



МАРШРУТИЗАТОРЫ ЗЕЛАКС

Справочник команд

Приложение А Вывод статистики

© 2001-2005 Зелакс. Все права защищены.

Редакция 01 от 17.01.2005

Россия, 124681 Москва, г. Зеленоград, ул. Заводская, дом 1Б, строение 2
Телефон: +7 (095) 748-71-78 (многоканальный) • <http://www.zelax.ru/>
Техническая поддержка: tech@zelax.ru • Отдел продаж: sales@zelax.ru

16.02.2005

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ВЫВОД СТАТИСТИКИ СЕТЕВЫХ ПРОТОКОЛОВ И УСТРОЙСТВ	5
1.1 IP-статистика, выводимая командой show ip statistics	5
1.2 TCP статистика, выводимая командой show tcp statistics	8
1.3 ICMP статистика, выводимая командой show icmp statistics	11
1.4 UDP статистика, выводимая командой show udp statistics	13
1.5 Вывод информации о протоколах маршрутизации	14
1.6 Вывод информации о текущем состоянии таблицы маршрутизации службы RIP	17
1.7 Вывод таблицы маршрутизации	18
1.8 Статистика интерфейса Serial	20
1.9 Статистика интерфейса Ethernet	22
1.10 Статистика очереди, выводимая командой "show queue {ethernet serial}"	24
1.10.1 Вывод статистики при типе выходной очереди FIFOQ	24
1.10.2 Вывод статистики при типе выходной очереди RED	25
1.10.3 Вывод статистики при типе выходной очереди RIO	26
1.10.4 Вывод статистики при типе выходной очереди WFQ	28
1.10.5 Вывод статистики при типе выходной очереди PRIQ	29
1.10.6 Вывод статистики при типе выходной очереди CBQ	30
1.10.7 Вывод статистики при типе выходной очереди HFSC	34
1.11 Команды просмотра статистики и состояния объектов frame relay	36
1.11.1 Вывод состояния LMI интерфейса	36
1.11.2 Вывод состояния ARP таблицы интерфейса	37
1.11.3 Вывод состояния PVC	38
1.11.4 Вывод таблицы коммутации пакетов frame relay	40
ПЕРЕЧЕНЬ РИСУНКОВ	41
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	42

ВВЕДЕНИЕ

Контроль состояния коммуникационного процессора осуществляется командами вывода статистики.

В ПРИЛОЖЕНИИ А приведены команды вывода статистики, представлены примеры её вывода и дано описание полей выводимой информации.

1. ВЫВОД СТАТИСТИКИ СЕТЕВЫХ ПРОТОКОЛОВ И УСТРОЙСТВ

1.1 IP-статистика, выводимая командой *show ip statistics*

Данная статистика являет собой суммарную по всем IP интерфейсам статистику, определенную в рамках SNMP. Статистика накапливается системой сразу после старта и может быть обнулена командой **clear ip-stats**. Поля, ненулевое значение которых указывает на то, что пакет, скорее всего, был не обработан системой, помечены звездочкой.

Пример вывода статистики:

```
Cons>show ip statistics
```

```
Total packets received          : 8
Bad checksums                    : 0
Packets too short                 : 0
Not enough data                   : 0
IP header too short               : 0
IP packet smaller than IP hdr     : 0
Fragments received                : 0
Fragments dropped                 : 0
Fragments timeouted               : 0
Packets forwarded                 : 0
Packets NOT forwarded             : 0
Redirects sent                     : 0
Unknown protocol                  : 0
Delivered to kernel               : 8
Locally generated output          : 8
Lost packets                       : 0
Reassembled                       : 0
Fragmented                        : 0
Output fragments                  : 0
Can't fragment                    : 0
Bad IP options                    : 0
No route drops                    : 0
Bad IP version                    : 0
Raw IP packets generated          : 0
Too long IP packets               : 0
Access denied by FW               : 0
Switched to another interface     : 0
Dropped while switching           : 0
```

Описание полей:

