной записи, не обрабатывать его и не проверять по остальным правилам с данным именем. Инструкции set данной записи игнорируются.

Если маршрут не соответствует ни одной записи с данным именем, он не обрабатывается.

Последний параметр определяет место данной записи в списке. Чем меньше этот параметр, тем выше приоритет записи.

ВНИМАНИЕ В отличие от списков доступа, для правил преобразования маршрутов по умолчанию принимается правило deny any. Иначе говоря, если команда redistribute ссылается на отсутствующее правило, то данный маршрут не обрабатывается.

Далее в меню (config-route-map)# определяются критерии, по которым следует проверять маршруты (команды match) и действия, которые следует предпринять (команды set). Общими для всех протоколов являются команды:

match interface <имя>

Рассматриваемый пакет RIP должен выходить из маршрутизатора NSG через указанный интерфейс. В данной реализации команда допускает указание только одного выходного интерфейса.

match ip address <access_list>

Сеть назначения для данного маршрута должен быть разрешена указанным списком доступа. Команда может использовать любой нумерованный или именованный список доступа.

match ip address prefix-list <prefix_list>

Команда аналогична предыдущей, но использует список доступа типа *prefix list* вместо *access list*.

match metric <0…4294967195>

Маршрут должен иметь указанную метрику.

ПРИМЕЧАНИЕ В отличие от реализации Cisco, правила *route map* применяются в NSG Linux не при внесении маршрутов из входящих сообщений RIP и OSPF в локальную таблицу маршрутизации, а при генерации исходящих сообщений на основе имеющейся таблицы.

set metric <0…4294967195>

В исходящих сообщениях установить для данного маршрута указанную метрику.

ПРИМЕЧАНИЕ Диапазон допустимых значений параметра в командах match metric и set metric выбран из соображений совместимости со всеми протоколами маршрутизации. Однако применительно к конкретным протоколам он может быть долее узким. Так, при использовании протокола RIP значения больше 16 не имеют смысла.

Другие команды match и set являются специфическими для конкретных протоколов маршрутизации и рассмотрены в соответствующих разделах данного документа.

Для удаления *route map* используются следующие команды в главном меню конфигурации Zebra (config)#:

no route-map <имя>
no route-map <имя> { permit | deny } <приоритет>

Первая команда удаляет правило целиком, вторая — только указанную запись.

§4.5. Протокол RIP

Программное обеспечение NSG Linux поддерживает протокол динамической маршрутизации RIP версий 1 и 2. В рамках данного протокола каждый маршрутизатор рассылает смежным маршрутизаторам сообщения об известных ему сетях и метриках маршрутов до этих сетей в соответствии со своей текущей таблицей маршрутизации. Эта информация может быть как получена средствами самого протокола RIP, так и импортирована из других источников (таблицы маршрутизации в ядре Linux, таблицы статических маршрутов. других протоколов динамической маршрутизации).

Маршрутизатор, получивший сообщение RIP, сверяет полученную информацию со своей таблицей маршрутизации и вычисляет маршрут с наименьшей метрикой. Процедуры создания, изменения и удаления записей в таблице маршрутизации регламентированы таким образом, чтобы гарантировать сходимость маршрутов в масштабе всей сети за конечное время.

ПРИМЕЧАНИЕ Рассылка сообщений RIP производится в виде UDP-датаграмм с портами источника и назначения 520. Исключением являются только запросы RIP, которые могут отсылаться с любого порта, и ответы на них, которые должны быть адресованы на тот же порт, с которого получен запрос. Это необходимо учитывать при настройке фильтров.

Протокол RIP относится к внутренним протоколам маршрутизации (Interior Gateway Protocol, IGP) и предназначен для использования внутри автономных систем (AS). По сравнению с другими протоколами динамической маршрутизации, он характеризуется следующими особенностями:

— Относительно прост и менее требователен к вычислительным ресурсам.
— Пригоден для систем ограниченного размера, в которых маршрут между любыми двумя сетями проходит не более чем через 15 маршрутизаторов.
— При отказе на одном из участков сети в крупных системах перестройка всех маршрутных таблиц может — по крайней мере, теоретически — занять достаточно больше время.
— Каждый маршрут или участок маршрута имеет фиксированную метрику; метрика не может учитывать текущую степень загрузки, величину задержки пакетов или другие параметры маршрута, измеряемые в реальном времени.

Основное практическое различие между версиями RIP 1 и 2 состоит в том, что RIP v1 не предусматривает передачу маски подсети назначения; длина маска определяется автоматически на основании класса сети (8, 16 или 24 бита). RIP v2 позволяет передавать полное описание произвольной IP-сети, состоящее из ее адреса и маски. Кроме того, RIP v2 предусматривает передачу адреса следующего узла на маршруте (*next hop*), дополнительного тега маршрута, а также аутентификацию сообщений. Для регулярной рассылки маршрутных таблиц в RIP v1 используются широковещательные сообщения (*broadcast*), в RIP v2 — групповые (*multicast*).

Реализация RIP в данной версии NSG Linux имеет следующие ограничения:

— Несмежные маски подсетей (*non-sequential netmasks*), допускаемые RIP v2, не поддерживаются.

ВНИМАНИЕ Команды настройки RIP в NSG Linux в целом весьма близки к командному языку маршрутизаторов Cisco Systems, однако не тождественны им. Имеется ряд существенных отличий в формате параметров, значениях по умолчанию и т.п. Наиболее существенные отличия выделены ниже подчеркиванием.

ВНИМАНИЕ Прежде чем приступать к настройке службы RIP, необходимо убедиться, что она включена в меню системных служб:

```
(config-nsg)# services rip enable
```

Если она отключена, необходимо ввести указанную команду, сохранить конфигурацию и перезагрузить устройство; только после этого можно приступать к настройкам, описанным ниже. При остановке службы RIP командой

```
(config-nsg)# services rip disable
```

все параметры конфигурации, связанные с ней, утрачиваются при очередной команде write file и не могут быть восстановлены.

ВНИМАНИЕ Для работы протоколов динамической маршрутизации на интерфейсах "точка-точка" следует явно указывать в конфигурации интерфейса IP-адрес удалённой стороны, например:

```
!
nsg
    tunnel ip 1
        ip address 1.2.3.5/32 peer 1.2.3.6
        exit
```

ВНИМАНИЕ При использовании протокола RIP поверх туннеля GRE необходимо помнить, что пакеты RIP всегда отправляются с TTL=1, поэтому не проходят далее туннеля. Для нормальной работы RIP-over-GRE необходимо вручную установить для них бóльшее значение TTL, например, 64.

§4.5.1. Общие параметры службы RIP

Для активации/дезактивации службы RIP используются следующие команды в главном меню конфигурации Zebra (config)# :

router rip Активировать службу RIP (если она не активна) и перейти в меню ее настройки.
no router rip Дезактивировать службу RIP (но не останавливать ее полностью).

Дальнейшее управление службой RIP производится из меню (config-router)#. Для настройки службы RIP в целом предусмотрены следующие команды:

version { 1 | 2 }
Использовать RIP v1 или v2, если иное не указано для конкретного интерфейса. По умолчанию используется RIP v2.

network <интерфейс>
no network <интерфейс>
Разрешить/запретить рассылку и прием сообщений RIP на указанном IP-интерфейсе. Параметром команды является символическое имя интерфейса, например, eth0.

network <ip-префикс>
no network <ip-префикс>
Разрешить/запретить рассылку и прием сообщений RIP на IP-интерфейсе или интерфейсах, адреса которых попадают в указанный диапазон. IP-префикс состоит из адреса и маски. Например, команда
network 10.0.0.0/24
разрешает работу RIP на всех интерфейсах, имеющих адреса в диапазоне от 10.0.0.0 до 10.0.0.255.

ПРИМЕЧАНИЕ Интерфейс может иметь несколько IP-адресов, т.е. принадлежать одновременно к нескольким IP-подсетям. При этом сообщения RIP отправляются и принимаются независимо во всех подсетях, к которым относится данный интерфейс.

neighbor <ip-адрес>
no neighbor <ip-адрес>
Явно указать IP-адрес смежного маршрутизатора, которому должны посылаться сообщения RIP, либо исключить данный узел из списка соседей. Узлам, перечисленным в данном списке, обновления маршрутной таблицы рассылаются не в виде широковещательных сообщений, а в виде одноадресных. Для каждого из таких соседей генерируется индивидуальное сообщение.
Данная команда используется в двух случаях:
— если данный сосед не принимает широковещательных сообщений
— если интерфейс, к которому подключен данный сосед, находится в пассивном режиме (см. ниже)

timers basic <update> <timeout> <garbage>
no timers basic
Установка специфических значений таймеров протокола RIP (в секундах) и возврат к значениям по умолчанию. Протокол использует три таймера для рассылки сообщений и удаления недействительных маршрутов:
update Интервал рассылки сообщений RIP, содержащих полную таблицу маршрутизации данного узла.
timeout Время жизни маршрута. Если за это время не получено новое сообщение RIP, содержащее данный маршрут, то маршрут считается недействительным и ему присваивается метрика 16. Сам маршрут при этом формально сохраняется в таблице маршрутизации в течение времени <garbage> для того, чтобы можно было отправить соседним узлам сообщение об удалении маршрута.
garbage Время хранения недействительного маршрута. По истечении этого времени маршрут окончательно удаляется из локальной таблицы маршрутизации. Если за это время все-таки будет получено сообщение RIP, содержащее данный маршрут, процедура удаления прекращается и маршрут снова становится действующим.
Значения таймеров по умолчанию, предусмотренные RFC — 30, 180 и 120 сек, соответственно.

passive-intreface <интерфейс>
no passive-intreface <интерфейс>
Перевести указанный IP-интерфейс в пассивный режим и обратно. Если интерфейс находится в пассивном режиме, то маршрутизатор принимает сообщения RIP, поступающие на этот интерфейс, но не рассылает через него свои широковещательные и групповые сообщения. Единственный тип сообщений RIP, которые могут отправляться через такой интерфейс — это одноадресные сообщения соседям, явно указанным командой neighbor. Параметром команды является символическое имя интерфейса. По умолчанию пассивный режим выключен.

Пример. На интерфейсе eth0 включена служба RIP, установлен пассивный режим, и явно указан единственный узел в этой сети, которому следует посылать сообщения RIP:

```
!
nsg
    port eth0 ip address 10.11.21.33/24
!
router rip
    network 10.0.0.0/8
    passive-interface eth0
    neighbor 10.11.21.55
!
```

Такая конфигурация позволяет не засорять сеть широковещательными сообщениями.

Еще ряд параметров RIP настраивается индивидуально для каждого интерфейса в меню конфигурации интерфейса (config-if)#:

ip rip send version [1] [2]
: Посылать через данный интерфейс только пакеты RIP v1, или только RIP v2, или обеих версий. Данная команда имеет приоритет над общим режимом, который задается командой version. Если установлен режим 1 2, то рассылаются как групповые, так и широковещательные пакеты.

ip rip receive version [1] [2]
: Принимать на данном интерфейсе только пакеты RIP v1, или только RIP v2, или обеих версий. Данная команда имеет приоритет над общим режимом, который задается командой version.

ip split-horizon
no ip split-horizon
: Использовать/не использовать на данном интерфейсе метод "расщепления горизонта" (*split horizon*), чтобы предотвратить образование ложных маршрутов. По умолчанию этот метод включен на всех интерфейсах.

§4.5.2. Длина маршрутов

Длина маршрута, или расстояние, назначается административно и определяет приоритет записей в локальной таблице маршрутизации. Для маршрутов, полученных средствами протокола RIP, длина по умолчанию устанавливается равной 120. Это значение может быть изменено и избирательно настроено при помощи следующих команд в меню управления службой RIP (config-router)#:

distance <1…255>
no distance <1…255>
: Выбор и отмена иного значения административного расстояния по умолчанию для всех маршрутов, полученных по RIP.

distance <1…255> <ip-префикс>
no distance <1…255> <ip-префикс>
: Выбор и отмена иного значения административного расстояния для всех маршрутов, полученных от маршрутизаторов из указанной сети. IP-префикс состоит из адреса и длины маски подсети, например, 10.0.0.0/8. Если пакет RIP получен от источника, IP-адрес которого находится в указанном диапазоне, то для маршрута устанавливается административное расстояние в соответствии с данной командой.

distance <1…255> <ip-префикс> <access_list>
no distance <1…255> <ip-префикс> <access_list>
: Выбор и отмена иного значения административного расстояния в случае, если IP-адрес источника сообщения RIP соответствует указанному IP-префиксу, а маршрут — списку доступа. Последним аргументом команды может быть именованный или нумерованный список доступа Zebra.

ПРИМЕЧАНИЕ Из трех вышеперечисленных команд одновременно может действовать только одна, имеющая наиболее подробный список параметров.

§4.5.3. Фильтрация маршрутов

Входящие и исходящие сообщения RIP могут подвергаться фильтрации, в результате которой из них удаляются определенные маршруты. Фильтрация осуществляется на основе списков доступа (*access lists, prefix lists*) индивидуально для каждого интерфейса, на котором включен RIP. Настройка фильтров производится в два этапа:

— определение маршрутов, которые должны быть разрешены или запрещены в сообщениях RIP, при помощи соответствующего *access list* или *prefix list* (см. п.4.3)
— создание фильтра RIP, использующего данный *access list* или *prefix list*

Создание фильтров производится из меню управления службой RIP (config-router)# при помощи команд:

distribute-list <access_list> { in | out }
distribute-list <access_list> { in | out } <интерфейс>
Фильтровать маршруты в соответствии с указанным *access list*. Для работы фильтров используются как именованные, так и нумерованные списки доступа (см. п.4.3).
Второй параметр определяет направление фильтруемых сообщений:
in — входящие пакеты RIP
out — исходящие пакеты RIP
Третий параметр — символическое имя интерфейса, на котором действует данный фильтр. Если этот параметр отсутствует, фильтр применяется ко всем интерфейсам.

distribute-list prefix <prefix_list> { in | out }
distribute-list prefix <prefix_list> { in | out } <интерфейс>
Фильтровать маршруты в соответствии с указанным *prefix list* (см. п.4.3). Такой формат списков доступа, определенный в пакете Zebra и NSG Linux, предоставляет более гибкие возможности по сравнению с *access list*, а именно, позволяет сортировать маршруты по подсетям внутри заданной общей сети. В остальном команда аналогична предыдущей.

Для удаления созданных фильтров используются эти же команды с приставкой no. При этом списки доступа, связанные с ними, сохраняются и могут быть использованы для других целей.

Для каждого направления на каждом интерфейсе может быть задан только один *access list* и один *prefix list*. Кроме того, для каждого направления могут быть заданы один *access list* и один *prefix list*, действующие на все интерфейсы. Фильтры на каждом интерфейсе объединяются оператором "И", т.е. в пакетах RIP пропускаются только те маршруты, которые удовлетворяют всем фильтрам одновременно.

Пример. В следующей конфигурации через интерфейс eth0 будут приниматься только маршруты, ведущие в сеть 10.0.0.0/8:

```
!
router rip
    distribute-list private in eth0
!
access-list private permit 10.0.0.0/8
access-list private deny any
!
```

Остальные маршруты, получаемые в сообщениях RIP через этот интерфейс, будут проигнорированы.

§4.5.4. Рассылка импортированных маршрутов

По умолчанию, в сообщения протокола RIP включаются маршруты двух типов:
— маршруты в сети, непосредственно подключенные к интерфейсам, на которых включен RIP
— маршруты, полученные средствами протокола RIP

Рассылка маршрутов, полученных из других источников, определяется в меню конфигурации службы RIP (config-router)# при помощи следующих команд:

redistribute { kernel | static | connected | ospf }
no redistribute { kernel | static | connected | ospf }
Включать или не включать в сообщения RIP маршруты указанного происхождения. По умолчанию RIP рассылает только маршруты, полученные средствами самого этого протокола.

redistribute { kernel | static | connected | ospf } metric <0...16>
Для маршрутов данного типа указывать в сообщениях RIP данное значение метрики. Если такой формат команды не используется, то указывается значение метрики по умолчанию (см. п.4.5.5).

ПРИМЕЧАНИЕ В терминах протокола RIP метрика реального маршрута может иметь значение от 0 до 15. Значение 16 показывает, что выбранная сеть недоступна.

redistribute { kernel | static | connected | ospf } route-map <имя>
: При рассылке маршрутов данного типа использовать правила, установленные списком преобразования маршрутов (*route map*) с указанными именем. Подробнее об использовании списков преобразования маршрутов см. п.4.5.6.

ПРИМЕЧАНИЕ Команды redistribute connected ... влияют на рассылку маршрутов в сети, непосредственно подключенные к тем интерфейсам маршрутизатора, на которых протокол RIP отключен. Если протокол RIP включен на некотором интерфейсе, то маршрут в сеть, непосредственно подключенную к нему, содержится в сообщениях RIP всегда.

ПРИМЕЧАНИЕ Интерфейсы Async PPP и PPPoE всегда имеют маску длиной 32 бита (при условии, что им назначен IP-адрес). В результате сообщения RIP содержат маршруты только на сами эти интерфейсы. Чтобы включить в них маршруты к удаленным сторонам соединений PPP, необходимо использовать команду redistribute connected.

default-information originate
no default-information originate
: Включать или не включать в сообщения RIP маршруты по умолчанию (т.е. записи маршрутной таблицы с назначением 0.0.0.0/0). По умолчанию рассылка таких маршрутов не производится.

route <ip-префикс>
no route <ip-префикс>
: Команда, специфическая для пакета Zebra и NSG Linux. Создать/удалить фиктивный маршрут в некоторую сеть через данный маршрутизатор. Этот маршрут будет рассылаться соседям в сообщениях RIP, но не будет использоваться в действительности. Подобные манипуляции с протоколами не рекомендуется производить без хорошего понимания процедур маршрутизации и обмена маршрутной информацией.

§4.5.5. Метрика маршрутов

При внесении маршрутов, полученных средствами протокола RIP, в локальную таблицу маршрутизации значение метрики увеличивается на единицу. Это же значение указывается для них в исходящих сообщениях. Для всех маршрутов, полученных иными средствами, в сообщениях RIP указывается, по умолчанию, метрика 1. Эти значения могут быть изменены и избирательно настроены несколькими способами:

— командами redistribute ... metric <0...16>
— командами redistribute ... route-map <имя> и соответствующими правилами преобразования маршрутов (*route map*, см. п.4.5.6)
— командами настройки метрики

В последнем случае используются следующие команды в меню управления службой RIP (config-router)#:

default-metric <1...16>
no default-metric <1...16>
: Выбор и отмена иного значения метрики по умолчанию для маршрутов, импортируемых в RIP из других источников. Данное значение используется, если метрика данного маршрута или категории маршрутов не назначена явно первыми двумя способами.

offset-list <access_list> { in | out } <0...16> [<интерфейс>]
no offset-list <access_list> { in | out } <0...16> [<интерфейс>]
: Установить и отменить приращение метрики для всех маршрутов, соответствующих указанному *access- list*, на всех интерфейсах или на одном указанном интерфейсе. Первым аргументом команды может быть именованный или нумерованный *access list* (см. п.4.3).
Второй аргумент определяет, в какой момент осуществляется увеличение метрики:
in — при внесении маршрутов из входящих сообщений RIP в таблицу маршрутизации
out— при генерации исходящих сообщений RIP на основе текущей таблицы маршрутизации
Третий аргумент — положительное значение, на которое следует увеличить метрику этих маршрутов (0 — оставить метрику без изменений).
Последним аргументом является символическое имя интерфейса; если имя не указано, команда относится ко всем интерфейсам.