- **Total packets received** – число IP пакетов, принятых всеми интерфейсами.
- ***Bad checksums** - число пакетов с неправильной IP контрольной суммой.
- ***Packets too short** - число пакетов, физическая длина которых меньше, чем указанная в заголовке.
- ***Not enough data** - в результате реасемблирования фрагментированного пакета не хватило данных в исходных фрагментах для построения результирующего пакета.
- ***IP header too short** - слишком короткий заголовок IP.
- ***IP packet smaller than IP hdr** - число пакетов, которые короче, чем минимальная длина IP заголовка.
- **Fragments received** - число принятых фрагментов.
- ***Fragments dropped** - число фрагментов, которые были отброшены системой по каким-либо причинам.
- ***Fragments timeouted** - число фрагментов, которые были слишком долго в очереди на дефрагментацию, и поэтому были отброшены системой.
- **Packets forwarded** - число пакетов, которые были маршрутизированы на другой хост.
- ***Packets NOT forwarded** - число пакетов, которые не были смаршрутизированы.
- **Redirects sent** - число посланных ICMP-REDIRECT сообщений.
- ***Unknown protocol** - число пакетов с неподдерживаемым типом инкапсулированного протокола.
- **Delivered to kernel** - число пакетов, доставленных ядру.
- **Locally generated output** - число пакетов, сгенерированных ядром маршрутизатора.
- ***Lost packets** - число потерянных пакетов.
- **Reassebled** - число пакетов, которые были удачно реасемблированы.
- **Fragmented** - число пакетов, которые были удачно фрагментированы.
- **Output fragments** - число фрагментов, посланных в сеть.
- ***Can't fragment** - число пакетов, которые система не смогла фрагментировать.
- ***Bad IP options** - число принятых пакетов с некорректными IP опциями в заголовке.
- ***No route drops** - число пакетов, отброшенных маршрутизатором по причине отсутствия маршрута.
- ***Bad IP version** - число пакетов с версией IP иной, чем 4.
- **Raw IP packets generated** - количество ``сырых" IP пакетов, сгенерированных ядром.

- ***Too long IP packets** - количество слишком длинных IP пакетов.
- ***Access denied by FW** - количество пакетов, не прошедших сетевой экран (Firewall).
- **Switched to another interface** - количество пакетов, переброшенных на другой интерфейс в обход стандартного механизма маршрутизации.
- ***Dropped while switching** - количество пакетов, которые не были сброшены.

1.2 TCP статистика, выводимая командой *show tcp statistics*'

Данная статистика является суммарной по всем интерфейсам статистикой, определенной в рамках SNMP. Статистика накапливается системой сразу после старта и может быть обнулена командой `clear tcp-stats`.

Пример вывода статистики:

```
Cons>show ip statistics
```

```
Connections initiated           : 1
Connections accepted           : 1
Connections established         : 2
Connections dropped             : 0
Conn. closed (includes drops)  : 0
Retransmit timeouts            : 0
Keepalive timeouts             : 0
Keepalive probes sent          : 0
Connections dropped in keepalive : 0
Total packets sent              : 8
Data packets sent               : 3
Data bytes sent                 : 84
Data packets retransmitted      : 0
Data bytes retransmitted        : 0
Ack-only packets sent           : 4
Window probes sent              : 0
Packets sent with URG only      : 0
Window update-only packets sent : 0
Control (SYN|FIN|RST) packets sent : 1
Total packets received          : 8
Packets received in sequence    : 3
Bytes received in sequence      : 84
Packets received with ccksum errs : 0
Packets received with bad offset : 0
Packets received too short      : 0
Duplicate-only packets received : 0
Duplicate-only bytes received   : 0
Packets with some duplicate data : 0
Dup. bytes in part-dup. packets : 0
Out-of-order packets received   : 0
Out-of-order bytes received     : 0
Packets with data after window  : 0
Bytes rcvd after window         : 0
Packets rcvd after "close"      : 0
```

Rcvd window probe packets	: 0
Rcvd duplicate acks	: 0
Rcvd acks for unsent data	: 0
Rcvd ack packets	: 5
Bytes acked by rcvd acks	: 85
Rcvd window update packets	: 0
Segments dropped due to PAWS	: 0
Bogus SYN, e.g. premature ACK	: 0
Resends due to MTU discovery	: 0
Listen queue overflows	: 0

Описание полей:

- **Connections initiated** - количество инициированных запросов на TCP соединение.
- **Connections accepted** - количество принятых запросов на TCP соединение.
- **Connections established** - количество установленных TCP соединений.
- **Connections dropped** - количество сброшенных соединений.
- **Conn. closed (includes drops)** - количество закрытых соединений (включая сброшенные).
- **Retransmit timeouts** - количество таймаутов перепосылок.
- **Keepalive timeouts** - кол-во таймаутов приема подтверждений TCP соединения.
- **Keepalive probes sent** - кол-во посланных подтверждений.
- **Connections dropped in keepalive** - кол-во сброшенных из-за не получения подтверждения соединений.
- **Total packets sent** - кол-во посланных TCP пакетов.
- **Data packets sent** - кол-во посланных TCP пакетов, содержащих данные.
- **Data bytes sent** - кол-во переданных байтов данных.
- **Data packets retransmitted** - кол-во пакетов с данными, которые перепосылались.
- **Data bytes retransmitted** - кол-во байтов данных, которые перепосылались.
- **Ack-only packets sent** - кол-во посланных ACK пакетов.
- **Window probes sent** - кол-во посланных запросов на размер TCP окна.
- **Packets sent with URG only** - количество пакетов, посланных с битом URG.
- **Window update-only packets sent** - количество пакетов, посланных для управления размером окна.
- **Control (SYN|FIN|RST) packets sent** - кол-во управляющих пакетов (с битами SYN/FIN/RST).
- **Total packets received** - кол-во принятых пакетов.
- **Packets received in sequence** - кол-во пакетов, принятых в том порядке, в котором они были посланы.
- **Bytes received in sequence** - кол-во байт, принятых в том порядке, в котором они были посланы.
- **Packets received with ccksum errs** - кол-во пакетов с ошибкой TCP контрольной суммы.