Для каждого направления может быть установлено только одно правило *offset list* для каждого интерфейса и одно общее правило (без указания интерфейса). Общее правило действует на все те интерфейсы, для которых не установлено индивидуальных правил в данном направлении.

ПРИМЕЧАНИЕ Приращение, установленное командой offset-list, не отменяет значений метрики, установленных другими способами, а прибавляется к ним. Пример:

```
!
port s1 ip address 11.0.0.1/8
ip route 12.0.0.0/8 11.0.0.2
access-list test permit any
router rip
    redistribute static metric 3
    network eth0
    network s1
    offset-list test out 2
!
```

При такой конфигурации маршрут в непосредственно подключенную сеть 11.0.0.0/8 будет рассылаться с метрикой 3, а статический маршрут в сеть 12.0.0.0/8 — с метрикой 5 вместо 1. Аналогично, в следующей конфигурации:

```
!
access-list test permit any
router rip
    offset-list test in 2
!
```

при внесении любых маршрутов, полученных по RIP, в таблицу маршрутизации их метрика будет увеличиваться не на 1, а на 3.

ПРИМЕЧАНИЕ Приоритет записей в локальной таблице маршрутизации на устройстве NSG определяется административной длиной маршрута, а не его метрикой. Величина метрики используется только в самих протоколах динамической маршрутизации и определяет работу протокола в масштабах сети как целого.

§4.5.6. Правила преобразования маршрутов (*route maps*)

При рассылке маршрутов, импортированных из других источников, некоторые поля сообщений RIP могут быть установлены административно. Для этого необходимо определить правила преобразования маршрутов (*route maps*) и использовать команду redistribute в следующем формате:

```
redistribute { kernel | static | connected | ospf } route-map <имя>
```

где последним параметром является имя *route map*. Подробное описание *route maps* приведено в п.4.4. Для создания *route map* используется команда

```
route-map <имя> { permit | deny } <приоритет>
```

в главном меню конфигурации Zebra (config)# . Далее в меню (config-route-map)# определяются критерии, по которым следует проверять маршруты (команды match) и действия, которые следует предпринять (команды set). Используются общие команды match и set (см. п.4.4):

```
match interface <имя>
match ip address <access_list>
match ip address prefix-list <prefix_list>
match metric <0…16>
set metric <0…16>
```

ПРИМЕЧАНИЕ Из соображений совместимости с другими протоколами, параметр команд match metric и set metric может принимать значения до 4294967195; однако применительно к протоколу RIP значения больше 16 не имеют смысла.

Кроме того, применительно к протоколу RIP, имеют смысл следующие специфические команды:

match ip next-hop <ip-адрес>
: Следующим узлом на маршруте (*next hop*) должен быть узел с указанным IP-адресом. Подразумевается, что данный узел значится в таблице маршрутизации RIP (см. команду show ip rip). В данной реализации команда допускает указание только одного следующего узла.

set ip next-hop <ip-адрес>
: Установить указанный IP-адрес в качестве следующего узла в исходящих пакетах RIP v2. В протоколе RIP v1 поле *next hop* не поддерживается.

Если запись, под которую подпадает данный маршрут, не содержит инструкцию set metric, то для него устанавливается значение метрики RIP, заданное командой default-metric (либо единица — по умолчанию). В случае, если параметр route-map использован одновременно с параметром metric следующим образом:

```
redistribute { kernel | static | connected | ospf }  metric <1…16>  route-map <имя>
```

то в протокол RIP импортируются только те маршруты, которые допускаются указанным правилом. При этом, если соответствующая запись содержит инструкцию set metric, то она выполняется; если такая инструкция отсутствует, то для маршрута устанавливается метрика, указанная общим ключом metric.

ПРИМЕЧАНИЕ В отличие от реализации Cisco, правила *route map* применяются в NSG Linux не при внесении маршрутов, полученных по RIP, в локальную таблицу маршрутизации, а при генерации исходящих сообщений RIP на основе имеющейся таблицы.

§4.5.7. Аутентификация RIP

Протокол RIP v2 предусматривает, дополнительно к обмену маршрутной информацией, аутентификацию источника этой информации. Аутентификация может выполняться двумя способами — с помощью простого текстового пароля либо алгоритма MD5 — либо быть отключена. Настройка аутентификации производится индивидуально для каждого интерфейса, на котором работает RIP, в соответствующем меню (config-if)#.

ip rip authentication mode { text | md5 }
no ip rip authentication mode { text | md5 }
Включить/выключить один из режимов аутентификации на интерфейсе.

ip rip authentication string <пароль>
no ip rip authentication string <пароль>
Установить/отменить пароль для простой текстовой аутентификации. Максимальная длина пароля — 15 символов.

ip rip authentication key-chain <последовательность>
no ip rip authentication key-chain <последовательность>
Установить/отменить набор ключей (*key chain*) для аутентификации на основе MD5. Сам набор определяется отдельной командой key chain (см. п.4.5.8).

Пример конфигурации:

```
!
key chain test
    key 1
        key-string proba
!
interface eth0
    ip rip authentication mode md5
    ip rip authentication key-chain test
!
```

§4.5.8. Ключи для аутентификации MD5

При аутентификации сообщений RIP на основе MD5 используются наборы ключей (*key chains*). Каждый набор определяется уникальным именем и может содержать несколько ключей с различными номерами и содержимым, с ограниченным или неограниченным временем действия. Для работы алгоритма MD5 используется ключ с наименьшим номером из числа действительных на данный момент.

Для управления ключами используются следующие команды:

— в главном меню конфигурации Zebra (config)# :

key chain <имя>
no key chain <имя>
Создать/изменить и удалить набор ключей с данным именем. Первая команда создает набор, если он не существует, и входит в режим его редактирования.

— в меню конфигурации набора ключей (config-keychain)# :

key <0...2147483647>
no key <0...2147483647>
Создать/изменить и удалить ключ с данным номером. Первая команда создает ключ, если он не существует, и входит в режим его редактирования. Номер ключа должен быть уникален в пределах одного набора; ключи, принадлежащие к разным наборам, могут иметь одинаковые номера.

— в меню конфигурации ключа (config-keychain-key)# :

key-string <строка>
no key-string [<строка>]
Создать тело данного ключа, или сделать данный ключ пустым.

accept-lifetime <начало> { <конец> | duration <секунды> | infinite }
send-lifetime <начало> { <конец> | duration <секунды> | infinite }

Интервалы времени, в течение которых разрешается использовать данный ключ для входящих и исходящих сообщений RIP, соответственно. Время начала действия ключа указывается в одном из следующих форматов:

hh:mm:ss dd mon yyyy
hh:mm:ss mon dd yyyy

где mon — первые три буквы месяца (в любом регистре), остальные параметры числовые. Время окончания действия ключа указывается одним из трех способов:
— явным образом в таком же формате, как и время начала
— в виде срока жизни ключа (duration), в секундах
— срок жизни неограничен (infinite)
По умолчанию все создаваемые ключи имеют неограниченный срок действия как на прием, так и на передачу.
Для отмены ограничений на действие ключа используются эти же команды, но не с префиксом no, а с достаточно ранним временем начала (например, раньше текущего времени) и неограниченным (infinite) сроком жизни.

ПРИМЕЧАНИЕ Для устойчивой работы протокола рекомендуется синхронизировать системное время на всех узлах при помощи служб NTP и/или SNTP, а также назначать время действия последовательных ключей с небольшим перекрытием. (Например, в пределах получаса действуют как старый, так и новый ключи.)

ПРИМЕЧАНИЕ Установка системного времени и даты, а также настройка клиента SNTP в данной версии NSG Linux производятся только средствами командной оболочки Linux. Поддержка их в основной командной оболочке системы планируется в последующих версиях.

§4.5.9. Просмотр информации RIP

Для просмотра информации о работе протокола RIP и ее результатах используются следующие команды show в меню обычного или привилегированного режима:

show ip rip

Вывести список маршрутов, полученных по RIP, а также список маршрутов, импортированных в службу RIP из других протоколов. Для маршрутов, полученных по RIP, указывается также время получения последнего пакета с этим маршрутом, и тег маршрута.

show ip protocols

Вывести информацию о состоянии службы RIP, включая настройки таймеров, фильтров, список активных интерфейсов, версии протокола и сведения о смежных узлах.

Пример вывода:

```
nsg> show ip protocols
Routing Protocol is "rip"
    Sending updates every 30 seconds with +/-50%, next due in 35 seconds
    Timeout after 180 seconds, garbage collect after 120 seconds
    Outgoing update filter list for all interface is not set
    Incoming update filter list for all interface is not set
    Default redistribution metric is 1
    Redistributing: kernel connected
    Default version control: send version 2, receive version 2
        Interface  Send  Recv
    Routing for Networks:
        eth0
        eth1
        1.1.1.1
        203.181.89.241
    Routing Information Sources:
        Gateway          BadPackets  BadRoutes  Distance  Last Update
        ...........      ...........  ...........  ...........  ...........
```

§4.5.10. Отладка RIP

Для отладки службы RIP предусмотрены следующие команды в меню привилегированного режима:

debug rip events
: Выводить информацию о событиях протокола RIP, включая прием и передачу пакетов, срабатывание таймеров и изменения в состояниях интерфейсов.

debug rip packet [send | recv]
: Выводить информацию о пакетах RIP: адрес источника, номер порта, дамп пакета. Необязательные ключи send и recv позволяют избирательно выводить только отправляемые или только принятые пакеты.

debug rip zebra
: Выводить информацию о взаимодействии служб RIP и Zebra. Для отладки протокола RIP основной интерес здесь представляет информация о добавлении и удалении маршрутов в ядре, а также об обмене сведениями о состоянии интерфейсов.

Для отмены вывода отладочной информации используются эти же команды с префиксом no.

show debugging rip
: Вывести информацию о текущих настройках отладчика RIP.

Вся выводимая информация направляется в файл, заданный командой log в главном меню конфигурации Zebra (config)#, например:

```
nsg(config)# log file /var/rip.log
```

Просмотреть содержимое данного файла можно средствами ОС Linux, например:

```
nsg# start-shell

BusyBox v0.60.5 (2003.08.25-14:15+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# cat /var/rip.log
2004/06/25 17:58:19 RIP: RECV packet from 11.0.0.1 port 520 on eth0
2004/06/25 17:58:20 RIP: update timer fire!
2004/06/25 17:58:20 RIP: SEND UPDATE to eth0 ifindex 3
2004/06/25 17:58:20 RIP: multicast announce on eth0
2004/06/25 17:58:21 RIP: update routes on interface eth0 ifindex 3
2004/06/25 17:58:21 RIP: SEND to socket 10 port 520 addr 224.0.0.9
2004/06/25 17:58:21 RIP: SEND UPDATE to s2 ifindex 14
2004/06/25 17:58:21 RIP: multicast announce on s2
2004/06/25 17:58:21 RIP: update routes on interface s2 ifindex 14
2004/06/25 17:58:21 RIP: SEND to socket 10 port 520 addr 224.0.0.9
.....
```

В данном случае можно видеть, что маршрутизатор получил один пакет RIP от узла 11.0.0.1, а также подготовил и отправил два пакета с групповыми адресами через интерфейсы eth0 и s2.

§4.6. Протокол OSPF

Протокол OSPF относится к внутренним протоколам маршрутизации (Interior Gateway Protocols, IGP). По сравнению с протоколом RIP, OSPF обеспечивает поддержку крупномасштабных сетей и более быструю сходимость. OSPF широко применяется в больших сетях, таких как магистральные сети поставщиков услуг и корпоративные сети.

ВНИМАНИЕ Прежде чем приступать к настройке службы OSPF, необходимо убедиться, что она включена в меню системных служб:

```
(config-nsg)# services ospf enable
```

Если она отключена, необходимо ввести указанную команду, сохранить конфигурацию и перезагрузить устройство; только после этого можно приступать к настройкам, описанным ниже. При остановке службы OSPF командой

```
(config-nsg)# services ospf disable
```

все параметры конфигурации, связанные с ней, утрачиваются при очередной команде write file и не могут быть восстановлены.

ВНИМАНИЕ Для работы OSPF необходимо корректно установить системное время на всех устройствах. На устройствах NSG–700 первых выпусков, не оборудованных часами (легко определяются по отсутствию батарейки на материнской плате) следует использовать службу NTP (см. п.4.12.5).

ВНИМАНИЕ При использовании протокола OSPF поверх туннеля GRE необходимо помнить, что пакеты OSPF всегда отправляются с TTL=1, поэтому не проходят далее туннеля. Для нормальной работы OSPF-over-GRE необходимо вручную установить для них бóльшее значение TTL, например, 64.

§4.6.1. Общие параметры службы OSPF

Для активации/дезактивации службы OSPF используются следующие команды в главном меню конфигурации Zebra (config)# :

router ospf

Активировать службу OSPF (если она не активна) и перейти в меню ее настройки.

no router ospf

Дезактивировать службу OSPF (но не останавливать ее полностью).

ПРИМЕЧАНИЕ В NSG Linux поддерживается только один процесс OSPF, поэтому номер процесса не указывается.

Дальнейшее управление службой OSPF производится из меню (config-router)#. Для настройки службы OSPF в целом предусмотрены следующие команды:

ospf router-id a.b.c.d
no ospf router-id

ospf abr-type { cisco | ibm | shortcut | standard }
no ospf abr-type { cisco | ibm | shortcut | standard }

ospf rfc1583compatibility
no ospf rfc1583compatibility

passive interface <имя>
no passive interface <имя>

timers spf <0…4294967295> <0…4294967295>
no timers spf

refresh group-limit <0…10000>
refresh per-slice <0…10000>
refresh age-diff <0…10000>

auto-cost refrence-bandwidth <1…4294967>
no auto-cost refrence-bandwidth

network *a.b.c.d/m* area *a.b.c.d*
network *a.b.c.d/m* area <0…4294967295>
no network *a.b.c.d/m* area *a.b.c.d*
no network *a.b.c.d/m* area <0…4294967295>

Разрешить/запретить работу протокола OSPF на IP-интерфейсе, имеющем указанный IP-префикс. Например, если интерфейс имеет префикс 10.0.0.1/8, то следующая команда делает его "видимым" для OSPF-маршрутизаторов:

```
router ospf
    network 10.0.0.0/8 area 0
```

ВНИМАНИЕ Для широковещательных интерфейсов длина маски, указанная в префиксе команды network и в префиксе интерфейса, должна быть строго одинаковой.
Для работы OSPF на интерфейсах "точка-точка" следует явно указывать в конфигурации интерфейса IP-адрес удалённой стороны. Этот же адрес указывается в качестве сети, например:

```
!
nsg
    tunnel ip 1
        ip address 1.2.3.5/32 peer 1.2.3.6
        exit
    exit
!
router ospf
    network 1.2.3.6/32 area 0
!
```

ВНИМАНИЕ Для корректной работы OSPF необходимо установить одинаковый размер MTU на всех смежных интерфейсах командой ip mtu. Если размер MTU устанавливается автоматически, то следует проверить установленные значения при помощи команд show (в меню порта) или show interface <имя> в корневом меню.

§4.6.2. Определение зоны OSPF (*OSPF area*)

Для определения зоны OSPF используются следующие команды в меню (config-router)# :

area *a.b.c.d* range *a.b.c.d/m*
area <0…4294967295> range *a.b.c.d/m*
no area *a.b.c.d* range *a.b.c.d/m*
no area <0…4294967295> range *a.b.c.d/m*

area *a.b.c.d* range *IPV4_PREFIX* suppress
no area *a.b.c.d* range *IPV4_PREFIX* suppress
area *a.b.c.d* range *IPV4_PREFIX* substitute *IPV4_PREFIX*
no area *a.b.c.d* range *IPV4_PREFIX* substitute *IPV4_PREFIX*

area *a.b.c.d* virtual-link *a.b.c.d*
area <0…4294967295> virtual-link *a.b.c.d*
no area *a.b.c.d* virtual-link *a.b.c.d*
no area <0…4294967295> virtual-link *a.b.c.d*

area *a.b.c.d* shortcut
area <0…4294967295> shortcut
no area *a.b.c.d* shortcut
no area <0…4294967295> shortcut

area *a.b.c.d* stub
area <0…4294967295> stub
no area *a.b.c.d* stub
no area <0…4294967295> stub

area *a.b.c.d* stub no-summary
area <0…4294967295> stub no-summary
no area *a.b.c.d* stub no-summary
no area <0…4294967295> stub no-summary

area *a.b.c.d* default-cost <0…16777215>
no area *a.b.c.d* default-cost <0…16777215>

area *a.b.c.d* export-list *NAME*
area <0…4294967295> export-list *NAME*
no area *a.b.c.d* export-list *NAME*
no area <0…4294967295> export-list *NAME*

area *a.b.c.d* import-list *NAME*
area <0…4294967295> import-list *NAME*
no area *a.b.c.d* import-list *NAME*
no area <0…4294967295> import-list *NAME*

area *a.b.c.d* authentication
area <0…4294967295> authentication
no area *a.b.c.d* authentication
no area <0…4294967295> authentication

area *a.b.c.d* authentication message-digest
area <0…4294967295> authentication message-digest

§4.6.3. Настройка OSPF на интерфейсах

Для настройки параметров OSPF на отдельном интерфейсе используются следующие команды в меню интерфейса (config-if)# :

ip ospf authentication-key <ключ>
no ip ospf authentication-key

Назначить/удалить простой текстовый ключ для аутентификации сообщений OSPF. Максимальная длина ключа — 8 символов.

ip ospf message-digest-key <номер> md5 <ключ>
no ip ospf message-digest-key

Назначить/удалить криптографический пароль для аутентификации сообщений OSPF с использованием алгоритма MD5. Параметры команды:

<номер> идентификатор ключа
<ключ> собственно ключ длиной до 16 символов

ip ospf cost <1...65535>
no ip ospf cost

Установить/удалить стоимость соединения для данного интерфейса. Данный параметр помещается в поле метрики сообщений LSA (*Link State Advertisement*) и используется для вычисления кратчайшего пути.

ip ospf dead-interval <1...65535>
no ip ospf dead-interval

Значение интервала *RouterDeadInterval*, используемого в таймерах ожидания (*Wait Timer*) и неактивности (*Inactivity Timer*), в секундах. Данная величина должна быть одинакова во всех маршрутизаторах, подключенных к одной сети. Значение по умолчанию — 40 сек.

ip ospf hello-interval <1...65535>
no ip ospf hello-interval

Значение интервала *HelloInterval*, в секундах. Таймер определяет интервал между посылкой пакетов *Hello* через данный интерфейс. Данная величина должна быть одинакова во всех маршрутизаторах, подключенных к одной сети. Значение по умолчанию — 10 сек.

ip ospf network { broadcast | non-broadcast | point-to-multipoint | point-to-point }
no ip ospf network

Явно указать тип сети, подключенной к данному интерфейсу, или удалить это указание.

ip ospf priority <0-255>
no ip ospf priority

Приоритет данного маршрутизатора в системе на основе OSPF. Чем выше данное значение, тем больше вероятность для данного маршрутизатора стать ведущим (*Designated Router*). Специальное значение 0 показывает, что данный маршрутизатор не должен становиться ведущим. Значение по умолчанию — 1.

ip ospf retransmit-interval <1...65535>
no ip ospf retransmit interval

Значение интервала *RxmtInterval*, в секундах. Данный таймер определяет интервал между посылкой пакетов с описанием базы данных (*Database Description*) и запросом о состоянии линии (*Link State Request*). Значение по умолчанию — 5 сек.

ip ospf transmit-delay
no ip ospf transmit-delay

Значение интервала *InfTransDelay*, в секундах. При передаче пакетов OSPF "возраст" сообщений LSA должен быть увеличен на указанное значение. Значение по умолчанию — 1 сек.