- **Packets received with bad offset** - кол-во пакетов с неправильным смещением.
- **Packets received too short** - кол-во слишком коротких пакетов.
- **Duplicate-only packets received** - кол-во принятых дубликатов пакетов.
- **Duplicate-only bytes received** - `--` байт.
- **Packets with some duplicate data** - пакеты с частично дублированными данными.
- **Dup. bytes in part-dup. packets** - количество дублированных байт в частично дублированных пакетах.
- **Out-of-order packets received** - кол-во пакетов, принятых не в том порядке, в котором они были посланы.
- **Out-of-order bytes received** - кол-во байт, принятых не в том порядке, в котором они были посланы.
- **Packets with data after window** - кол-во пакетов, с данными, выходящими за границы окна.
- **Bytes rcvd after window** - кол-во байт, вышедших за границы окна.
- **Packets rcvd after "close"** - кол-во пакетов, принятых после начала закрытия соединения.
- **Rcvd window probe packets** - кол-во принятых пакетов на запрос размера окна.
- **Rcvd duplicate acks** - кол-во принятых дублированных пакетов подтверждения.
- **Rcvd acks for unsent data** - кол-во принятых пакетов подтверждения для непосланных данных.
- **Rcvd ack packets** - кол-во принятых пакетов подтверждения.
- **Bytes acked by rcvd acks** - кол-во подтвержденных принятых байт.
- **Rcvd window update packets** - кол-во принятых пакетов, контролирующих размер окна.
- **Segments dropped due to PAWS** - кол-во сброшенных сегментов из-за превышения размера окна.
- **Bogus SYN, e.g. premature ACK** - кол-во ошибочных SYN пакетов.
- **Resends due to MTU discovery** - кол-во перепосланных пакетов из-за сообщения о превышении размера MTU.
- **Listen queue overflows** - переполнение очереди ожидания соединения.

1.3 ICMP статистика, выводимая командой `show icmp statistics`

Данная статистика является суммарной по всем IP интерфейсам статистикой, определенной в рамках SNMP. Статистика накапливается системой сразу после старта и может быть обнулена командой `clear icmp-stats`. Поля, ненулевое значение которых указывает на то, что пакет скорее всего был не обработан системой, помечены звездочкой.

Пример вывода статистики:

```
Cons>show icmp statistics
*** General ICMP statistics ***

Calls to icmp_error()                : 0
No error because old ip too short     : 0
No error because old packet was ICMP  : 0
ICMP code out of range received      : 0
*Too short packets                   : 0
*Bad checksum                         : 0
Calculated bounds mismatch           : 0
Number of responses                   : 1
*Broadcast/Multicast echo requests dropped : 0
*Broadcast/Multicast timestamp requests dropped : 0
```

*** Per-type ICMP statistics ***

TYPE	RECEIVED	SENT
ECHO-REPLY	: 0	1
UNREACH	: 0	0
SOURCE-QUENCH	: 0	0
REDIRECT	: 0	0
ECHO	: 1	0
ROUTER-ADVERT	: 0	0
ROUTER-SOLICIT	: 0	0
TTL-EXCEED	: 0	0
PARAM-PROBE	: 0	0
TIME-STAMP	: 0	0
TIME-STAMP-REPLY	: 0	0
INFORMATION	: 0	0
INFORMATION-REPLY	: 0	0
MASK	: 0	0
MASK-REPLY	: 0	0

Описание полей:

- **Calls to icmp_error()** - число раз вызова ядра на посылку ICMP сообщения об ошибке.
- **No error because old ip too short** - ICMP сообщение не было послано в ответ на IP пакет, потому что пакет был слишком короткий.
- **No error because old packet was ICMP** - ICMP сообщение не было послано в ответ на IP пакет, потому что в нем был инкапсулирован ICMP.
- **ICMP code out of range received** - принятый ICMP код не поддерживается.
- **Too short packets** - слишком короткий пакет.
- **Bad checksum** - неправильная контрольная сумма ICMP.
- **Number of responses** - количество ответов.
- **Broadcast/Multicast echo requests dropped** - количество сброшенных широковещательных запросов на эхо-ответ.
- **Broadcast/Multicast timestamp requests dropped** - количество сброшенных широковещательных запросов на засечку по времени.
- **ECHO-REPLY** - количество принятых/отправленных эхо-ответов (отправленных - ядром)
- **UNREACH** - количество принятых/отправленных сообщений о недостижимости (сети, порта и т.д.).
- **SOURCE-QUENCH** - количество принятых/отправленных сообщений о перегрузке сети.
- **REDIRECT** - количество принятых/отправленных сообщений о перенаправлении шлюза по умолчанию.
- **ECHO** - количество принятых/отправленных эхо-запросов (отправленных - ядром).
- **ROUTER-ADVERT** - количество принятых/отправленных сообщений о маршрутизаторах в сети.
- **ROUTER-SOLICIT** - количество принятых/отправленных сообщений Router Solicitation.
- **TTL-EXCEED** - количество принятых/отправленных сообщений о превышении времени жизни пакета.
- **PARAM-PROBE** - количество принятых/отправленных пакетов о проверке параметров.
- **TIME-STAMP** - количество принятых/отправленных запросов на засечку времени.
- **TIME-STAMP-REPLY** - количество принятых/отправленных ответов на засечку времени.
- **INFORMATION** - количество принятых/отправленных INFORMATION запросов.
- **INFORMATION-REPLY** - количество принятых/отправленных INFORMATION ответов.
- **MASK** - количество принятых/отправленных запросов о маске подсети.
- **MASK-REPLY** - количество принятых/отправленных ответов о маске подсети.

1.4 UDP статистика, выводимая командой `show udp statistics`

Данная статистика является суммарной по всем IP интерфейсам статистикой, определенной в рамках SNMP. Статистика накапливается системой сразу после старта и может быть обнулена командой `clear udp-stats`. Поля, ненулевое значение которых указывает на то, что пакет, скорее всего, был не обработан системой, помечены звездочкой.

Пример вывода статистики:

```
Cons>show udp statistics
```

```
Total input packets           : 93
*Packet shorter than header    : 0
*Checksum error                : 1
*Data length larger than packet : 0
*No socket on port             : 24
Of above, arrived as broadcast : 24
*Not delivered, input socket full : 0
Total output packets           : 23
```

Описание полей:

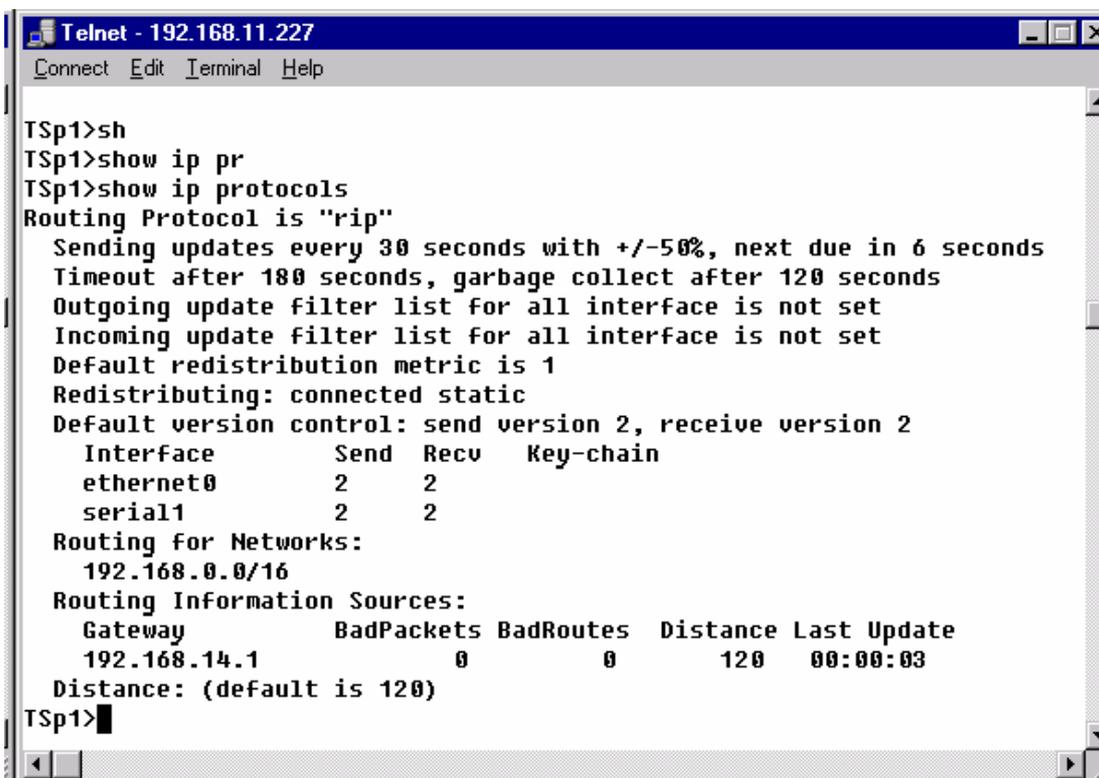
- **Total input packets** - кол-во принятых UDP пакетов.
- **Packet shorter than header** - кол-во пакетов короче, чем длина заголовка.
- **Checksum error** - кол-во пакетов с неправильной контрольной суммой UDP.
- **Data length larger than packet** - кол-во пакетов, длина данных которых больше, чем указано в заголовке.
- **No socket on port** - кол-во пакетов, для которых не нашлось сокета, "слушающего" данный порт.
- **Of above, arrived as broadcast** - кол-во широковещательных пакетов, для которых не нашлось сокета, "слушающего" данный порт.
- **Not delivered, input socket full** - кол-во отброшенных пакетов, по причине переполнения входного буфера сокета.
- **Total output packets** - кол-во посланных пакетов.