§4.6.4. Импорт маршрутов из других протоколов

При импорте маршрутов, полученных из других источников, в протокол OSPF производится назначение атрибутов, специфических для данного протокола. Правила преобразования таких маршрутов в маршруты OSPF устанавливаются следующими командами в меню (config-router)# :

redistribute { kernel | connected | static | rip }
redistribute { kernel | connected | static | rip } *route-map*
redistribute { kernel | connected | static | rip } metric-type { 1 | 2 }
redistribute { kernel | connected | static | rip } metric-type { 1 | 2 } route-map <имя>
redistribute { kernel | connected | static | rip } metric <0…16777214>
redistribute { kernel | connected | static | rip } metric <0…16777214> route-map <имя>
redistribute { kernel | connected | static | rip } metric-type { 1 | 2 } metric <0…16777214>
redistribute { kernel | connected | static | rip } metric-type { 1 | 2 } metric <0…16777214> route-map <имя>
no redistribute { kernel | connected | static | rip }

default-information originate
default-information originate metric <0…16777214>
default-information originate metric <0…16777214> metric-type { 1 | 2 }
default-information originate metric <0…16777214> metric-type { 1 | 2 } route-map <имя>
default-information originate always
default-information originate always metric <0…16777214>
default-information originate always metric <0…16777214> metric-type { 1 | 2 }
default-information originate always metric <0…16777214> metric-type { 1 | 2 } route-map <имя>
no default-information originate

distribute-list <имя> out { kernel | connected | static | rip | ospf }
no distribute-list <имя> out { kernel | connected | static | rip | ospf }

default-metric <0…16777214>
no default-metric

distance <1…255>
no distance <1…255>

distance ospf { intra-area | inter-area | external } <1…255>
no distance ospf

Следующая команда используется в меню (config)# :

router zebra
no router zebra

§4.6.5. Просмотр информации OSPF

Для просмотра информации о работе протокола OSPF и ее результатах используются следующие команды show в меню обычного или привилегированного режима:

show ip ospf
show ip ospf interface [*INTERFACE*]
show ip ospf neighbor
show ip ospf neighbor *INTERFACE*
show ip ospf neighbor detail
show ip ospf neighbor *INTERFACE* detail

show ip ospf database

show ip ospf database { asbr-summary | external | network | router | summary }
show ip ospf database { asbr-summary | external | network | router | summary } *link-state-id*
show ip ospf database { asbr-summary | external | network | router | summary } *link-state-id* adv-router *adv-router*
show ip ospf database { asbr-summary | external | network | router | summary } adv-router *adv-router*
show ip ospf database { asbr-summary | external | network | router | summary } *link-state-id* self-originate
show ip ospf database { asbr-summary | external | network | router | summary } self-originate

show ip ospf database max-age

show ip ospf database self-originate

show ip ospf refresher

show ip ospf route

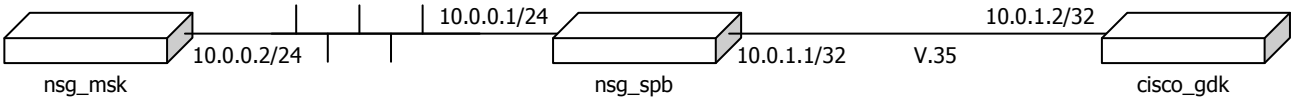
§4.6.6. Отладка OSPF

Для отладки службы OSPF предусмотрены следующие команды в меню привилегированного режима:

```
debug ospf packet { hello | dd | ls-request | ls-update | ls-ack | all } { send | recv } [detail]
no debug ospf packet { hello | dd | ls-request | ls-update | ls-ack | all } { send | recv } [detail]

debug ospf ism
debug ospf ism { status | events | timers }
no debug ospf ism
no debug ospf ism { status | events | timers }

debug ospf nsm
debug ospf nsm { status | events | timers }
no debug ospf nsm
no debug ospf nsm { status | events | timers }

debug ospf lsa
debug ospf lsa { generate |
no debug ospf lsa
no debug ospf lsa { generate |

debug ospf zebra
debug ospf zebra { interface | redistribute }
no debug ospf zebra
no debug ospf zebra { interface | redistribute }

show debugging ospf
```

§4.6.7. Пример настройки OSPF

Имеется цепочка из 3 устройств. Маршрутизатор nsg_spb находится в центре сети. По одну сторону от него находится устройство nsg_msk, подключенное по сети Ethernet. По другую сторону — устройство cisco_gdk, подключённое по синхронному каналу V.35.



Протокол Cisco-HDLC и скорость в синхронном порту 64000 на обоих устройствах nsg_spb и cisco_gdk установлены по умолчанию, поэтому в конфигурации они не отражаются. Конфигурации устройств:

```
!
nsg
    port eth0
        ip address 10.0.0.2/24
        exit
!
router ospf
    ospf router-id 10.0.0.2
    network 10.0.0.0/24 area 0
!
```

```
!
nsg
    port eth0
        ip address 10.0.0.1/24
        exit
    card s1 im-v35
    port s1
        mode internal
        ip address 10.0.1.1/32 peer 10.0.1.2
        exit
!
interface s1.0
    ip ospf network point-to-point
!
router ospf
    ospf router-id 10.0.1.1
    redistribute connected
    network 10.0.0.0/24 area 0
    network 10.0.1.2/32 area 0
!
```

```
!
interface Serial0/1
    ip address 10.0.1.2 255.255.255.0
    ip ospf network point-to-point
    no dce-terminal-timing-enable
!
router ospf 1
    network 10.0.1.0 0.0.0.255 area 0
!
```

В результате оба крайних устройства имеют маршруты друг на друга:

```
cisco_gdk#show ip route
.....................
C   10.0.1.0/24 is directly connected, Serial0/1
O   10.0.0.0/24 [110/65] via 10.0.1.1, 00:21:12, Serial0/1
```

§4.7. Протокол BGP

Данная глава находится в разработке. Временно рекомендуется пользоваться следующим документом:

Kunihiro Ishiguro. GNU Zebra. A routing software package for TCP/IP networks, Chapter 7

Подробное описание команд BGP будет представлено в последующих версиях данного Руководства.

ВНИМАНИЕ Прежде чем приступать к настройке службы BGP, необходимо убедиться, что она включена в меню системных служб:

```
(config-nsg)# services bgp enable
```

Если она отключена, необходимо ввести указанную команду, сохранить конфигурацию и перезагрузить устройство; только после этого можно приступать к настройкам, описанным ниже. При остановке службы BGP командой

```
(config-nsg)# services bgp disable
```

все параметры конфигурации, связанные с ней, утрачиваются при очередной команде write file и не могут быть восстановлены.

ВНИМАНИЕ Для работы протоколов динамической маршрутизации на интерфейсах "точка-точка" следует явно указывать в конфигурации интерфейса IP-адрес удалённой стороны, например:

```
!
nsg
    tunnel ip 1
        ip address 1.2.3.5/32 peer 1.2.3.6
        exit
```

ВНИМАНИЕ При использовании протокола BGP поверх туннеля GRE необходимо помнить, что пакеты BGP всегда отправляются с TTL=1, поэтому не проходят далее туннеля. Для нормальной работы BGP-over-GRE необходимо вручную установить для них бóльшее значение TTL, например, 64.

§4.8. Трансляция сетевых адресов и портов. Коммутация пакетов IP.

§4.8.1. Общие сведения о процедурах NAT и коммутации пакетов IP

Механизм трансляции сетевых адресов и портов (Network Address Translation, NAT) предназначен для преобразования IP-адресов и номеров портов TCP/UDP между сетью, подключенной к данному IP-интерфейсу маршрутизатора, и всеми остальными IP-сетями, подключенными к маршрутизатору. Как правило, он используется для подключения некоторой ограниченной подсети, адресное пространство которой не соответствует глобальному распределению IP-адресов, ко всему внешнему миру, использующему единое иерархическое адресное пространство. Соответственно, две сети, между которыми производится трансляция адресов, называются обычно *внутренней* и *внешней* (хотя в общем случае это могут быть и две сети с глобальными IP-адресами, и две сети с приватными IP-адресами).

ПРИМЕЧАНИЕ Следующие диапазоны IP-адресов выделены исключительно для использования в корпоративных сетях, изолированных от глобальной сети Интернет посредством NAT:

в классе A — 10.0.0.0 ... 10.255.255.255
в классе B — 172.16.0.0 ... 172.31.255.255
в классе C — 192.168.0.0 ... 192.168.255.255

Эти адреса и сети называются приватными (*private*), локальными или "серыми", в отличие от глобальных или "белых" адресов, распределяемых централизованно.

Именно эти адреса желательно использовать при построении внутренних сетей. Напротив, в глобальной сети Интернет такие адреса использоваться не могут, а любой пакет, приходящий извне с таким адресом источника или назначения, должен уничтожаться — это либо результат ошибки в конфигурации чьего-то маршрутизатора, либо попытка злонамеренного вмешательства в работу данной сети.

Понятие NAT является собирательным и включает в себя целый ряд различных алгоритмов, по которым производится преобразование IP-адресов и портов при обращении из внутренней сети во внешнюю, из внешней во внутреннюю, или в обоих направлениях. Использование трансляции сетевых адресов решает три важные задачи:

— Позволяет использовать одни и те же IP-адреса во многих подсетях, поскольку адресное пространство каждой внутренней подсети является локальным, а не частью глобального адресного пространства всего Интернет. Уникальными должны быть только внешние IP-адреса, назначенные интерфейсу с поддержкой NAT; эти адреса распределяются централизованно поставщиками услуг Интернет. Во внутренней сети могут использоваться любые адреса, назначенные администратором или, например, унаследованные в ходе какой-либо реструктуризации сети.
— Устраняет нехватку глобальных IP-адресов, присущую протоколу IPv4. За одним или несколькими глобальными адресами может стоять намного большее число хостов внутренней сети.
— Скрывает истинную структуру внутренней сети от внешнего мира и тем самым повышает ее безопасность.

Преобразование адресов, а также коммутация IP-пакетов, могут применяться как ко всему потоку трафика, проходящему через интерфейс, так и избирательно к некоторым определенным пакетам (при подключении соответствующего *access list*). Данная версия NSG Linux поддерживает следующие основные механизмы, доступные через основную командную оболочку:

— Преобразование адреса источника в статически заданный IP-адрес или диапазон адресов (*Source NAT, SNAT*).
— Наиболее важный частный случай Source NAT — преобразование адреса источника в IP-адрес того интерфейса, через который отсылается данный пакет (*IP Masquerading*). В частности, адрес выходного интерфейса может назначаться динамически.
— Преобразование адреса назначения в статически заданный IP-адрес или диапазон адресов (*Destination NAT, DNAT*, или *виртуальные сервера*). Для протоколов TCP или UDP может быть преобразован также номер порта назначения.
— Коммутацию или копирование пакета на заданный выходной интерфейс и/или шлюз (*Switching*).

Другие алгоритмы преобразования IP-адресов и портов могут быть реализованы средствами непосредственно ОС Linux, а именно, пакета IPtables. В принципе, доступны все возможности IPtables. Настраивать их из основной командной оболочки NSG Linux возможно, используя механизм сценариев (см. Часть 6).

Механизм NAT работает и настраивается независимо для каждого IP-интерфейса. Включение и выключение NAT производится в меню физического порта, Frame Relay DLCI, либо Ethernet VLAN в зависимости от того, какая инкапсуляция и физическая среда передачи расположены под требуемым IP-интерфейсом. Одновременно на одном интерфейсе могут быть определены несколько алгоритмов.

ВНИМАНИЕ Механизмы NAT всегда следует включать на интерфейсе, обращенном во *внешнюю* сеть.

ВНИМАНИЕ При первом запуске NAT или IP-коммутации подгружаются необходимые модули ядра Linux. При этом производительность устройства снижается (в силу более сложной программной обработки). Удаление правил NAT само по себе не приводит к выгрузке этих модулей. Чтобы избавиться от них и восстановить исходную производительность, необходимо сохранить полученную конфигурацию и перезагрузить устройство в целом.

§4.8.2. Потоки трафика

Ключевым понятием для работы механизма NAT является *поток*. В общем случае этим термином обозначается некоторая совокупность взаимосвязанных пакетов. Для каждого потока создается соответствующая запись в таблице NAT, непосредственно определяющая уникальное сочетание адресов и портов источника и назначения. Строгая интерпретация этого понятия зависит от типа трафика. Поток имеет ограниченное время жизни, по истечении которого запись из таблицы NAT удаляется. Подробное описание механизма идентификации потоков в Linux содержится в документе:

Oskar Andreasson. Iptables Tutorial 1.2.0. Chapter 7. The state machine.
http://iptables-tutorial.frozentux.net/iptables-tutorial.html#STATEMACHINE

Для пакетов TCP поток определяется наиболее просто, поскольку данный протокол по природе своей ориентирован на соединения. Потоком считается вся последовательность отправленных и полученных пакетов, относящихся к одному TCP-соединению, начиная от трехшаговой процедуры установления соединения (SYN — SYN/ACK — ACK) и заканчивая разрывом соединения. Между этими двумя процедурами соединение находится в состоянии ESTABLISHED и по нему передаются данные. Время жизни соединения зависит от текущей фазы соединения, в частности:

— для установленного соединения — 5 суток
— для соединения после нормального завершения (процедура FIN — ACK) — 2 минуты (чтобы пропустить все остаточные пакеты, которые могли задержаться на промежуточных узлах)
— для соединения после аварийного завершения (RST) — 10 секунд

После прохождения очередного пакета отсчет времени жизни соединения начинается заново. Полный список таймаутов для различных фаз TCP-соединения, установленных по умолчанию, приведен в вышеупомянутом документе. При необходимости эти значения могут быть изменены средствами ОС Linux.

Для пакетов UDP принадлежность пакетов к тому или иному потоку определяется на основе сочетания адресов и номеров портов. Время жизни потока после посылки первого пакета (т.е. пакета, который не ассоциируется ни с одним существующим потоком) — 30 секунд. Если за это время получен один ответный пакет, то поток считается установленным и время жизни увеличивается до 180 секунд. Если в течение 180 секунд не принято и не получено ни одного пакета, относящегося к данному потоку, запись о потоке уничтожается.

Для пакетов ICMP понятие потока применяется только к типам пакетов, подразумевающим парное взаимодействие Request/Reply (например, Echo Request / Echo Reply). После посылки пакета Request время ожидания ответа — 30 секунд. После получения соответствующего пакета Reply запись о потоке уничтожается, поскольку никаких последующих пакетов не предполагается.

Смешанные потоки TCP/ICMP или UDP/ICMP могут возникать, например, когда в ответ на попытку установления TCP-соединения или попытку отправки UDP-датаграммы приходит пакет ICMP Host Unreachable или Network Unreachable. Такие пакеты учитываются в рамках ранее открытого потока TCP или UDP, соответственно, и перенаправляются соответствующему хосту — инициатору потока. После этого запись о потоке уничтожается.

Для пакетов остальных протоколов транспортного подуровня, работающих поверх IP (например, для NETBLT, MUX или EGP) используется механизм, аналогичный UDP. Время жизни потока по умолчанию — 600 секунд.

Сложные протоколы прикладного уровня подразумевают или допускают использование одновременно нескольких портов TCP или UDP. К таким протоколам относятся FTP, TFTP, IRC и Amanda. Например, использует порты TCP 21 (ftp) и 20 (ftp-data).

§4.8.3. Source NAT и IP-маскарадинг

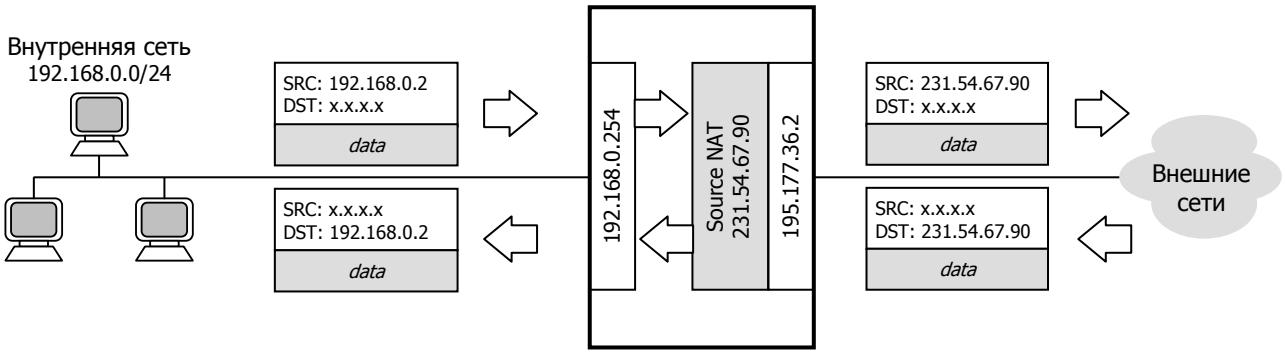
Трансляция адреса источника позволяет нескольким хостам внутренней сети выходить во внешнюю сеть, используя один или несколько внешних адресов. Обмен IP-пакетами происходит следующим образом:

— Внутренний интерфейс получает пакет от хоста во внутренней сети, адресованный во внешнюю сеть. Этот пакет маршрутизируется на некоторый внешний интерфейс.

— Внутренний IP-адрес источника (как правило, приватный) заменяется указанным внешним адресом. В общем случае, для пакетов TCP и UDP номер порта источника может заменяться некоторым новым номером порта, относящимся к данному интерфейсу (см. ниже).

— В списке потоков NAT создается запись о вновь созданном потоке, после чего пакет отправляется во внешнюю сеть.

— Когда из внешней сети поступает пакет, в котором адрес назначения равен IP-адресу Source NAT на данном интерфейсе, номер порта назначения или идентификатор пакета, указанные в заголовке, проверяются по списку потоков NAT. Если запись о потоке с таким номером порта или идентификатором найдена, то выполняется обратное преобразование, и пакет маршрутизируется во внутреннюю сеть. Если такая запись не найдена, т.е. ни один из внутренних хостов ранее не обращался к данному внешнему хосту (в пределах контролируемых таймаутов), пакет уничтожается.



Если от внешнего хоста будет получен пакет с адресом назначения, принадлежащим внутренней сети, то такой пакет считается некорректным и будет безусловно уничтожен. Если извне получен пакет с адресом назначения, не принадлежащим внутренней сети, то в NSG Linux он будет пропущен через интерфейс без изменений и отправлен далее в соответствии с таблицей маршрутизации.

Внешний IP-адрес или диапазон адресов для Source NAT задается статически и, в общем случае, никак не связан с адресами, назначенными интерфейсам устройства NSG. При этом, однако, необходимо следить за тем, чтобы на вышестоящих маршрутизаторах был известен обратный маршрут к этим адресам.

Для пакетов TCP и UDP номер порта источника по возможности сохраняется. Однако если два хоста из внутренней сети пытаются одновременно посылать пакеты с одним и тем же номером порта, то для одного из них номер порта будет изменен. При этом все возможные номера портов разбиваются на три диапазона: менее 512, от 512 до 1023, и старше 1023. Эти диапазоны инвариантны, т.е. порты из каждого диапазона отображаются только на порты из того же диапазона.

IP-адрес и номер порта назначения в любом случае остаются неизменными.

NAT в форме IP-маскарадинга работает аналогично Source NAT и используется для той же самой цели. Различие состоит в том, что вместо фиксированного внешнего адреса (или диапазона адресов) используется IP-адрес выходного интерфейса. Если интерфейс имеет несколько IP-адресов, то для маскарадинга используется первый по списку адрес из сети, подходящей для отправки данного пакета.

Кроме того, если используемый IP-интерфейс переходит в состояние DOWN, то все записи IP-маскарадинга, относящиеся к нему, уничтожаются. После включения интерфейса все потоки, проходящие через него, идентифицируются заново.