1.5 Вывод информации о протоколах маршрутизации

Вывод информации о состоянии службы RIP производится путем исполнения команды `show ip protocols` в корневом каталоге дерева команд коммуникационного процессора. Если служба RIP не запущена, никакая информация на экран не выводится, и появляется приглашение к вводу следующей команды. В противном случае на экран выводится следующая информация о текущей конфигурации и состоянии службы RIP.

- Значения таймеров периода рассылки маршрутов, времени жизни маршрутов и времени активности маршрутов.
- Фильтры на маршруты во входящих и исходящих сообщениях.
- Типы маршрутов, распространяемых протоколом.
- Значение метрики, с которой распространяются маршруты.
- Версия протокола RIP, используемая на интерфейсах.
- Группы адресов и имена интерфейсов, на которых работает служба RIP.
- Список обнаруженных соседних маршрутизаторов RIP.
- Дистанция динамических маршрутов, полученных по протоколу RIP.

Рассмотрим более подробно выводимую информацию (Рис. 1).



```
Telnet - 192.168.11.227
Connect Edit Terminal Help
TSp1>sh
TSp1>show ip pr
TSp1>show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 6 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: connected static
  Default version control: send version 2, receive version 2
    Interface          Send Recv  Key-chain
    ethernet0          2     2
    serial1            2     2
  Routing for Networks:
    192.168.0.0/16
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    192.168.14.1          0           0          120     00:00:03
  Distance: (default is 120)
TSp1>
```

Рис. 1 Вывод информации о протоколах маршрутизации.

Значения таймеров периода рассылки маршрутов, времени жизни маршрутов и времени активности маршрутов.

Сообщение имеет вид:

Sending updates every {период рассылки маршрутов} **seconds with +/-50%, next due in** {интервал времени до следующей рассылки} **seconds**
Timeout after {время жизни маршрута} **seconds, garbage collect after** {время активности маршрута} **seconds**

Информация о фильтрах на входящие и исходящие маршруты.

Сообщение имеет вид:

Outgoing update filter list for all interfaces is {номер списка доступа для исходящих маршрутов со всех интерфейсов}

{имя интерфейса} **is filtered by** {номер списка доступа для исходящих маршрутов с интерфейса {имя интерфейса}}

Incoming update filter list for all interfaces is {номер списка доступа для входящих маршрутов со всех интерфейсов}

{имя интерфейса} **is filtered by** {номер списка доступа для входящих маршрутов с интерфейса {имя интерфейса}}

Значение метрики по умолчанию.

Сообщение имеет вид:

Redistributing: {типы маршрутов}

Служба RIP может распространять следующие типы маршрутов:

- **static** – статические маршруты;
 - **connected** – автоматические маршруты, появляющиеся при присвоении адреса интерфейсу;
 - **ospf** – маршруты, полученные службой OSPF;
 - **bgp** – маршруты, полученные службой BGP.
- Информация о версии RIP и об интерфейсах, на которых она работает.

Сообщение имеет вид:

Default version control: send version {номер версии по умолчанию передаваемых пакетов}, **receive version** {номер версии по умолчанию принимаемых пакетов}

Interface	Send	Recv	Key-chain
-----------	------	------	-----------

В полях таблицы находится следующая информация:

- **Interface** – {Имя интерфейса}
- **Send** – {Версия RIP на передачу}
- **Recv** – {Версия RIP на прием}
- **Key-chain** – {Имя списка ключей аутентификации}

Список интерфейсов и сетей, на которых запущена служба RIP.

Сообщение имеет вид:

Routing for network:

*{Сеть в формате адрес/длина маски}
{Имя интерфейса}*

Список обнаруженных соседних маршрутизаторов RIP.

Сообщение выглядит в виде таблицы.

Routing Information Sources:

Gateway	BadPackets	BadRoutes	Distance	Last Update
----------------	-------------------	------------------	-----------------	--------------------

В полях таблицы содержится следующая информация:

- **Gateway** – адрес маршрутизатора;
- **BadPackets** – количество пакетов с ошибками;
- **BadRoutes** – количество неверных маршрутов;
- **Distance** – административная дистанция маршрутизатора;
- **Last Update** – интервал времени с последнего обновления маршрутов в формате часы:минуты:секунды.

Список административных дистанций.

Сообщение выглядит в виде таблицы:

Distance: (default is {дистанция по умолчанию})

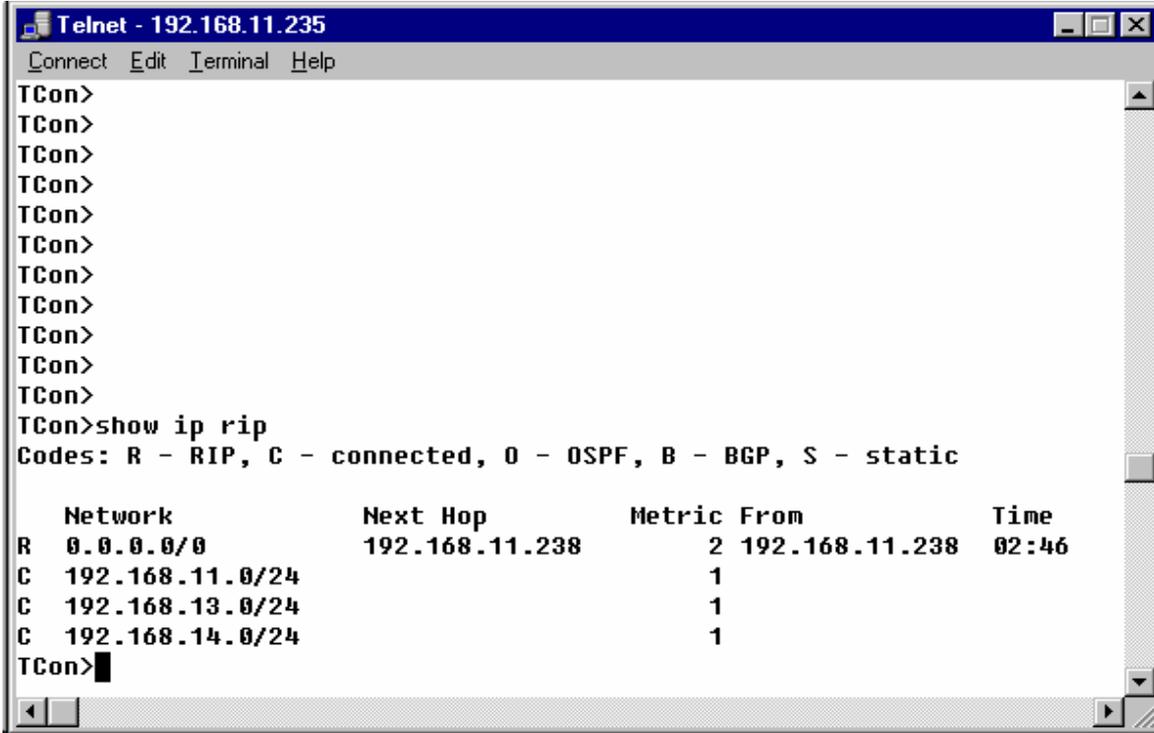
Address	Distance	List
----------------	-----------------	-------------

В полях таблицы содержится следующая информация:

- **Address** – адрес или группа адресов, для которых устанавливается дистанция;
- **Distance** – значение дистанции;
- **List** – дополнительный список адресов, для которых устанавливается дистанция.

1.6 Вывод информации о текущем состоянии таблицы маршрутизации службы RIP

Вывод информации о состоянии таблицы маршрутизации службы RIP производится путем исполнения команды `show ip rip` в корневом каталоге дерева команд коммуникационного процессора. Если служба RIP не запущена, никакая информация на экран не выводится, и появляется приглашение к вводу следующей команды. В противном случае на экран выводится список маршрутов, распространяемых службой RIP (Рис. 2).



```
Telnet - 192.168.11.235
Connect Edit Terminal Help
TCon>
TCon>show ip rip
Codes: R - RIP, C - connected, O - OSPF, B - BGP, S - static

  Network          Next Hop          Metric From          Time
R  0.0.0.0/0        192.168.11.238    2 192.168.11.238    02:46
C  192.168.11.0/24          1
C  192.168.13.0/24         1
C  192.168.14.0/24         1
TCon>
```