IP-маскарадинг предназначен, в основном, для ситуации, когда IP-адрес выходного интерфейса назначается динамически и априори неизвестен, а маршрутизация в вышестоящей сети производится только на этот адрес. Типичный пример — PPP-подключение к входному пулу поставщика услуг Интернет.

Допускается также использовать IP-маскарадинг в том случае, когда адреса выходных интерфейсов являются статическими. С качественной точки зрения, это решает поставленную задачу. Однако IP-маскарадинг является более сложным алгоритмом, чем Source NAT с фиксированным адресом, требует больше вычислительных ресурсов и сильнее сказывается на производительности устройства. Поэтому для статических задач рекомендуется указывать адрес интерфейса явным образом.

§4.8.4. Алгоритм Destination NAT

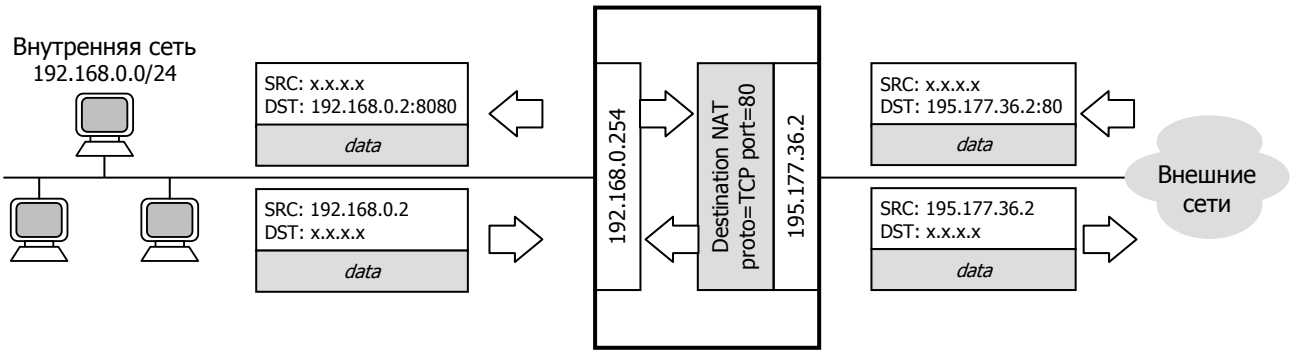
Алгоритм Destination NAT (DNAT), иначе называемый "виртуальными серверами", предназначен для доступа из внешней сети к определенным хостам во внутренней сети по явно указанным IP-адресам, протоколам и портам. При этом для внешнего мира все эти серверы будут расположены по внешнему IP-адресу интерфейса NSG. Обработка пакетов производится следующим образом:

— Внешний интерфейс NSG получает пакет, не относящийся ни к одному из потоков, инициированных со стороны внутренней сети.

— Если для данного протокола, работающего поверх IP (TCP, UDP, ICMP, GRE, PPTP, VPN и др.) и для данного номера порта назначения (TCP, UDP) или типа пакетов (ICMP) имеется запись в таблице NAT, то IP-адрес назначения заменяется на указанный адрес во внутренней сети; если указано, заменяется также порт назначения. Если подходящей записи не найдено, пакет уничтожается.

— Пакет передается на указанный адрес и порт во внутренней сети.

— Когда от внутреннего хоста приходит ответный пакет, IP-адрес источника заменяется внешним IP-адресом интерфейса, через который пакет отправляется во внешний мир, и пакет отправляется по назначению.



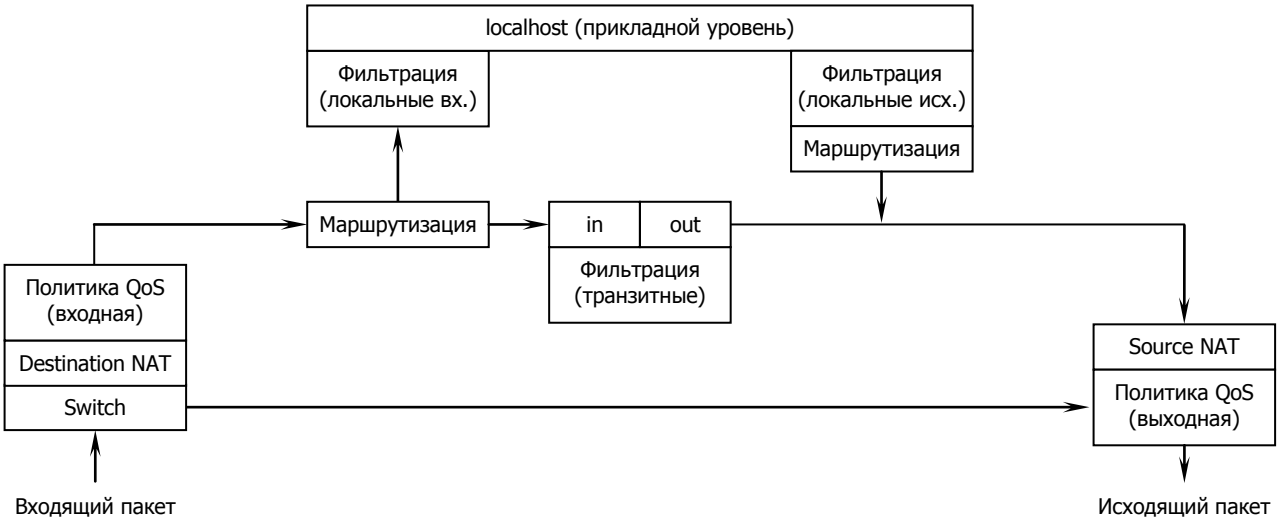
В NSG Linux пакеты, не подпадающие под явно установленные правила Destination NAT, передаются без изменений. При необходимости их можно отфильтровать с помощью *access-lists*; в этом случае механизм Destination NAT одновременно является фильтром, пропускающим из внешнего мира только пакеты с определенными IP-адресами и номерами портов назначения.

§4.8.5. Коммутация IP-пакетов

При использовании коммутации пакет, подпавший под действие установленного *access list*, безусловно передается на указанный выходной интерфейс (или маршрутизируется на указанный шлюз). Такую обработку можно рассматривать как статическую маршрутизацию по расширенному набору критериев (по совокупности IP адресов источника/назначения, протоколов, портов, протокольных флагов и т.п.). При этом пакет не проходит последующие правила NAT, обычной маршрутизации и фильтрации.

ВНИМАНИЕ В NSG Linux, в отличие от базового ПО NSG, при коммутации пакета его значение TTL уменьшается на единицу.

Взаимосвязь и очередность выполнения процедур IP-коммутации, NAT, маршрутизации, а также фильтрации (см. п.4.9) и QoS (см. п.4.10) показана на рисунке.



§4.8.6. Синтаксис команд NAT и IP-коммутации

Для управления настройкой NAT и IP-коммутации используются следующие команды в меню IP-интерфейсов (физических портов, VLAN, Frame Relay DLCI):

nat source [access-list {<1…32000 > | any }] [prio <1…32000>] { masquerade | <ip-адрес1> [< ip-адрес2>]}
no nat source prio <1…32000>

Создание и удаление правила для Source NAT. Правило применяется к исходящим пакетам, т.е. пакетам, отправляемым устройством NSG через данный интерфейс.
Параметр access-list указывает на номер *access-list* NSG, используемого для отбора трафика. Параметр необязательный. По умолчанию предполагается access-list any, т.е. правило применяется ко всему трафику.
Параметр prio позволяет установить порядок применения правил NAT (1 — наивысший приоритет). На одном интерфейсе может быть создано несколько правил Source NAT. По умолчанию, вновь создаваемые правила помещаются в конец списка. При создании правила с номером, попадающим в середину существующего списка, все последующие правила автоматически сдвигаются на одну позицию вниз.
Последним параметром команды должно быть ключевое слово masquerade либо по крайней мере один IP-адрес. Если указано masquerade, то в качестве адреса источника подставляется текущий IP-адрес интерфейса. Если указан один IP-адрес, то он и подставляется. Если задан диапазон адресов (два адреса — начальный и конечный, через пробел), то для каждого потока IP-пакетов выбирается случайный адрес из этого диапазона.

nat masquerade [access-list {<1…32000 > | any }] [prio <1…32000>]
nat snat [access-list {<1…32000 > | any }] [prio <1…32000>] <ip-адрес1> [< ip-адрес2>]

Синонимы nat source […] masquerade и nat source […] <ip-адрес1> [< ip-адрес2>], соответственно. Данный синтаксис введен для совместимости с предыдущими версиями NSG Linux и не рекомендуется к использованию в новых конфигурациях. При построении конфигурации автоматически заменяется на эти конструкции.

nat destination [access-list {<1…32000> | any }] [prio <1..32000>] <ip-адрес1> [< ip-адрес2>] ↵
[to-port <порт1> [<порт2>]
no nat destination prio <1…32000>

Создание и удаление правила для Destination NAT. Правило применяется к входящим пакетам, т.е. пакетам, принимаемым устройством NSG через данный интерфейс.
Параметры access-list и prio используются аналогично команде nat source … Так же используются и IP-адреса; в частности, при указании двух IP-адресов образуется виртуальный кластер серверов, и каждый новый запрос клиентов отправляется поочередно на следующий физический сервер.
Параметр to-port указывает, на какой номер порта назначения следует отправлять пакеты. Если заданы два номера портов, то используются (случайным образом) все номера из указанного диапазона.

ВНИМАНИЕ Если производится преобразование портов, то обязательно должен быть использован расширенный *access-list* (access-list ext-ip …) с указанием типа протокола TCP либо UDP в каждом правиле permit. Данный *access-list* должен быть создан до настройки NAT.

ВНИМАНИЕ Созданные группы правил Source NAT и Destination NAT действуют независимо друг от друга. Если одна группа содержит несколько правил, то применяется первое правило, под которое попал данный пакет; последующие правила из этой группы игнорируются.

switch [access-list {<1…32000> | any }] [prio <1..32000>] to-intreface <имя> [tee]
switch [access-list {<1…32000> | any }] [prio <1..32000>] gateway <ip-адрес> [tee]
switch [access-list {<1…32000> | any }] [prio <1..32000>] to-intreface <имя> gateway <ip-адрес> [tee]
no switch prio <1…32000>

Создание и удаление правила для коммутации пакетов. Правило применяется к входящим пакетам, т.е. пакетам, принимаемым устройством NSG через данный интерфейс.
Параметры to-interface и gateway определяют имя интерфейса и IP-адрес шлюза, на который следует безусловно перенаправлять пакеты. Если интерфейс является широковещательным, то адрес шлюза необходимо указывать обязательно. В остальных случаях достаточно одного из двух параметров.
Необязательный ключ tee, указываемый последним, вместо режима коммутации включает режим копирования пакетов на заданный интерфейс. При этом основной пакет обрабатывается обычным образом, т.е. проходит все правила NAT, маршрутизацию и фильтры.

ВНИМАНИЕ Если на интерфейсе, одновременно с NAT или IP-коммутацией, используется IPsec, то необходимо либо использовать механизм NAT Traversal (если он поддерживается обеими сторонами туннеля), либо исключить эти пакеты из-под действия NAT и IP-коммутации с помощью соответствующих *access lists*. Подробнее о данной проблеме см. Часть 5.

§4.9. Фильтрация IP-трафика

Для фильтрации IP-пакетов используются простые или расширенные списки доступа NSG (*access lists*). Управление данными списками производится в меню (config-nsg)# (см. §§4.3.4, 4.3.5). Созданный список доступа подключается индивидуально к требуемому IP-интерфейсу и отключается от него посредством следующих команд в меню соответствующих объектов (физических портов, DLCI, VLAN и т.п.):

access-group <transit | local > <input | output > <номер_access_list>
no access-group <transit | local > <input | output > <номер_access_list>

Подключение/отключение списка доступа на интерфейсе. Параметры команды:

local	Фильтр применяется к локальным пакетам (т.е. посылаемым или принимаемым различными службами непосредственно на устройстве NSG).
transit	Фильтр применяется к транзитным пакетам.
input	Фильтр применяется к пакетам, входящим в устройство NSG через данный интерфейс.
output	Фильтр применяется к пакетам, исходящим из устройства NSG через данный интерфейс.
<номер_access_list>	Номер используемого списка доступа.

Все параметры обязательны как при подключении, так и при отключении фильтра.

Таким образом, фильтрация производится раздельно не только по интерфейсам, но и по следующим четырем категориям пакетов:

access-group transit input
access-group transit output
access-group local input
access-group local output

По умолчанию, т.е. если фильтрация для некоторой категории пакетов не включена, интерфейс пропускает все пакеты этой категории. При включенной фильтрации используются следующие правила:

— Для пакетов, попавших под какое-либо правило permit из данного *access-list*, применяется политика ACCEPT, т.е. пакет пропускается.
— Для пакетов, попавших под правило deny, выполняется политика DROP, т.е. пакет уничтожается.
— Для всех пакетов, дошедших до конца *access-list* и не попавших ни под одно правило, применяется политика DROP.
— Если указанный список доступа существует, но пуст, то для всех пакетов применяется политика DROP.
— Если указанный список доступа отсутствует, то для всех пакетов применяется политика ACCEPT (т.е. фильтрация не включается).

Если фильтрация используется совместно с NAT, то преобразование Destination NAT выполняется до применения фильтров, а Source NAT — после применения фильтров. Таким образом, в фильтрах следует указывать следующие IP-адреса:

— для всех входных фильтров (access-group ... input) — внутренний Destination IP
— для всех выходных фильтров (access-group ... output) — внутренний Source IP

Подробно схему совместного применения NAT, QoS и фильтров см. в п.4.8.5.

§4.10. Механизмы управления трафиком

§4.10.1. Описание и подключение алгоритмов управления трафиком

Для ограничения входящего и исходящего трафика в NSG Linux используются так называемые *политики* управления трафиком. Политика представляет собой совокупность правил, позволяющих классифицировать трафик и применить установленные ограничения к каждому классу. Для создания и удаления политик используется следующая команда в меню (config-nsg)#:

policy-map <имя>
no policy-map <имя>

Первая команда создает политику (если политика с таким именем не существует) и входит в меню ее редактирования. Вторая — удаляет существующую политику.

В меню настройки политики устанавливается, в первую очередь, тип политики:

type { tbf | ingress | cbq | sfq | prio }

Каждая политика характеризуется определенным алгоритмом, или дисциплиной, обработки трафика. В данной версии NSG Linux поддерживаются следующие дисциплины для исходящего трафика:

TBF	Token Bucket Filter
CBQ	Class Based Queuing
SFQ	Stochastic Fairness Queueing
PRIO	приоритизация по битам TypeOfService/DiffServ

и для входящего трафика:

INGRESS	управление входящим трафиком на основе классов

Далее задаются параметры политики, специфические для каждого из типов. Подробное описание настроек для отдельных дисциплин приведено в следующих параграфах.

ПРИМЕЧАНИЕ При смене типа политики все настройки прежнего типа утрачиваются.

Созданные политики подключаются в меню объектов, через которые происходит передача IP-трафика, а именно:
— физических портов с инкапсуляцией Cisco-HDLC, PPP, RAW, Ethernet — меню (config-port-XXX)#
— виртуальных каналов Frame Relay — меню (config-dlci-NN)#
— виртуальных ЛС Ethernet — меню (config-vlan-NN)#
— туннелей — меню (config-tuni-NN)# и (config-tunx-NN)#
при помощи следующих команд:

service-policy input <имя>
no service-policy input <имя>

Установить/удалить политику для обработки входящего трафика. Параметр <имя> должен указывать на существующую политику типа ingress. Если политика не задана, принимается весь входящий трафик.

service-policy output <имя>
no service-policy output <имя>

Установить/удалить политику для обработки входящего трафика. Параметр <имя> должен указывать на существующую политику типа tbf, cbq, sfq или prio. Если политика не задана, то по умолчанию используется алгоритм pfifo_fast (см. следующий параграф).

ПРИМЕЧАНИЕ Данная команда используется также для того, чтобы включить указанный порт, DLCI или VLAN в состав многоканального IP-интерфейса типа teql (см. §4.10.9).

service-policy show

Вывести настройки и статистику обработки трафика.

ВНИМАНИЕ Процедуры QoS выполняются на уровне IP-маршрутизатора, т.е. до передачи пакета в драйвер физического порта. В драйвере порта образуется своя очередь, длина которой определяется параметром tx-ring-limit. Пакеты, поставленные в эту очередь, уже неподконтрольны процедурам вышестоящих уровней и отправляются в порядке поступления. Иначе говоря, если в этой очереди имеется 8 пакетов из низкоприоритетных очередей, и после этого поступает высокоприоритетный пакет (например, VoIP), то механизм QoS сможет отправить его раньше других поступающих пакетов — но не раньше пакетов, уже поставленных в эту очередь. Для надлежащей обработки приоритетного трафика следует уменьшить значение tx-ring-limit.

Подробное описание алгоритмов и методов управления трафиком в ОС Linux приведено в документе: *Linux Advanced Routing & Traffic Control HOWTO* (ftp://nsg.net.ru/pub/nsg-linux/doc/lartc.pdf).

§4.10.2. Политика PFIFO_FAST

Политика PFIFO_FAST жестко определена и распределяет пакеты по трем исходящим очередям в зависимости от комбинации бит Type of Service (DiffServ). Эта политика относится к *исходящему* трафику и применяется по умолчанию на всех интерфейсах, для которых не указана иная политика, а также в тех классах cbq, для которых не определены никакие дочерние объекты (т.е. ни политика типа tbf, ни внутренние подклассы).

Базовые значения поля ToS (DiffServ)

Двоичное	Десятичное	Назначение
1000	8	Minimize Delay
0100	4	Maximize Throughput
0010	2	Maximize Reliability
0001	1	Minimize Monetary Cost
0000	0	Normal Service

Значение поля ToS bin	dec	Выходная очередь
0000	0	1
0001	1	2
0010	2	1
0011	3	1
0100	4	2
0101	5	2
0110	6	2
0111	7	2
1000	8	0
1001	9	0
1010	10	0
1011	11	0
1100	12	1
1101	13	1
1110	14	1
1111	15	1

Приоритеты очередей являются абсолютными, т.е. если в очереди 0 имеются пакеты, то очереди 1 и 2 не обслуживаются до тех пор, пока не будет опустошена очередь 0. Если очередь 0 пустая, то обслуживается очередь 1. Если пакетов нет ни в очереди 0, ни в очереди 1, то обслуживается очередь 2.

При необходимости вместо этой политики можно определить аналогичную политику типа cbq, в которой число очередей и схема распределения пакетов могут быть перенастроены по усмотрению пользователя средствами командной оболочки Linux.

Пример. Чтобы обеспечить приоритетную передачу пакетов Voice-over-IP, нужно на VoIP-шлюзе выставить следующие биты ToS:

mininize delay=1
maximize throughput=0

остальные два бита несущественны. В этом случае пакеты, поступающие от шлюза, будут ставиться устройством NSG в нулевую очередь.

§4.10.3. Политика TBF

Политика типа tbf формирует *исходящий* трафик с помощью алгоритма "дырявого ведра" (token bucket). Управление осуществляется для интерфейса в целом, без разделения трафика на классы. В меню политики доступны следующие специфические параметры:

rate <8000...100000000>
: Установить информационную скорость на интерфейсе (бит в секунду).

burst <64...1000000>
: Установить размер "ведра" (*bucket*) в байтах.

limit <64...1000000>
: Установить размер очереди в байтах.