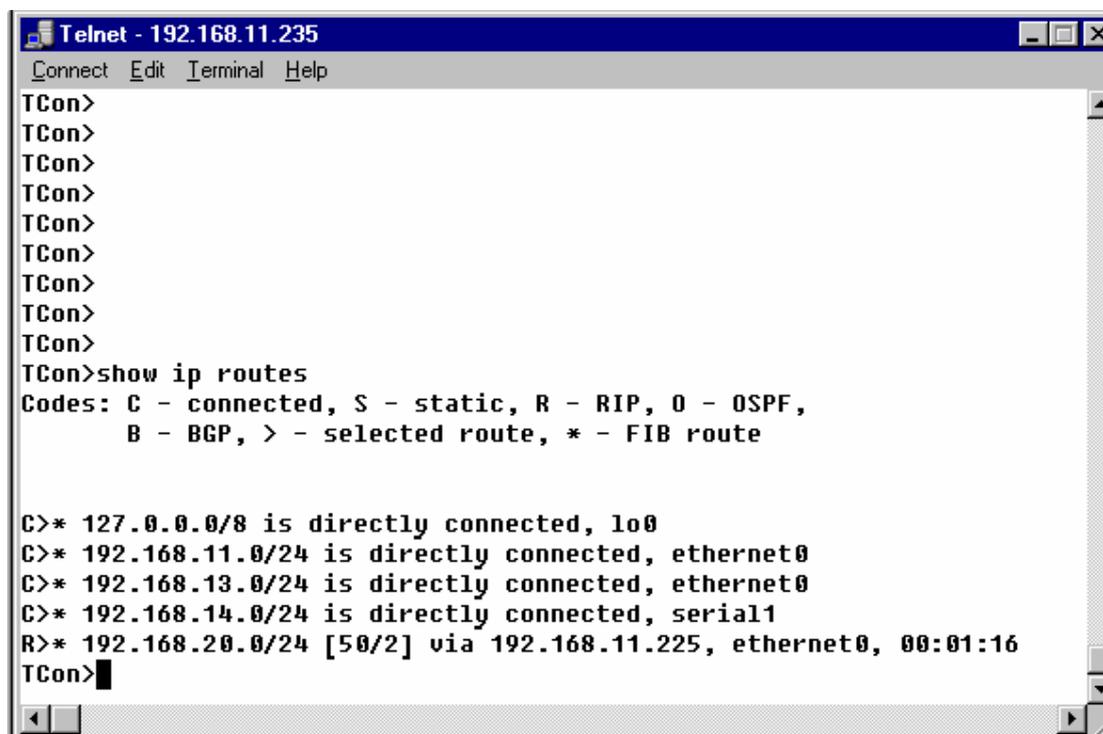
Рис. 2. Список маршрутов, распространяемых службой RIP.

Описание полей:

- **Network** – группа адресов, которую определяет маршрут;
- **Next Hop** – адрес шлюза;
- **Metric** – метрика маршрута;
- **From** – адрес маршрутизатора, приславшего информацию о маршруте;
- **Time** – время с последнего обновления записи таблицы.

1.7 Вывод таблицы маршрутизации

Команда `show ip routes` в корневом каталоге дерева команд коммуникационного процессора выводит содержимое базы данных маршрутизации, включая и неактивные маршруты (Рис. 3).



```
Telnet - 192.168.11.235
Connect Edit Terminal Help
TCon>
TCon>show ip routes
Codes: C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 192.168.11.0/24 is directly connected, ethernet0
C>* 192.168.13.0/24 is directly connected, ethernet0
C>* 192.168.14.0/24 is directly connected, serial1
R>* 192.168.20.0/24 [50/2] via 192.168.11.225, ethernet0, 00:01:16
TCon>
```

Рис. 3. Вывод таблицы маршрутизации.

Каждая запись таблицы содержит описание маршрута, группу адресов, которую он описывает, дистанцию и метрику (если маршрут не автоматический), адрес шлюза (если маршрут не автоматический), интерфейс, с которого будут отправляться пакеты, и время с последнего обновления (только для динамических маршрутов).

Описание маршрута – информация в формате {тип [флаг «активный»] [флаг «первый в базе»]}. Тип описывает происхождение маршрута. Имеются следующие типы:

- S – *static*, или *статические маршруты*;
- C – *connected*, или *автоматические маршруты, появляющиеся при присвоении адреса интерфейсу*;
- O – *ospf*, или *маршруты, полученные службой OSPF*;
- B – *bgp*, или *маршруты, полученные службой BGP*;
- R – *rip*, или *маршруты, полученные службой RIP*.

Флаг «активный» (выводится символом `>`) означает, что данный маршрут базы является действующим. При наличии нескольких одинаковых маршрутов активным становится маршрут с наименьшей административной дистанцией.

Флаг «первый в базе» означает, что маршрут имеет наименьшую административную дистанцию.

Группа адресов, которую описывает маршрут, выводится в формате адрес/длина маски. Дистанция и метрика определяются способом получения маршрута.