При выборе параметров управления трафиком необходимо учитывать следующие соображения:

— По смыслу задачи, величина rate не может быть больше максимальной скорости физического порта (параметр baudrate) или максимальной информационной скорости Frame Relay DLC (величина cir*(1+be/bc))

— Для нормальной работы алгоритма значения burst и limit должны быть не меньше максимального размера пакета (величина MTU + длина заголовков), который может быть отправлен интерфейсом

— Для эффективного использования памяти и достижения заданной информационной скорости рекомендуется соблюдать условия:

burst ≥ 2*packet, где packet — максимальный размер пакета
burst ≥ rate/400 (это ограничение связано с дискретностью таймера Linux — 1/100 сек)
limit ≥ 2*burst

— Если burst ≥ baudrate/800, и в течение некоторого времени интерфейс не передает данные, а затем поступает сразу большая порция данных (например, начинается передача большого файла), то в течение некоторого времени ему разрешается передавать с максимальной скоростью, допускаемой портом. После этого скорость опускается до значения rate. Продолжительность начального всплеска трафика будет тем больше, чем больше был период молчания перед ним. Максимальная продолжительность всплеска составляет 8*burst/baudrate секунд (с округлением до 1/100 сек). Чтобы ограничить время, в течение которого один пользователь может узурпировать полосу пропускания, следует ограничить величину burst.

Пример. К порту Fast Ethernet применена следующая политика с именем max200k:

```
!
nsg
    policy-map max200k
        type tbf
        rate 200000
        burst 3100
        limit 20480
        exit
    port eth0
        service-policy output max200k
        exit
    exit
!
```

В данном случае исходящий трафик через порт ограничен средней скоростью 200 Кбит/с. На максимальной скорости может быть передано не более 3100 байт подряд. Максимальная длина очереди — 20 Кбайт.

§4.10.4. Классы трафика

Ряд политик управления трафиком предполагает, что весь входящий трафик разбивается на классы (с дальнейшей внутренней иерархией или без нее), к каждому из которых применяется свой алгоритм и параметры управления. В меню политик, основанных на использовании классов, доступны следующие общие команды для управления классами:

class <имя>
no class <имя>
: Создать/редактировать класс или удалить существующий класс, соответственно. Метод разделения трафика на классы используется также в ряде других политик.

Дальнейшая настройка производится в меню выбранного класса (config-class-XXX)#:

prio <0...65535>
: Определяет приоритет фильтрации в данный класс в рамках данной политики (1 — наивысший). Значение 0 означает, что приоритет явным образом не задается и класс будет рассматриваться в той очередности, в какой он был создан.
Два класса могут иметь равные приоритеты; в этом случае они также будут рассматриваться в той очередности, в какой они были созданы.

match { access-list <номер> | no }
: Определяет *access-list*, по которому будет производиться отбор пакетов в данный класс. Параметр <номер> может указывать как на расширенный, так и на стандартный *access-list* NSG (см. пп.4.3.4, 4.3.5). В случае, если используется стандартный *access-list*, анализируется только адрес источника (*source address*), указанный в пакете.
Возможны и другие механизмы отбора пакетов в некоторый класс. Например в очереди типа prio пакеты распределяются по классам на основе значения поля ToS/DiffServ.

Значение no (установлено по умолчанию) показывает, что для данного класса НЕ используется механизм отбора пакетов при помощи *access-lists*. Соответственно, если не настроен параметр match то параметры prio и police в рамках данного класса также не используются. То же самое происходит в случае, если *access-list* с указанным номером не существует.

police no Пропускать все пакеты данного класса без ограничений.

police tbf rate <8000...100000000> burst <64...1000000> [mtu <256...65535>] action { drop | pass | continue }

Установить параметры TBF для данного класса и правила для трафика, выходящего за установленные ограничения.

Параметры rate и burst имеют тот же смысл, что и для алгоритма TBF.

Параметр mtu (Maximum Transfer Unit) определяет максимальную длину пакета (в байтах), подпадающего под данную политику. Если длина пакета превышает установленную величину, пакет обрабатывается по правилам, установленным для избыточного трафика.

Параметр action определяет, как следует поступить с избыточными пакетами:

drop	Сбросить пакет
pass	Пропустить пакет без дальнейшей проверки. Фактически такой класс никак не ограничивает трафик, однако пакеты, выходящие за установленные ограничения, учитываются в статистике отдельно.
continue	Продолжить проверку данного пакета по менее приоритетным классам и отправить его в соответствии с тем классом, под который он подойдет.

Входящие пакеты проверяются на соответствие заданным классам поочередно, в порядке убывания приоритета (возрастания значения prio) класса. Если пакет удовлетворяет списку *access-list*, установленному для некоторого класса, то к нему применяется алгоритм TBF с параметрами, установленными для этого класса. Если при этом оказывается, что пакет не вписывается в ограничения, установленные параметрами rate и burst, то с ним выполняется действие, определенное параметром action.

Если пакет не подпадает ни под один из определенных классов, он пропускается без использования ограничений. Чтобы изменить это правило, следует определить класс с минимальным приоритетом и указанием на тривиальный *access-list*, под который будут подпадать любые пакеты.

§4.10.5. Политика INGRESS

Политика типа ingress основана на использовании классов и ограничивает *входящий* трафик при помощи алгоритма TBF. К каждому из классов, определенных в рамках данной политики, применяется алгоритм "дырявого ведра" со своими параметрами.

В тривиальном случае в политике типа ingress может быть определен единственный класс с критерием отбора по такому *access-list*, под который будут подпадать любые пакеты. Если некоторому интерфейсу назначена такая политика, то ограничения применяются ко всему трафику интерфейса в совокупности.

§4.10.6. Политика CBQ

Политика типа cbq позволяет регулировать доступную полосу пропускания и распределять ее между многими категориями *исходящего* трафика. Она включает в себя одноименный алгоритм управления трафиком в целом и описания классов, к которым применяется раздельное управление трафиком в рамках этого целого. Каждый из классов, в свою очередь, содержит в себе дочернюю дисциплину CBQ, TBF или (по умолчанию, т.е. если дисциплина не установлена явным образом) PFIFO_FAST.

Дочерняя политика CBQ, в свою очередь, также может содержать в себе классы; таким образом, формируется разветвленное дерево классов с неограниченным количеством уровней. Узлами этого дерева являются классы, для которых установлена дисциплина CBQ с несколькими дочерними классами (*child classes*); конечными элементами ("листьями") — классы, для которых установлена дисциплина TBF, дисциплина CBQ без использования классов, либо не установлена никакая (в этом случае используется дисциплина PFIFO_FAST).

§4.10.6.1. Алгоритм CBQ без использования классов

Алгоритм CBQ опирается не на подсчет фактически отправленных данных, а на контроль времени, прошедшего между посылкой пакетов. Помимо этого, в вычисления оказываются вовлечены еще ряд усредненных или оценочных величин, а также искусственные ограничители для предельных ситуаций. Все они устанавливаются административно, что, с одной стороны, придает алгоритму CBQ большую гибкость, но с другой — делает его весьма сложным и не всегда результативным.

Например, если скорость в линии составляет 10 Мбит/с, а желаемая скорость исходящего трафика — 1 Мбит/с, то соединение должно оставаться незагруженным в течение 90% времени. Исходя из этого, вычисляется средняя величина интервала, который следовало бы выдерживать перед посылкой следующего пакета. С другой стороны, при отправке пакетов подсчитывается фактический средний интервал между ними (по методу экспонециально-взвешенного среднего, EWMA). Разница между расчетным и фактическим значениями показывает расхождение между теоретической и практической скоростями передачи; в терминах Linux эта переменная называется *avgidle*. Качественно и количественно возможны следующие ситуации:

avgidle = 0 — Идеальный случай: соединение загружено в заданной степени, т.е. пакеты отправляются по одному строго через расчетные интервалы времени.

avgidle < 0 — Соединение перегружено, т.е. пакеты отправляются слишком часто. Необходимо "придержать" очередь сообразно абсолютной величине *avgidle*.

avgidle > 0 — Соединение недогружено, т.е. пакеты отправляются слишком редко. Необходимо "ускорить" очередь сообразно абсолютной величине *avgidle*.

Основной "костяк" алгоритма CBQ образуют следующие параметры, доступные в меню политики этого типа:

bandwidth <8000...100000000>
: Физическая полоса пропускания для данного канала (порта, Frame Relay DLC и т.п.) в бит/с. Используется для расчета интервалов между пакетами. Значение данного параметра не обязательно равно номинальному быстродействию интерфейса; в частности, с его помощью можно указать некоторое практическое быстродействие интерфейса Ethernet Half-Duplex, составляющее, по природе сетей этого типа, 60–65% от номинального.

rate <8000...100000000>
: Желаемая полоса пропускания для трафика, подпадающего под действие данной дисциплины, в бит/с.

cell { 2 | 4 | 8 | 16 | 32 | 64 }
: Количество градаций, используемых для расчета задержки очередного пакета в очереди. Например, для передачи пакета длиной 800 байт и 806 байт требуется практически одинаковое время, поэтому нет практического смысла в слишком точных вычислениях. Значение данного параметра устанавливаются по степеням 2. Бóльшие значения обеспечивают более точное поддержание требуемой скорости передачи, меньшие — снижают нагрузку на процессор.

Особые ситуации могут возникать, когда какой-либо из используемых параметров принимает слишком малые или слишком большие (по абсолютной величине) значения. Чтобы они не вводили систему в неработоспособное или физически бессмысленное состояние, используются ограничители, приведенные ниже.

maxburst <1...1000000>
: Если в течение длительного времени не было пакетов на передачу, то контрольная переменная *avgidle* может принимать очень большие значения. В результате, когда появятся пакеты, разрешенная скорость для их передачи может оказаться сколь угодно большой. Чтобы избежать этого, вводится предельное значение *maxidle*. В данной реализации алгоритма оно устанавливается через эквивалентный параметр maxburst, определяющий, сколько пакетов среднего размера разрешается передать подряд, без задержки; после этого значение будет принудительно обнулено, чтобы ввести дальнейшую передачу в регулируемый режим.

avpkt <32...65535>
: Средний размер пакета, в байтах. Используется для расчета *maxidle* (как maxburst/avpkt).

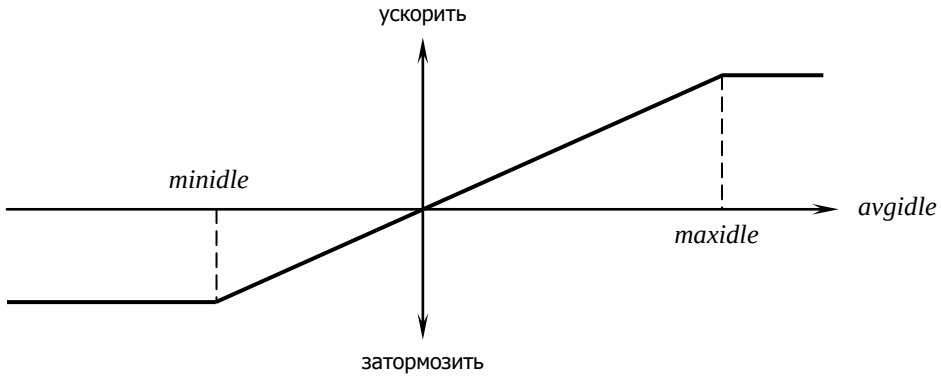
minburst <1...100000000>
: Дискретность реакции алгоритма на превышение установленной скорости передачи. В идеале, система должна была бы задержать следующий пакет строго на установленное время, затем передать его, и снова ждать. Однако в связи с тем, что системный таймер работает с точностью 10 мс, это не всегда практично; удобнее выдержать больший интервал, а затем отправить сразу несколько пакетов — именно это число устанавливается данным параметром.
Более высокие значения minburst обеспечивают более точное управление трафиком на длительных промежутках времени, однако приводят к менее равномерному трафику в миллисекундном масштабе.

minidle <1...100>
: Слишком большой всплеск трафика может привести переменную *avgidle* к чрезмерно низким отрицательным значениям, и в этом случае придется ждать очень долго, прежде чем можно будет послать хотя бы один пакет. Чтобы избежать этого, вводится предельное значение *minidle*, ниже которого переменная опускаться не может.
Значение данного параметра устанавливается в "отрицательных микросекундах", т.е. minidle 10 соответствует порогу в –10 мкс.

mpu <32...65535>

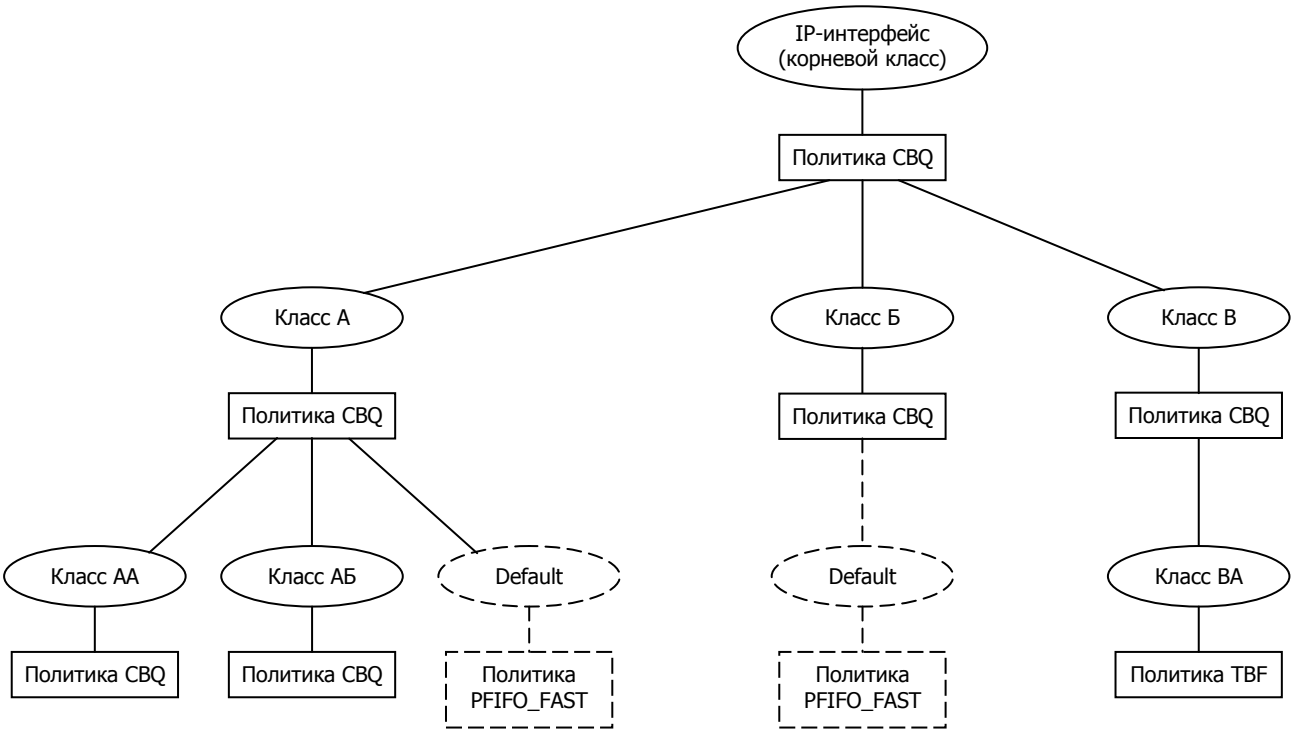
Минимальная длина пакета, в байтах. Например, в сети Ethernet даже пакет нулевой длины занимает ненулевую полосу пропускания: в него добавляются байты-заполнители до длины 64 байт — минимально допустимой в рамках данной технологии. Соответственно, для передачи такого пакета также требуется ненулевое время. Точное знание минимальной длины пакета необходимо для корректной работы алгоритма CBQ.

Влияние этих ограничителей на отправку пакетов схематически показано на рисунке.



§4.10.6.2. Создание классов

Политика CBQ позволяет создать иерархическое дерево классов, для каждого из которых определена своя политика формирования трафика. Пример относительно несложного дерева показан на рисунке.



Создание классов и назначение политик для них происходит по следующим правилам:

— Если в рамках одной политики CBQ определено несколько классов, то всем им должны быть назначены политики также типа CBQ.
— Если в рамках одной политики CBQ определен единственный класс, то этому классу может быть назначена политика CBQ либо TBF.
— Каждому классу обязательно должна быть назначена некоторая политика. Классы, которым не назначена никакая политика, игнорируются.
— Если пакет подпадает под "родительский" класс, но не подпадает ни под один из определенных в нем дочерних классов, он относится к классу "по умолчанию", с политикой PFIFO_FAST и минимальными приоритетами самого класса и политики в нем. То же самое происходит, если внутри политики CBQ не определено ни одного класса. Чтобы изменить это правило, следует явно определить дочерний класс с минимальным приоритетом и значением match any.

Для управления классами в меню политики типа cbq используются общие команды, приведенные в §4.10.4, а также следующая команда в меню класса (config-class-XXX)#:

service-policy <имя>
Подключение/отключение дочерней политики, действующей внутри данного класса. Указанная политика должна быть определена ранее и иметь тип cbq либо tbf. (При этом tbf допускается только для класса, являющегося единственным в вышестоящей политике.)
Данная команда позволяет определять внутри класса политику, которая, в свою очередь, также может быть основана на использовании классов. Таким образом, формируется иерархическое дерево классов.
К каждому классу может быть подключена только одна политика. Если никакая политика не подключена, то по умолчанию используется общая политика PFAST_FIFO.

ПРИМЕЧАНИЕ Чтобы отключить политику, в данной версии NSG Linux следует ввести пустое имя:
service-policy ""

Входящие пакеты проверяются на соответствие заданным классам поочередно, в порядке убывания приоритета (возрастания значения prio) класса. Если пакет удовлетворяет списку *access-list*, установленному для некоторого класса, то к нему применяется политика, установленная для этого класса.

§4.10.6.3. Управление трафиком с использованием иерархических классов

Если в рамках политики типа CBQ определены несколько дочерних классов, то распределение полосы пропускания между ними производится по двухступенчатой схеме. Во-первых, дочерние классы CBQ имеют свои приоритеты, установленные для соответствующих дочерних политик. Если в классе с более высоким приоритетом (меньшим значением prio) имеются пакеты для передачи, то остальные классы не обслуживаются. Во-вторых, если несколько дочерних классов имеют равный приоритет, то для них используется алгоритм взвешенно-случайного выбора (Weighted Round Robin, WRR). Для управления этими механизмами предусмотрены следующие параметры в меню политики (config-policy-map-XXX)#:

prio <0...65535>
Приоритет данной политики и связанного с ней класса среди других классов/политик данного узла иерархического дерева.

ВНИМАНИЕ Следует отличать данный параметр от параметра prio класса, определяющего порядок отбора пакетов в различные классы.

allot <1...100000000>
Базовая "ставка" (в байтах), распределяемая за один такт алгоритма WRR между классами с равным приоритетом.

weight <1...100>
Вес данного класса среди других классов с таким же приоритетом, в условных единицах. Веса всех классов с одинаковым приоритетом автоматически нормализуются на единицу, поэтому существенны лишь относительные значения этого параметра для различных классов. В качестве исходного приближения, рекомендуется установить weight=rate/10.

Помимо этого механизма, допускается перераспределение полосы пропускания между различными классами в пределах одного узла, если пакеты для передачи имеются не во всех классах. Возможность перераспределения регулируется параметрами:

isolated { yes | no }
Запретить/разрешить данному классу предоставлять дополнительную полосу пропускания соседним классам:
yes Запретить
no Разрешить

bounded { yes | no }
Запретить/разрешить данному классу запрашивать дополнительную полосу пропускания у соседних классов:
yes Запретить
no Разрешить

В частности, если все классы в одном узле имеют конфигурацию isolated yes и bounded yes, то каждый из них имеет жестко фиксированную полосу пропускания, определенную параметром rate. Напротив, если все классы имеют конфигурацию isolated no и bounded no, то общая полоса пропускания перераспределяется между классами динамически, в пределах максимальных значений rate, установленных для каждого класса.

§4.10.6.4. Пример конфигурации политики CBQ

Постановка задачи, для удобства сравнения, взята из документа: *Linux Advanced Routing & Traffic Control HOWTO*. На интерфейсе Fast Ethernet требуется ограничить трафик Web-сервера до скорости 5 Мбит/с, трафик SMTP — 3 Мбит/с. В совокупности оба вида трафика должны занимать не более 6 Мбит/с, но в пределах этой величины могут перераспределять полосу пропускания между собой.

```
!
nsg
    access-list ext-ip 101
        add 1 permit tcp any eq 80 any
        exit
    access-list ext-ip 102
        add 1 permit tcp any eq 25 any
        exit
    policy-map web5
        type cbq
        bandwidth 100000000
        rate 5000000
        cell 8
        avpkt 1000
        maxburst 20
        prio 5
        allot 1514
        weight 500000
        isolated no
        bounded no
        exit
    policy-map mail3
        type cbq
        bandwidth 100000000
        rate 3000000
        cell 8
        avpkt 1000
        maxburst 20
        prio 5
        allot 1514
        weight 300000
        isolated no
        bounded no
        exit
    policy-map web5_mail3
        type cbq
        bandwidth 100000000
        rate 6000000
        cell 8
        avpkt 1000
        maxburst 20
        prio 8
        allot 1514
        weight 600000
        bounded yes
        class web
            prio 1
            match access-list 101
            policy-map web5
            exit
        class mail
            prio 2
            match access-list 102
            policy-map mail3
            exit
        exit
    port eth0 service-policy output web5_mail3
    exit
!
```

§4.10.7. Политика PRIO

Политика PRIO основана на использовании классов и предназначена для управления приоритизацией трафика в соответствии со значениями бит TypeOfService/DiffServ. В меню политики данного типа имеются следующие специфические команды:

bands <1…64>
: Количество классов. На основе этой цифры формируется список классов class 0 (наивысший приоритет), class 1 и т.д. до максимального значения B=bands-1.

priomap <0..B> <0..B> … <0..B>
: Массив из 16-ти элементов, указывающих, в какой класс направлять пакеты у которых поле TOS равно номеру элемента в массиве. Каждый элемент массива может иметь значение в диапазоне от 0 до B=bands-1.

а также общий узел class, содержащий команды для настройки каждого класса (см. §4.10.4).

Отбор пакетов в тот или иной класс осуществляется в два этапа:

1. На основе фильтров.
2. Пакеты, не попавшие ни в один из классов, определенных фильтрами, сортируются в соответствии с заданной картой приоритетов (priomap).

Следует заметить, что механизм фильтрации на основе исключительно поля ToS требует существенно меньше вычислительных ресурсов, чем фильтрация по *access-lists*, и мало сказывается на производительности устройства.

По умолчанию (при создании) очередь типа prio настроена в точности так же, как и стандартная очередь PFIFO_FAST интерфейса, т.е. содержит 3 класса и делит весь трафик между ними в соответствии со стандартной картой приоритетов (1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1).

Пример настройки политики PRIO:

```
access-list ext ip 123
    add 1 permit ip any any
    exit
policy-map PRIO
    type prio
    bands 4
    priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
    class 0
        match no
        police no
        prio 0
        exit
    class 1
        match no
        police no
        prio 0
        exit
    class 2
        match access-list 100
        police no
        prio 0
        exit
    class 3
        match access-list 123
        police tbf rate 10000000 burst 100000 action pass
        prio 0
        exit
    exit
```

§4.10.8. Политика SFQ

Политика SFQ (Stochastic Fairness Queueing) заключается в том, что трафик разделяется на большое число потоков (по сеансам TCP, потокам UDP и т.п. — аналогично тому, как это делается в механизме NAT) и каждому потоку предоставляется полоса пропускания псевдослучайным образом. Использование такого алгоритма имеет смысл только на полностью загруженных соединениях. Он обеспечивает относительно равномерное распределение полосы пропускания между потоками и не позволяет какому-либо одному потоку полностью занять всю имеющуюся полосу. Алгоритм SFQ менее точен, чем остальные, зато требует значительно меньшего объема вычислений.

Данная политика в большой степени самонастраивающаяся, поэтому для нее в меню (config-policy-map-XXX)# имеются только две специфические команды:

perturb <1...100>

Время между перестройкой алгоритма распределения полосы пропускания (в секундах). Псевдослучайный алгоритм использует хэширование, поэтому является только односторонне однозначным: два различных потока могут случайно оказаться "в одной лунке рулетки" и в результате получить один и тот же квант полосы пропускания на двоих. Периодическая перестройка хэширования позволяет гарантировать, что если такая ситуация и возникает, она продлится не дольше ограниченного времени. Значение по умолчанию — 10 сек.

quantum <64...100000>

Количество байт, отправляемое из одной очереди за один шаг работы алгоритма. Значение по умолчанию — 1516. Не следует устанавливать данный параметр меньше максимального MTU пакетов, проходящих через данный интерфейс.

§4.10.9. Балансировка трафика между несколькими IP-интерфейсами

Утилита *teql* позволяет объединить несколько IP-интерфейсов, привязанных к различным объектам канального уровня (физическим портам, виртуальным каналам Frame Relay, виртуальным LAN), в один логический IP-интерфейс. С точки зрения IP-маршрутизатора, такой интерфейс представляет собой единое целое, а внутри него балансировка исходящего трафика и резервирование каналов связи осуществляются прозрачным образом.

ВНИМАНИЕ Механизм teql воздействует только на *исходящий* трафик. Чтобы обеспечить распределение трафика между несколькими каналами связи при передаче в двух направлениях (например, между двумя офисами в корпоративной сети), необходимо настроить teql на обеих сторонах.

Настройка многоканальных IP-интерфейсов осуществляется в три этапа:

— Создание интерфейса teql
— Подключение "реальных" IP-интерфейсов к интерфейсу teql
— Маршрутизация трафика на IP-интерфейс teql

Создание и удаление многоканальных IP-интерфейсов производится в меню (config-nsg)# командами:

teql <1...255>
no teql <1...255>

Первая команда создает в системе IP-интерфейс с именем teqlN (если он не существует) и входит в меню его настройки, вторая — удаляет существующий интерфейс. Созданные интерфейсы можно просмотреть командой ifconfig. Интерфейсы teql обладают всеми параметрами, присущими "реальным" IP-интерфейсам (за исключением, естественно, параметров канального и физического уровня). На них, как и на "реальных" IP-интерфейсах, можно активизировать QoS, NetFlow и другие услуги. В частности, интерфейс teql может быть подключен, наравне с "реальными", к другому teql, т.е. многоканальные IP-интерфейсы могут включаться каскадно.

Подключение "реальных" IP-интерфейсов к многоканальному производится в меню физических портов, Frame Relay DLCI, VLAN и т.п., причем в составе одного интерфейса teql могут присутствовать разные типы "реальных" IP-интерфейсов. Для этого используется команда service-policy в следующем формате:

service-policy output teql <номер>
no service-policy output teql <номер>

Включение текущего IP-интерфейса в состав многоканального интерфейса и исключение из него. Последний параметр должен указывать на существующий многоканальный интерфейс. Если многоканальный интерфейс с указанным номером не существует, команда будет проигнорирована.

Как можно видеть по формату данной команды, для IP-интерфейса возможны три опции относительно исходящего IP-трафика: либо на нем определена некоторая политика QoS (см. §4.10.1), либо он включен в состав многоканального интерфейса, либо ни то, ни другое (и в этом случае на нем используется политика pfifo_fast).

Каждому IP-интерфейсу, включенному в состав многоканального, должен быть назначен какой-либо IP-адрес. В большинстве случаев рекомендуется назначать "реальным" IP-интерфейсам заведомо фиктивные адреса, чтобы не запутывать маршрутизацию, а интерфейсу teql — "настоящий" IP-адрес. Назначать "настоящие" IP-адреса отдельным интерфейсам целесообразно лишь в том случае, если через них избирательно маршрутизируется какой-то трафик, помимо teql.

Если в состав многоканального IP-интерфейса не включено ни одного "реального" IP-интерфейса, то интерфейс номинально существует, но его пропускная способность равна нулю.

После того, как многоканальный IP-интерфейс создан и настроен, следует определить маршрутизацию пакетов через него — так же, как и через обычный IP-интерфейс (см. п.4.2):

```
ip route <ip-префикс> teqlN
```

ПРИМЕЧАНИЕ В продуктах некоторых других производителей аналогичный механизм называется *load balancing in per-packet mode*.

Пример настройки многоканального интерфейса:

```
!
nsg
    teql 3
        ip address 10.0.0.1/30
        exit
    port s1 ip address 192.168.0.1/31
    port s1 service-policy output teql 3
    port s2 ip address 192.168.0.2/31
    port s2 service-policy output teql 3
    exit
!
ip route 123.45.67.0/24  teql3
!
```

§4.10.10. Балансировка трафика между несколькими маршрутами

В настоящей версии NSG Linux реализована поддержка множественных маршрутов с балансировкой нагрузки (аналог *equal cost multipath* в ОС Linux, *load balancing in per-session mode* в продуктах других производителей). Именно, если в таблице маршрутизации имеется несколько маршрутов с одинаковой сетью назначения и одинаковой метрикой, пакеты будут распределяться между всеми этими маршрутами.

Сами маршруты могут быть получены автоматически для непосредственно подключенных сетей, статически (средствами основной командной оболочки), или с помощью протоколов динамической маршрутизации. Исключением являются маршруты, созданные непосредственно средствами ОС Linux.

По умолчанию, данная функция включена. Если такое поведение системы нежелательно, то следует использовать метрики (или механизм переназначения метрик при динамической маршрутизации) для выбора приоритетных маршрутов.

§4.11. Мониторинг IP-трафика

Мониторинг трафика в устройстве, работающем под управлением NSG Linux, может производиться с помощью внешних серверов, собирающих и обрабатывающих статистику в формате NetFlow. Механизм генерации статистических сообщений реализован на основе утилиты *fprobe*.

Для сбора статистики необходимо определить в устройстве шаблоны (*templates*) для отсылки пакетов NetFlow и назначить каждому из интересующих IP-интерфейсов какой-либо из этих шаблонов. Для создания и удаления шаблонов NetFlow используется следующая команда в меню (config-nsg)#:

netflow <1…255>
no netflow <1…255>
: Первая команда создает шаблон с указанным номером (если он не существует) и входит в меню его редактирования. Вторая — удаляет существующий шаблон.

В меню настройки шаблона содержатся следующие команды:

adm-state { up | down }
: Административное состояние шаблона. Данный параметр включает или выключает сбор статистики для всех интерфейсов, которые настроены на использования данного шаблона. По умолчанию, все создаваемые шаблоны находятся в состоянии UP.

description <описание>
: Текстовое описание шаблона, для удобства администрирования. Максимальная длина описания — 255 символов. Значение по умолчанию — пустая строка.

destination1 { <ip-адрес> [<порт>] | no }
destination2 { <ip-адрес> [<порт>] | no }
destination3 { <ip-адрес> [<порт>] | no }
destination4 { <ip-адрес> [<порт>] | no }
: Адреса и номера портов серверов NetFlow, которым отсылается статистика. Всего в шаблоне можно задать до 4 серверов. Если не задано ни одного сервера (т.е. для всех четырех параметров указано значение no), то сбор статистики с помощью данного шаблона выключен, также как и при adm-state down.
Номер порта UDP для сбора статистики является опциональным. Если номер не указан, то по умолчанию статистика отсылается на порт 9996.

source <ip-адрес>
: Обратный адрес в пакетах, отправляемых серверу статистики. Если этот адрес явным образом не указан, то в качестве адреса источника указывается IP-адрес интерфейса, через который отправляется пакет.

version { 1 | 5 | 7 }
: Версия пакетов NetFlow. По умолчанию, пакеты генерируются в формате NetFlow v5.

После того, как шаблон создан, его следует включить на интересующих администратора интерфейсах.Для этой цели используется следующая команда в меню физического порта, Frame Relay DLCI, либо VLAN:

ip netflow {<номер_шаблона> | disable }
: Включить сбор и отсылку статистики на основе указанного шаблона, либо выключить сбор статистики. Допускается использовать только номера шаблонов, существующих на данный момент.

Все указанные настройки вступают в силу немедленно. Кроме того, при удалении шаблона автоматически устанавливается значение ip netflow disable на всех интерфейсах, использовавших данный шаблон.

Пример минимальной настройки NetFlow для интерфейса Fast Ethernet устройства NSG–900:

```
!
nsg
netflow 1
    destination1 123.145.167.189
    exit
port eth0
    ip netflow 1
    exit
!
```

§4.12. Прикладные службы IP

§4.12.1. Ping и Traceroute

Программное обеспечение NSG Linux содержит утилиты ping и traceroute для контроля прохождения пакетов в IP-сети. Вызов утилит со стандартными параметрами производится из меню простого или привилегированного режима:

```
nsg> ping { ip-адрес | имя-хоста }
nsg> traceroute { ip-адрес | имя-хоста }
```

Для обращения по имени хоста необходимо, чтобы на устройстве была включена служба DNS (в данной версии NSG Linux не поддерживается).

Вызов этих утилит с расширенным набором параметров (размер пакетов и т.п.) возможен непосредственно из командной оболочки ОС Linux.

§4.12.2. Netping

Функция netping позволяет организовать периодическую проверку доступности заданного хоста в Интернет или корпоративной сети. Через заданное время устройство NSG посылает на него серии пакетов ICMP Echo Request (до первого успешного ответа, для экономии трафика). Если за заданное число запросов ответ не получен ни разу, то попытка считается неудачной. Если заданное число попыток подряд оказывается неудачными, то хост считается недоступным и выполняется сценарий, установленный для этого случая. Например, можно принудительно поставить основной IP-интерфейс в состояние DOWN, чтобы направить трафик по резервному маршруту. При этом netping продолжает выполняться; как только хост снова станет доступным, может быть выполнен другой установленный сценарий.

В отличие от механизмов *keepalive*, протоколов управления канального уровня (в PPP, Cisco-HDLC, Frame Relay) и детектирования физического сигнала на интерфейсе, функция netping реализована на сетевом уровне протокольного стека. Таким образом, она позволяет контролировать не только соединение со смежным устройством сети, но и целиком весь маршрут до заданного хоста (или некоторую проблемную его часть, например, участок "последней мили" и местного поставщика услуг Интернет).

Настройка netping производится в меню (config-nsg)# с помощью следующей команды:

netping <1...255>
no netping <1...255>
: Создание агента netping и вход в меню его конфигурации, и удаление агента, соответственно. Всего в устройстве может быть определено до 255 агентов.

Дальнейшая настройка производится в меню агента netping:

adm-state { up | down }
: Административное состояние агента: включён/выключен. По умолчанию, вновь созданный агент находится в состоянии UP.

description <строка>
: Текстовое описание агента, для удобства администрирования.

destination <ip-адрес>
: IP-адрес удалённого хоста, доступность которого требуется контролировать. Параметр обязательный.

source < ip-адрес >
: IP-адрес, который будет указываться в исходящих запросах ICMP Echo Request в качестве адреса источника. Адрес должен принадлежать одному из интерфейсов данного устройства.
Если адрес не указан, то по умолчанию *source address* устанавливается равным IP-адресу того интерфейса, с которого отправляется пакет в соответствии с текущей таблицей маршрутизации.

interval <1...64000>
: Интервал между попытками. Значение по умолчанию — 10 сек.

packets <1...100>
: Максимальное число запросов, посылаемых в ходе одной попытки. Как только получен пакет ICMP Echo Reply, попытка считается успешной и прекращается. Если на указанное число пакетов ICMP Echo Request не получено ни одного ответа, попытка считается недачной. Значение по умолчанию — 3. Время ожидания ответа в данной версии NSG Linux не настраивается и равно 10 сек.
При значении 1 получается просто одиночный *ping*, посылаемый через interval секунд.

retry <1...100>
: Максимальное число попыток. Если столько попыток подряд оказываются неудачными, то детектируется состояние отказа и выполняется сценарий, заданный параметром failure-script.
Значение по умолчанию — 5.

start-delay {<1…604800> | hour | day | week }
: Задержка перед началом посылки *ping*. Отрабатывается при старте системы, а также каждый раз после отработки failure-script. Предназначена для того, чтобы дать время на поднятие интерфейсов (особенно сотовых, требующих длительного времени на старт/рестарт модуля и регистрацию в сети). Таким образом, можно избежать ложных срабатываний оттого, что *netping* уже запущен, а соединение ещё не установлено. Значение по умолчанию — 10 сек.

failure-script {<0…1000> | start }
: Номер сценария, выполняемого при обнаружении недоступности хоста, т.е. при retry неудачных попыток подряд.

ВНИМАНИЕ В данной версии NSG Linux failure-script выполняется как при первом обнаружении отказа, так и впоследствии каждый раз после retry попыток до тех пор, пока прохождение пинга не возобновится. Если требуется, чтобы он выполнялся только один раз, то в скрипте необходимо сделать проверку текущего состояния системы по какому-либо параметру, позволяющему различить состояние до исполнения failure-script и после него. Если такого параметра нет, то нужно скриптом выставлять некоторый флаг в отдельном файле.

restore-script {<0…1000> | start }
: Номер сценария, выполняемого при восстановлении доступности хоста, т.е. при первой удачной попытке после того, как был детектирован отказ.

test-script {<0…1000> | start }
: Номер сценария, который должен использоваться для выполнения проверки вместо простого *ping*. Предназначен для задач, в которых пользователь нуждается в иных, или более сложных, средствах контроля. Например, такой скрипт может содержать утилиту *traceroute* с ключами -f, -I и -w. Или же с его помощью может вызываться некоторая утилита, устанавливающая соединение по заданному порту TCP и в случае успеха немедленно разрывающая его; таким образом, можно удостовериться в работоспособности не только сети и стека IP на удалённой стороне, но и конкретного приложнения на ней.

Для того, чтобы агент netping запустился необходимо, чтобы был указан, как минимум, один из сценариев failure-script и restore-script. Внутри сценария, обрабатывающего события, могут быть использованы следующие переменные окружения:

NET_PING_EVENT со значением FAILURE либо RESTORE
NET_PING_DESTINATION_IP
NET_PING_SOURCE_IP

Время ожидания пинга не настраивается и равно 10сек. Однако вместо *ping* можно использовать (посредством test-script) другие утилиты, например, *traceroute*, предоставляющие такую возможность.

Во всех трёх последних параметрах вместо номера сценария 0 может быть использовано ключевое слово start для обозначения скрипта, выполняемого при старте системы. Два или более сценариев могут быть подключены последовательно при помощи параметра next N.

Подробно о сценариях Linux см. Часть 6.

ВНИМАНИЕ Если *netping* используется для рестарта интерфейса или всего устройства, то необходимо использовать параметр start-delay. Или же, как минимум, время срабатывания *netping* должно быть гарантированно больше, чем время, необходимое для инициализации интерфейса и установления соединения. (В противном случае интерфейс будет рестартовать бесконечно.) Рекомендуется, чтобы они отличались не менее чем в 2–3 раза.

ПРИМЕЧАНИЕ В данной версии NSG Linux имеется известная проблема: netping не позволяет манипулировать составом bridge groups (включать и исключать из них порты и другие объекты).

Пример. Через каждые 600 секунд будет выполняться ping из не более чем 3 пакетов на адрес 11.0.0.52 с обратным адресом 10.0.0.1. В случае, если 2 попытки окажутся неудачными, будет выполнен однократно сценарий 10. После этого при первой же удачной попытке пинга будет однократно выполнен сценарий 20. Оба сценария записывают событие в файл /var/log/netping_1.log.

```
script add 10 "echo `date` DOWN > /var/log/netping_1.log"
script add 20 "echo `date` UP > /var/log/netping_1.log"
netping 1
    adm-state up
    destination 11.0.0.52
    source 10.0.0.1
    interval 600
    packets 3
    retry 2
    restore-script 20
    failure-script 10
    exit
```

§4.12.3. Сервер и ретранслятор DHCP

Служба DHCP (Dynamic Host Configuration Protocol) обеспечивает автоматическое назначение IP-адресов, маски, адреса шлюза по умолчанию и другой ключевой информации хостам локальной сети. (В более общем случае — любой сети второго уровня, включая удаленные сегменты.) При включении хост-клиент рассылает по сети широковещательный запрос, содержащий единственную доступную ему на данный момент информацию — его собственный MAC-адрес в качестве адреса источника. Сервер DHCP, получив такой запрос, посылает на этот MAC-адрес ответ со всей информацией, предписанной для данного хоста индивидуально или по умолчанию. Получив ответ, хост настраивает свой IP-интерфейс согласно принятой информации, и далее он готов к работе по протоколу IP. Кроме того, если клиент представляет собой бездисковую станцию, то он получает IP-адрес сервера BOOTP или TFTP, на котором находится загрузочный файл для него, имя файла, и загружается по сети.

ПРИМЕЧАНИЕ Не следует включать два или более серверов DHCP одновременно в одной сети. В противном случае результаты могут быть непредсказуемы.

Для настройки серверов DHCP используются следующие команды в меню (config-nsg)#:

dhcp <1...255>
no dhcp <1...255>
: Создание/изменение и удаление сервера DHCP, соответственно. В устройстве могут быть одновременно определены несколько серверов DHCP, каждый из которых функционирует на своем IP-интерфейсе.

Дальнейшая настройка производится в меню указанного сервера DHCP (config-dhcp-N)#:

description <описание>
: Текстовое описание сервера, для удобства администрирования. Максимальная длина описания — 255 символов. Значение по умолчанию — пустая строка.

adm-state { up | down }
: Административное состояние сервера. По умолчанию все вновь создаваемые серверы включены.

mode { server | relay }
: Режим работы службы DHCP: сервер (по умолчанию) или ретранслятор.

Дальнейший список параметров зависит от выбранного режима. При работе в качестве ретранслятора требуются два обязательных параметра:

client-interface <имя>
no client-interface <имя>
: Имя интерфейса, на котором должен работать ретранслятор. Ретранслятор может работать на нескольких интерфейсах одновременно, в этом случае каждый из них вводится и удаляется отдельной командой.

server-interface <имя>
no server-interface <имя>
: Имя интерфейса, подключенного к сети, в которой находится сервер DHCP.

При работе в качестве сервера DHCP определяется один и только один интерфейс, на котором он должен работать. Если устройство NSG должно обслуживать несколько раздельных сетей, то для каждой из них следует создать в системе свой сервер DHCP с непересекающимися диапазонами IP-адресов.

interface <имя>
no interface <имя>
: Имя интерфейса, на котором должен работать данный сервер. Параметр обязательный.

Далее необходимо определить параметры, передаваемые клиентам. Их можно разделить на несколько функциональных групп. По умолчанию все эти параметры имеют значение no либо пустая строка (""), означающие, что данный параметр не передается.

а) Общие параметры IP. Ключевыми для работы клиентов являются, как правило, следующие настройки:

ip-address <начальный_ip-адрес> through <конечный_ip-адрес>
: Диапазон IP-адресов, назначаемых клиентам. Данный параметр является обязательным.

mask-length { <0...32> | no }
: Длина маски сети, передаваемая клиентам. При значении no (установлено по умолчанию) длина маски не передается, в этом случае клиент может устанавливать себе маску самостоятельно в соответствии с алгоритмом, заданным в его собственном программном обеспечении (как правило — в соответствии с классом сети).

gateway {<ip-адрес> | no }
Адрес шлюза по умолчанию.

dns1 {<ip-адрес> | no }
dns2 {<ip-адрес> | no }
Адреса первичного и вторичного серверов DNS.

ПРИМЕЧАНИЕ Адреса DNS, передаваемые клиентам, в общем случае никак не связаны с адресами DNS, которые используются самим устройством NSG. Например, если маршрутизатор NSG используется для подключения локальной сети по сеансовому соединению PPP (через коммутируемое модемное соединение, GPRS, CDMA и т.п.), то обычно он должен служить ретранслятором DNS для локальных клиентов: принимать запросы и переадресовывать их на DNS, динамически назначаемые оператором при установлении PPP-соединения. В этом случае в качестве dns1 следует указать IP-адрес самого устройства NSG.

broadcast {<ip-адрес> | no }
Широковещательный адрес. Данный параметр полезен в случае, если длина маски клиенту не передается.

б) Управление назначением и аннулированием IP-адресов:

lease-time <1...4294967295>
Срок использования IP-адресов (в секундах). По истечении этого срока клиент DHCP обязан запросить новую конфигурацию, даже если он все это время находился в сети. Значение по умолчанию — 86400 сек (1 сутки).

max-leases <1...65535>
Максимальное число динамических IP-адресов, выдаваемых одновременно. Это число может быть меньше, чем размер пула. Сервер автоматически контролирует наличие в сети хостов с адресами, попадающими в динамически выделяемый пул. Если на момент поступления DCHP-запроса в сети уже имеется хост с таким адресом, сервер пропустит его и будет пытаться выделить следующий адрес. Ограничение max-leases позволяет определить пул "с запасом", т.е. заведомо больше, чем реально требуемое число динамических адресов.
От данного параметра зависит объем оперативной памяти, занимаемой сервером. Значение по умолчанию — 254.

static add <ip-адрес> <mac-адрес>
Добавление статической записи, т.е. жестко заданной пары IP–MAC. При запросе от клиента с указанным MAC-адресом ему всегда будет назначаться этот и только этот MAC-адрес; сервер автоматически исключает статически указанные адреса из общего пула.
Определяющим в данной паре является IP-адрес, т.е. нельзя создать две записи с одним IP.
Создать две или более записей с разными IP-адресами для одного MAC-адреса — возможно, но такая конфигурация будет по сути некорректной; в этом случае действовать будет только последняя введенная запись.

static delete [<ip-адрес>]
Удаление записи для указанного IP-адреса. Если команда введена без адреса, выводится текущий список соответствий между IP- и MAC-адресами.

ПРИМЕЧАНИЕ Команда static сама по себе представляет подузел меню. При настройке больших таблиц соответствия удобно войти в данное меню и использовать команды add, delete, а также display и другие общие команды.

ВНИМАНИЕ Сервер DHCP не проверяет наличие хостов с адресами, содержащимися в статической таблице. Если статический адрес уже занят (например, установлен на некотором хосте вручную), в сети возникнет конфликт IP-адресов.

в) Параметры BOOTP и TFTP. Используются, если клиенты представляют собой станции, загружаемые по сети:

siaddr {<ip-адрес> | no }
sname <имя>
IP-адрес и имя (до 255 знаков) сервера BOOTP, на котором находится загрузочный файл.

tftp-boot-server1 {<ip-адрес> | no }
tftp-boot-server2 {<ip-адрес> | no }
IP-адреса основного и резервного серверов TFTP, на которых находится загрузочный файл.

boot-file <имя>
boot-size {<1...4294967295> | no }
Имя загрузочного файла (до 255 знаков) и его размер (в байтах).

г) Дополнительные параметры клиентской системы:

domain-name <имя>
Имя домена, которое клиенту следует использовать при разрешении имен с помощью DNS.

host-name <имя>
Имя, назначаемое клиенту. Имя не обязательно должно коррелировать с именем домена.

root-path <строка>
Путь в директорию, которая должна быть смонтирована на клиенте как корневая.

д) Параметры других вспомогательных служб:

wins1 {<ip-адрес> | no }
wins2 {<ip-адрес> | no }
Адреса первичного и вторичного серверов WINS (Windows Internet Name Service).

ntp-server {<ip-адрес> | no }
Адрес сервера NTP (Network Time Protocol, RFC–1305).

time-server {<ip-адрес> | no }
Адрес сервера системного времени (RFC 868). В отличие от NTP, данный сервер сообщает время в формате числа секунд от 0 часов 1 января 1900 г.

timezone {<-86400...86400> | no }
Сдвиг времени относительно согласованного мирового времени (UTC), в секундах. Восточному полушарию соответствуют положительные значения, западному — отрицательные.

е) Другие опции DHCP:

Для настройки произвольных параметров BOOTP/DHCP, в том числе адресов редко используемых прикладных серверов (*cookie-server*, *lpr-server*, *log-server* и др), унаследованных служб и фирменных полей пакета DHCP, не перечисленных выше, в общем случае можно использовать следующую команду:

option <1...255> <имя> <формат> <значение>
Установка произвольной опции DHCP, передаваемой клиенту. Первый параметр представляет собой номер опции в пакете DHCP. Второй параметр — имя опции; оно может быть произвольным, поскольку представляет собой не более чем ее текстовое описание. Третий параметр определяет формат, в котором будет вводиться значение опции:

hex	Шестнадцатеричное значение
int16	Целое число в диапазоне –32768 … 32767
int32	Целое число в диапазоне –2147483648 … 2147483647
ip	IP-адрес в дотовой десятичной нотации
string	Произвольная строка символов
uint8	Целое число в диапазоне 0 … 255
uint16	Целое число в диапазоне 0 … 65535
uint32	Целое число в диапазоне 0 … 4294967295

Последний параметр устанавливает собственно значение опции, которое должно вводиться в выбранном формате. В общем случае любые опции могут вводиться в шестнадцатеричном виде, поскольку именно в таком виде они хранятся в системе и передаются клиентам; остальные форматы предназначены только для удобства пользователя. Например, стандартную опцию — адрес сервера *syslog* — можно определить следующим образом:

```
option 7 log-server ip 123.45.67.89
```

При помощи команды option можно определить, в частности, любые опции, перечисленные выше. При этом, если для некоторой опции уже установлено некоторое непустое значение, то значение, введенное в команде option, добавляется к нему, а не вместо него.

ПРИМЕЧАНИЕ Рекомендуется использовать команду option только для тех опций DHCP, для которых не предусмотрены отдельные команды. Не рекомендуется переопределять стандартные опции, регламентированные RFC–2132.

§4.12.4. Клиент DHCP

Клиент DHCP позволяет настроить параметры IP-интерфейса и некоторые параметры всего стека IP в целом при помощи удалённого сервера. Управление клиентом производится следующими командами в меню IP-интерфейса (порта, туннеля, FR DLCI и т.п.):

ip address dhcp
no ip address dhcp
: Включение и выключение клиента DHCP на интерфейсе.

Команда доступна в меню IP-интерфейсов всех типов — как широковещательных, так и "точка-точка". При включении клиента DHCP удаляются все адреса, назначенные данному интерфейсу статически, и наоборот, при назначении статического IP-адреса клиент DHCP выключается.

ip dhcp client request [route]
: Выбор параметров, которые могут быть избирательны получены данным интерфейсом по DHCP:
  route — Адрес шлюза по умолчанию. При этом в таблицу маршрутизации добавляется соответствующая запись. Запись существует до тех пор, пока данный интерфейс находится в состоянии UP.

  Если указанные параметры присутствуют в списке, то для них принимаются значения, полученные от сервера DHCP. Если параметр отсутствует, то значение, присланное сервером, игнорируется. По умолчанию, все вышеперечисленные параметры не принимаются.

В данной версии NSG Linux настраивается выбор только одного параметра — шлюза по умолчанию; помимо этого, клиент DHCP воспринимает от сервера следующие общие параметры:

— IP-адрес и маску подсети (принимаются безусловно).

— Широковещательный адрес (принимается безусловно, если присутствует в ответе сервера).

— Адреса серверов DNS. Для того, чтобы они возымели действие, необходимо установить в меню клиента/ретранслятора DNS указание на данный интерфейс (см. п.4.12.6), например:

```
port eth0
    ip address dhcp
    exit
dns client
    update-from eth0
    exit
```

§4.12.5. Клиент NTP

Клиент NTP (Network Time Protocol v.3, RFC–1305) позволяет синхронизировать системное время от заданного сервера сетевого времени. Для входа в меню клиента NTP используется следующая команда в меню (config-nsg)#:

ntp client — Вход в меню клиента NTP.

Дальнейшие настройки производятся в меню (config-ntp-client)# при помощи команд:

description <описание>
: Текстовое описание клиента, для удобства администрирования. Максимальная длина описания — 255 символов. Значение по умолчанию — пустая строка.

adm-state { up | down }
: Включение и выключение клиента сетевого времени, соответственно. По умолчанию, клиент выключен.

host { <имя_сервера> | <ip-адрес> }
: Имя или адрес используемого сервера NTP. Если сервер указывается именем, то в устройстве должен быть включён и настроен клиент DNS (см. п.4.12.6). По умолчанию установлено имя сервера 2.ru.pool.ntp.org.

period { <30 … 604800> | day | hour | week }
: Интервал между обращениями к серверу, в секундах либо 1 раз в сутки/час/неделю, соответственно. По истечении указанного времени клиент начинает слать запросы 1 раз в 5 секунд, пока не получит ответ, после чего снова "засыпает" до следующей процедуры синхронизации. Значение по умолчанию — 1 сутки.

§4.12.6. Клиент и ретранслятор DNS

Служба DNS обеспечивает преобразование алфавитно-цифровых имен хостов в IP-адреса. Служба включает:

Клиент — Обеспечивает разрешение IP-адресов для локальных служб устройства NSG, например, для команды ping remote.server.name.

Ретранслятор — Выступает в качестве сервера DNS (например, для компьютеров локальной сети), однако не разрешает IP-адреса самостоятельно, а ретранслирует запросы указанным серверам DNS. Полученные ответы ретранслируются обратно клиентам, а также кэшируются для ответов на повторные запросы.

Клиент Dynamic DNS — Регистрирует устройство NSG с его текущим IP-адресом на сервере Dynamic DNS.

Вход в меню настройки указанных компонент производится из меню (config-nsg)# следующими командами:

dns client — Вход в меню клиента DNS.

dns proxy — Вход в меню ретранслятора DNS.

Дальнейшая настройка осуществляется отдельно для каждой из указанных компонент. Настройка клиента и ретранслятора описана ниже, настройка клиента Dynamic DNS — в следующем параграфе.

а) Настройка клиента DNS.

name-server <ip-адрес> [1 | 2 | 3]
no name-server <ip-адрес>

Статическое добавление и удаление сервера DNS, соответственно. Всего клиент может работать с максимум 3 серверами. Последний опциональный параметр — приоритет данного сервера в списке. По умолчанию, каждый вновь добавляемый сервер помещается в конец списка.

update-from <имя_интерфейса>
no update-from <имя_интерфейса>

Указание/отмена имени интерфейса, от которого клиенту DNS следует получать адреса серверов DNS, назначаемые извне. Интерфейс получает эти адреса при старте (по DHCP или протоколам семейства PPP, при соответствующих настройках) и далее передает их клиенту DNS. Полученные таким образом адреса серверов DNS действительны, пока данный интерфейс находится в состоянии UP.

Для получения адресов серверов DNS могут быть указаны несколько интерфейсов (каждый — отдельной командой). В этом случае используются адреса от того интерфейса, который последним перешел в состояние UP; все остальные адреса игнорируются. Если он переходит в состояние DOWN, то используются адреса от предыдущего интерфейса. Если в качестве источника адресов не выбрано ни одного интерфейса, или все они находятся в состоянии DOWN, то используются адреса, заданные статически.

timeout <1...50>
retry <1...5>

Время ожидания ответа от сервера, в секундах, и число попыток обращения к серверам, соответственно. Значения по умолчанию — 5 сек. и 2 попытки. Если за указанное время ответ не получен, будет послан новый запрос следующему серверу в списке, пока список не будет исчерпан. После этого запрос снова будет послан первому серверу в списке и т.п. Когда весь круг будет пройден retry раз, попытки разрешить данное имя прекращаются, но клиент пытается сформировать новое имя с использованием суффиксов domain и search (см. ниже) и разрешить их.

rotate { yes | no }

Алгоритм выбора серверов DNS:

- no — Клиент всегда будет обращаться к заданным серверам последовательно, начиная с первого в списке. (Установка по умолчанию.)
- yes — Три сервера DNS будет циклически меняться местами в списке, так что каждый следующий запрос будет сначала направляться новому серверу.

check-names { yes | no }

Проверка запрашиваемых имен хостов.

- no — Проверка не производится. (Установка по умолчанию.)
- yes — Если запрашиваемое имя хоста содержит недопустимые символы, запрос будет отвергнут.

domain <имя>

Имя локального домена. Максимальная длина — 255 символов. Если имя не задано, то клиент будет пытаться вычленить имя домена из административного имени устройства (hostname) и возьмет в этом качестве часть имени, находящуюся справа от крайней левой точки. Например, если устройство имеет имя nsg.mydomain.xxx, то в качестве имени домена будет взято mydomain.xxx.

search <имя> <имя> ...
Список суффиксов, которые могут добавляться к запрашиваемому имени. Элементы списка разделяются пробелами. Максимальная длина списка — 255 символов. По умолчанию список содержит одно значение — имя локального домена (см. ниже).

Иерархия имен доменов и хостов предполагает, что имена хостов, расположенных в локальном домене, известны локальному серверу, например, установлены вручную. При этом они могут указываться в запросах в короткой форме, т.е. без указания имени домена — например, theotherhost. (Говоря более строго, имя не содержит точек.) Если локально разрешить такое имя не удается, то служба DNS пытается разрешить имя otherhost.domain на внешних серверах DNS.

В более общем случае, если параметр search задан явно, ищутся имена otherhost.search1, otherhost.search2 и т.д. Параметры domain и search являются взаимоисключающими.

б) Настройка ретранслятора DNS.

dns proxy adm-state { up | down }
Включение и выключение DNS relay/proxy, соответственно. Ретранслятор включается одной командой на всех интерфейсах устройства.

dns proxy nameserver { auto | <ip-адрес1> [secondary <ip-адрес2>] }
Выбор DNS-серверов, используемых ретранслятором:
- auto — Автоматический выбор DNS-серверов из числа установленных вручную для клиента DNS, а также полученных автоматически посредством любого из поддерживаемых механизмов. Например, адреса DNS могут быть получены в ходе установления PPP-соединения (см. выше).
- nameserver... — Фиксированный список из одного или двух DNS-серверов. При этом правила работы со списком (количество попыток, порядок ротации и т.п.) наследуются из локального настроенного клиента (см. выше). В этом случае адреса серверов, используемых для ретрансляции, могут быть никак не связаны с адресами серверов, используемых локальным клиентом DNS.

ПРИМЕЧАНИЕ В данной версии NSG Linux ретранслятор DNS работает только по протоколу UDP. Это необходимо учитывать при настройке NAT и фильтрации пакетов.

Пример. Совместная настройка служб DHCP server и DNS relay на устройстве, используемом для подключения малого офиса по сети CDMA. Адреса DNS априори неизвестны и назначаются устройству оператором сети. Для клиентов, находящихся в локальной сети, шлюзом по умолчанию и сервером DNS является устройство NSG.

```
!
    nsg
        virtual-template 1
            ppp ipcp accept-dns yes
            .....
            exit
        port s1
            encapsulation ppp
            virtual-template 1
            nat source masquerade
            .....
            exit
        port eth0
            encapsulation ethernet
            ip address 192.168.0.1/24
            exit
        dns client update-from s1
        dns proxy
            adm-state up
            nameserver auto
            exit
        dhcp 1
            ip-address 192.168.0.2 through 192.168.0.21
            mask-length 24
            gateway 192.168.0.1
            dns1 192.168.0.1
            max-leases 20
            exit
!
```

§4.12.7. Клиент Dynamic DNS

Служба Dynamic DNS (DynDNS, DDNS) позволяет хосту, IP-адрес которого априори неизвестен, зарегистрироваться на специализированном сервере DNS (для входа на сервер клиент аутентифицируется с помощью имени и пароля) и установить соответствие между своим именем и своим IP-адресом. После этого любые третьи хосты могут обратиться к DNS-серверу и определить текущий IP-адрес данного хоста по его имени.

Использование Dynamic DNS актуально для устройств, динамически получающих IP-адреса при каждом подключении к Интернет или корпоративной интрасети. В первую очередь, это относится к сеансовым соединениям PPP dial-up, PPP по сотовым сетям в пакетном режиме (GPRS, CDMA, 3G), PPPoE по городским сетям Ethernet и линиям ADSL, и др. Удалённые филиалы организации, подключённые таким образом, с помощью DynDNS становятся полноправными узлами корпоративной сети, доступными для обращений со стороны центрального офиса.

В качестве серверов DynDNS могут использоваться различные платные и бесплатные серверы, предлагающие свои услуги в Интернет. Для корпоративных приложений, как правило, предпочтительно — с точки зрения надёжности и безопасности — организовать собственный сервер DynDNS на границе своей интрасети.

Для определения собственного IP-адреса клиент DynDNS использует тот же или сторонний сервер, сообщающий ему, под каким адресом он виден из Интернет. Как определение IP-адреса, так и регистрация его на сервере DynDNS производятся с помощью HTTP-запросов. Формат этих запросов, в общем случае, не регламентирован стандартами и является специфичным для того или иного сервера.

Для настройки клиента DynDNS используются следующие команды в меню (config-nsg)#:

dns dyndns <1…255>
dns no dyndns <1…255>
: Создание клиента DynDNS и вход в меню его настройки, и удаление клиента DynDNS, соответственно. Одновременно в устройстве может быть создано несколько клиентов DynDNS.

Дальнейшая настройка производится в меню клиента DynDNS:

adm-state { up | down }
: Включение и выключение клиента DynDNS, соответственно.

dyndns-system <имя>
: Выбор поставщика услуг DynDNS и выбор базового алгоритма работы. Имя поставщика может иметь одно из следующих значений:

custom@http_svr_basic_auth	DNS-система в общем случае
default@dyndns.org	DNS-система dyndns.org, общий случай (значение по умолчанию)
customdns@dyndns.org	DNS-система dyndns.org, услуга Custom DNS
dyndns@dyndns.org	DNS-система dyndns.org, услуга Dynamic DNS
default@freedns.afraid.org	DNS-система freedns.afraid.org
default@no-ip.com	DNS-система www.no-ip.com
default@zoneedit.com	DNS-система www.zoneedit.com

Следующие два параметра используются в случае, если используется нестандартная система, например, собственный сервер DynDNS. Пример использования этих параметров см. ниже.

dyndns-server { auto | {<имя.сервера> | <ip-адрес>} port <номер> path </путь/скрипт>}
: Используемый сервер DynDNS. Если установлено auto (значение по умолчанию), то используется сервер, соответствующий выбранной системе. В противном случае необходимо указать явным образом сервер (задаётся алфавитно-цифровым именем или IP-адресом), номер порта TCP (по умолчанию 80) и локальный путь к скрипту, выполняющему регистрацию на сервере. Алгоритм обслуживания выбирается в зависимости от параметра dyndns-system.

ip-server { auto | {<имя.сервера> | <ip-адрес>} port <номер> path </путь/скрипт> }
: Используемый сервер определения IP-адреса. Если установлено auto (значение по умолчанию), то используется сервер, соответствующий выбранной системе. В противном случае необходимо указать явным образом сервер (задаётся алфавитно-цифровым именем или IP-адресом), номер порта TCP (по умолчанию 80) и локальный путь к скрипту, определяющему IP-адрес. Алгоритм получения адреса выбирается в зависимости от параметра dyndns-system; в общем случае используется первый IP-адрес, содержащийся в ответе этого сервера.

ВНИМАНИЕ Если сервер Dynamic DNS задан алфавитно-цифровым именем (что, в частности, имеет место для всех стандарных систем), то для работы службы DynDNS необходимо предварительно настроить обычного клиента DNS (см. п.4.12.6).

host-name <имя>
: Сетевое имя данного устройства NSG, для которого устанавливается IP-адрес. Это имя должно быть указано в учётной записи пользователя на сервере DynDNS. Максимальная длина — 64 символа.

ПРИМЕЧАНИЕ В общем случае, определяемое имя может не совпадать с именем, определённым как hostname данного устройства.

user-name <имя>

Имя пользователя, используемое для аутентификации на сервере DynDNS. (Т.е. имя владельца учётной записи.) Максимальная длина имени — 255 символов.
Пользователь с данным именем должен быть внесен в общий список пользователей на устройстве NSG (см. Часть 3) и иметь открытый (open или xor) пароль.

update-period {<10 ... 864000> | <интервал>}

Частота проверки IP-адреса. Если при очередной проверке обнаруживается изменение адреса, то происходит обновление информации на сервере DynDNS. Значение параметра может указываться либо в секундах, либо одним из ключевых слов:

10-seconds
20-seconds
30-seconds
1-minut
3-minuts
5-minuts
10-minuts
15-minuts
30-minuts
1-hours
2-hours
4-hours
8-hours
12-hours
16-hours
1-day
2-days
3-days
5-days
7-days
10-days

По умолчанию, IP-адрес проверяется через каждые 3 минуты.

ПРИМЕЧАНИЕ Вместо регулярной проверки изменения IP-адреса, или наряду с ней, для поддержания информации DynDNS в максимально актуальном состоянии можно использовать принудительный рестарт клиента DynDNS при изменении статуса IP-интерфейса (например, при разрыве и переустановлении PPP-соединения). Такой рестарт может быть реализован с помощью механизма state-script на требуемом интерфейсе (см. Часть 6).

forced-update-period {<10 ... 864000> | <интервал>}

Частота принудительной перерегистрации на сервере DynDNS. Обновление информации производится даже в том случае, если IP-адрес в течение длительного времени остаётся неизменным. Это необходимо для поддержания данной записи DynDNS в статусе активной.
Значение параметра может указываться либо в секундах, либо одним из ключевых слов — таких же, как и для update-period. По умолчанию, перерегистрация производится 1 раз в сутки.

ПРИМЕЧАНИЕ Распространение DNS-информации о доменах третьего уровня, которые предлагаются большинством систем DynDNS, занимает конечное время, на практике — около 5 минут. Предполагается, что средняя продолжительность сеанса составляет существенно бóльшую величину — например, порядка часа. Если специфика задачи требует более оперативного обновления информации, то в настойках сетевых служб на вызывающем компьютере следует указать в качестве DNS-сервера непосредственно сервер используемой системы.

Пример 1. Типовая конфигурация клиента DynDNS для сервера no-ip.com:

```
!
nsg
    users
        user-name "vasia" open "12345"
        exit
    dns dyndns 1
        user-name "vasia"
        host-name vasia.no-ip.com
        dyndns-system default@no-ip.com
        exit
!
```

Пример 2. Конфигурация для нестандартного сервера в корпоративной сети:

```
nsg
    users
        user-name "vasia" open "12345"
        exit
    dns dyndns 1
        user-name "vasia"
        ip-server "123.45.67.89" port 8080 path "/get_ip.php"
        dyndns-server "123.45.67.89" port 8080 path "/set_ip.php"
        dyndns-system custom@http_svr_basic_auth
        update-period 60
        forced-update-period 43200
        host-name "mycompany.router"
        exit
!
```

§4.12.8. Агент SNMP

Программное обеспечение NSG Linux содержит встроенного агента SNMP v1–3. Настройка агента описана в Части 1 данного руководства.

§4.12.8. Клиент RADIUS

Клиент RADIUS в данной версии NSG Linux обеспечивает аутентификацию пользователей, подключающихся к устройству, их авторизацию для пользования услугой PPP, и отсылку учетной информации на удаленный сервер RADIUS. Настройка RADIUS описана в Части 3 данного руководства.

§4.12.10. Клиент TACACS+

Клиент TACACS+ NSG Linux обеспечивает аутентификацию и авторизацию пользователей, подключающихся к устройству для доступа к физическим асинхронным портам по Reverse Telnet. Настройка TACACS+ описана в Части 3 данного руководства.

§4.12.11. Клиент VRRP

Клиент VRRP (Virtual Router Redundancy Protocol, IETF RFC–2338) предназначен для построения отказоустойчивых пулов маршрутизаторов. Два или более маршрутизаторов объединяются в виртуальный роутер, имеющий уникальный идентификатор (VRID), один или несколько IP-адресов (которые могут принадлежать одному из физических устройств, или не принадлежать одному из них) и виртуальный же MAC-адрес (не совпадающий ни с одним из MAC-адресов физических устройств). Каждому из устройств, входящих в состав виртуального роутера, назначается некоторый приоритет от 1 до 255. Устройство, имеющее наибольший приоритет, объявляется ведущим (Master) и начинает обрабатывать все пакеты с IP- и MAC-адресами виртуального устройства. Таким образом, весь трафик клиентов направляется через него.

Ведущий маршрутизатор регулярно рассылает ведомым *multicast*-сообщения, свидетельствующие об его работоспособности. Если он выходит из строя, то остальные члены группы сравнивают свои приоритеты, и устройство с наибольшим приоритетом снова становится ведущим.

Специальным является случай, когда IP-адрес виртуального роутера принадлежит одному из физических устройств, входящих в него. В этом случае оно всегда становится ведущим и имеет наивысший приоритет — 255.

Говоря более строго, членами виртуального роутера являются не физические устройства в целом, а их IP-интерфейсы. Интерфейсы должны быть широковещательного типа (физические порты Ethernet, VLAN). Один и тот же интерфейс может одновременно входить в состав нескольких виртуальных роутеров с различными VRID и адресами. Настройка VRRP выполняется в меню IP-интерфейсов NSG следующими командами:

ip vrrp <1...255>
: Создание виртуального роутера и вход в меню его настройки.

no ip vrrp <1...255>
: Выключение и удаление виртуального роутера. Если выключается ведущий маршрутизатор, то он рассылает остальным членам группы специальное оповещение со значением приоритета 0, чтобы побудить их приступить к немедленным выборам нового ведущего.

Меню настройки виртуального роутера содержит команды:

address <ip-адрес>
no ip address <ip-адрес>

Назначение и удаление IP-адресов интерфейсу виртуального роутера. Интерфейс может иметь до девяти IP-адресов (в данной версии NSG Linux). Если интерфейс физического роутера становится ведущим, то эти адреса добавляются к списку IP-адресов, присвоенных ему.

adm-state { up | down }

Административное состояние виртуального роутера. По умолчанию, все вновь созданные виртуальные роутеры включены.

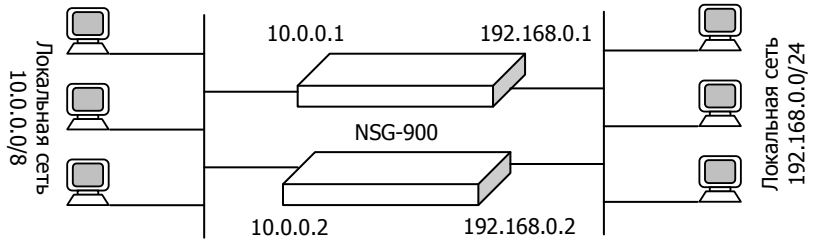
delay <1...255>

Интервал рассылки оповещений в том случае, если данный интерфейс работает как ведущий, в секундах. Значение по умолчанию — 1 сек. — позволяет минимизировать периоды неработоспособности виртуального роутера в целом.

prio <1...255>

Приоритет данного физического устройства перед остальными членами группы. Бóльшие значения соответствуют более высокому приоритету. Приоритет 255 следует назначать в том и только в том случае, если IP-адрес виртуального роутера принадлежит данному физическому интерфейсу; в остальных случаях приоритет может принимать значения от 1 до 254. Значение по умолчанию — 100.

Пример. Два устройства NSG–900/2WL с дополнительными сменными модулями Ethernet образуют виртуальный шлюз между двумя локальными сетями, резервируя друг друга. IP-адреса, статически назначенные их интерфейсам, показаны на рисунке. Для балансировки нагрузки при штатной работе обоих устройств на клиентских хостах указаны в качестве шлюзов по умолчанию примерно поровну 192.168.0.1 и 192.168.0.2, 10.0.0.1 и 10.0.0.2.



Конфигурация устройства А:

```
!
hostname alef
nsg
    card s1 im-et10
    port eth0
        ip address 10.0.0.1/8
        ip vrrp 1
            ip address 10.0.0.1
            prio 255
            exit
        ip vrrp 2
            address 10.0.0.2
            exit
        exit
    port s1
        ip address 192.168.0.1/24
        ip vrrp 1
            ip address 192.168.0.1
            prio 255
            exit
        ip vrrp 3
            ip address 192.168.0.2
            exit
        exit
    exit
!
```

Конфигурация устройства Б:

```
!
hostname bet
nsg
    card s1 im-et10
    port eth0
        ip address 10.0.0.2/8
        ip vrrp 1
            ip address 10.0.0.1
            exit
        ip vrrp 2
            ip address 10.0.0.2
            prio 255
            exit
        exit
    port s1
        ip address 192.168.0.2/24
        ip vrrp 1
            ip address 192.168.0.1
            exit
        ip vrrp 3
            ip address 192.168.0.2
            prio 255
            exit
        exit
    exit
!
```

В данном случае в одной локальной сети определены виртуальные роутеры с идентификаторами 1 и 2, в другой — 1 и 3. Виртуальные роутеры с идентификатором 1 существуют в обеих сетях, что не противоречит протоколу: это две независимые сущности, никак не пересекающиеся друг с другом.

§4.12.12. Принт-сервер

Встроенный сервер печати поддерживается в устройствах NSG, оснащенных встроенным или сменным портом USB. Сервер печати использует механизм Raw Socket Printing over TCP/IP port, идентичный технологии JetDirect компании Hewlett-Packard. На клиентском ПК настраивается сетевой TCP/IP принтер со следующими ключевыми параметрами:

IP-адрес	Адрес, назначенный устройству NSG
Номер порта TCP	9100 (по умолчанию), 9101 или 9102
Протокол печати	RAW (в ОС Windows) или JetDirect (в ОС Linux и других Unix-системах)

К порту USB устройства NSG может подключаться большинство моделей USB-принтеров (за исключением отдельных моделей Windows-принтеров), а также устройств, взаимодействующих с операционной системой через порт LPT или виртуальный LPT-over-USB (например, Web-камеры). К одному устройству NSG может быть подключено до 3 принтеров.

Принципиально важным обстоятельством является то, что драйвер принтера в любом случае работает на клиентском ПК, поэтому установка какого-либо программного обеспечения, специфичного для конкретной модели принтера, на устройстве NSG не требуется. Наряду с драйвером, полностью обеспечивается функциональность утилит мониторинга и управления, которыми комплектуется большинство современных принтеров. Обмен данными, по умолчанию, предполагается двусторонний. Подробно о процедурах сетевой печати см. документ NSG: *Настройка клиентов сетевой печати TCP/IP*.

Принт-сервер корректно обеспечивает совместный доступ многих пользователей к одному принтеру. Если в момент обращения к принтеру он уже занят заданием от другого пользователя, прием задания от нового пользователя будет отложен и возобновлен тогда, когда принтер выполнит предыдущие задания.

Для настройки принт-сервера необходимо, в первую очередь, указать тип интерфейсного модуля:

```
card sN um-usb
```

Дальнейшая настройка производится в меню физического порта (config-port-sN)#:

encapsulation { printer | unused }
: Выбор типа внешнего устройства, подключенного к порту USB (или его отмена, соответственно). Другие типы USB-устройств в данной версии NSG Linux не поддерживаются. После выбора инкапсуляции в меню появляются пункты, специфичные для выбранного класса устройств.

description <описание>
: Текстовое описание устройства, для удобства администрирования. Максимальная длина описания — 255 символов. Значение по умолчанию — пустая строка.

adm-state { up | down }
: Административное состояние устройства. По умолчанию все вновь создаваемые USB-устройства включены.

tcp-port { 9100 | 9101 | 9102 }
: Выбор номера порта TCP, на котором будет работать данный принтер. По умолчанию установлен TCP порт 9100. Если к устройству подключено более одного принтера, то для каждого из них необходимо назначить уникальный номер порта.

bidirectional { yes | no }
: Выбор двух- или одностороннего (от удаленного клиента к принтеру) режима передачи. По умолчанию установлен двусторонний режим. Если драйвер и программное обеспечение принтера не поддерживают его, можно установить односторонний.

ПРИМЕЧАНИЕ Применительно к данной версии NSG Linux, принтер должен быть включен до старта устройства, либо подключен один раз к работающему устройству. Если принтер отключается и подключается снова в процессе работы, то для его корректного опознания необходимо рестартовать устройство NSG. Данное ограничение является временным и будет устранено в последующих версиях.

ПРИМЕЧАНИЕ Функциональность принт-сервера может не быть совместимой с отдельными моделями, сериями и марками принтеров.

§4.12.13. Удаленное управление по Telnet и SSH

Для удаленного управления устройством могут использоваться встроенные серверы Telnet и SSH, принимающие входящие соединения на портах TCP 23 и 22, соответственно. Сервер Telnet не требует настройки. Настройка сервера SSH описана в Части 1 данного руководства.

§4.12.14. Telnet и Reverse Telnet

Устройство под управлением NSG Linux может использоваться для доступа из сети к физическому асинхронному порту посредством Reverse Telnet. Также возможно установление Telnet-соединения от "тупого" терминала, подключенного к асинхронному порту, к удаленному хосту в IP-сети (Telnet клиент). Для использования этих функций следует сконфигурировать порт с инкапсуляцией reverse-telnet либо telnet, соответственно. Подробно см. Часть 3 данного руководства.

Для ручного запуска клиента Telnet следует использовать следующую команду из меню простого или привилегированного режима:

```
nsg> telnet { ip-адрес | имя-хоста } [tcp-порт]
```

Для обращения по имени хоста необходимо, чтобы на устройстве была включена служба DNS (в данной версии NSG Linux не поддерживается).

Вызов клиента Telnet с дополнительными параметрами возможен непосредственно из командной оболочки ОС Linux.

§4.12.15. SSH клиент и Reverse SSH

Устройство под управлением NSG Linux может использоваться для безопасного доступа из сети к физическому асинхронному порту посредством протокола SSH. По аналогии с Reverse Telnet, такую функциональность уместно назвать Reverse SSH. Также возможно установление безопасного SSH-соединения от "тупого" терминала, подключенного к асинхронному порту, к удаленному хосту в IP-сети (SSH клиент). Обе эти функции настраиваются в данной версии NSG Linux непосредственно средствами ОС Linux (см. Часть 6).

Ручной вызов клиента SSH также производится непосредственно из командной оболочки ОС Linux.

§4.12.16. Другие прикладные службы

Помимо вышеперечисленных, в программном обеспечении NSG Linux имеется ряд других прикладных служб IP, таких как клиенты для передачи файлов по протоколам TFTP, FTP, HTTP. Данные службы доступны непосредственно средствами ОС Linux. Краткое описание их использования приведено в Части 6.