Команда `show ip routes kernel` в корневом каталоге дерева команд коммуникационного процессора выводит содержимое таблицы маршрутизации ядра маршрутизатора, включая ARP маршруты (Рис. 4).

```

Telnet - 192.168.11.235
Connect Edit Terminal Help
TCon>
TCon>
TCon>
TCon>sho
TCon>show ip rou
TCon>show ip routes k
TCon>show ip routes kernel
Destination      Gateway/Mask/Hw  Flags    Refs    Use  Expire  Interface
default          192.168.11.238  UGD      0       0    0      ethernet0
10.0.0.0         192.168.11.34   UGD      0       0    0      ethernet0
10.0.0.0         192.168.11.29   UGD      0       0    0      ethernet0
127.0.0.1        127.0.0.1       UH       2       77   0       lo0
192.168.11.0     255.255.255.0   U        10      0    878     ethernet0
192.168.11.29    00:00:00:00:00:00 UHL      1       0    878     ethernet0
192.168.11.37    00:C0:DF:05:9A:54 UHL      1      92  84254   ethernet0
192.168.11.238   00:01:AF:02:43:3D UHL      0       3   84138   ethernet0
192.168.11.255   FF:FF:FF:FF:FF:FF UHL      0      94    0       ethernet0
192.168.13.0     255.255.255.0   U        1       0    18      ethernet0
192.168.13.255   FF:FF:FF:FF:FF:FF UHL      0      94    0       ethernet0
192.168.14.0     255.255.255.0   U        1       0    0       serial1
192.168.14.255   192.168.14.1    UH       1      95    0       serial1
TCon>

```

Рис. 4. Вывод таблицы маршрутизации ядра маршрутизатора.

Описание полей:

- **Destination** – адрес назначения;
- **Gateway/Mask/Hw** – адрес шлюза/маска/канальный адрес. Содержимое поля зависит от типа отображаемого маршрута;
- **Flags** – флаги маршрута. **U** – маршрут активен; **H** – адресом назначения маршрута является адрес узла; **G** – маршрут на шлюз; **D** – маршрут создан в результате отработки процедуры «ICMP redirect»; **L** – ARP маршрут, **S** – статический маршрут;
- **Refs** – количество обращений к элементу таблицы;
- **Use** – количество пакетов, переданных с использованием данного элемента таблицы;
- **Expire** – время жизни элемента таблицы, определяется как момент времени, после которого запись должна быть обновлена; само время – число секунд с момента старта системы;
- **Interface** – сетевой интерфейс, на который заведен маршрут.

1.8 Статистика интерфейса Serial

```
Cons>show interface serial 0
```

Описание полей:

- **Interface serial0** – имя интерфейса;
- **Index 3, MTU = 1500, Metric 0** – индекс, максимальный размер передаваемого блока, метрика интерфейса;
- **Status (admin/proto/port) = (up/down/up)** – состояние интерфейса: административное, инкапсуляция, порт; up – рабочее состояние, down – нерабочее состояние;
- **BROADCAST RUNNING NOARP** – установленные флаги;
- **Input 0 packets, 0 bytes, 0 errors** – кол-во входящих пакетов, байт, ошибок;
- **Output 23 packets, 460 bytes, 0 errors** – кол-во исходящих пакетов, байт, ошибок;
- **IP address 192.168.11.242/24 broadcast 192.168.11.255** – сетевой адрес / маска подсети, широковещательный адрес интерфейса;
- **Input queue is fifo: 0 drops, 0 "no proto"** – входящая очередь, тип FIFO, кол-во отброшенных пакетов, кол-во пакетов с неопределенным протоколом.

Исходящая очередь FIFO

- **Output queue is fifo: 0/50/0/0 (size/max/drops)** – исходящая очередь, тип FIFO, текущий размер/максимальный размер/отброшено пакетов.

Исходящая очередь FIFOQ

- **Output queue is fifoq: 0/50/0/0 (size/max/pkts/drops)** – исходящая очередь, тип FIFOQ, текущий размер/максимальный размер/передано пакетов/отброшено пакетов.

Исходящая очередь RED

- **Output queue is red: 0/40/0/0 (size/max/pkts/drops)** – исходящая очередь, тип RED текущий размер/максимальный размер/передано пакетов/отброшено пакетов.

Исходящая очередь RIO

- **Output queue is rio: high 0/30/0/0 medium 0/0/0/30 low 0/0/0/0 (size/max/pkts/drops)** – исходящая очередь, тип RIO, три очереди RED с разным приоритетом, текущий размер/максимальный размер/передано пакетов/отброшено пакетов.

Исходящая очередь WFQ

- **Output queue is wfq: 32 queues size of 24000 bytes** – исходящая очередь, тип WFQ, количество очередей и их размер.

Исходящая очередь PRIQ

- **Output queue is priq: 2 classes** – исходящая очередь, тип PRIQ, количество классов.

Исходящая очередь CBQ

- **Output queue is cbq: 3 classes 100000bps 0/0** – исходящая очередь, тип CBQ, количество классов, общая полоса пропускания.

Исходящая очередь HFSC

- **Output queue is hfsc: 3 classes 99992bps 0/0 (pkts/drops)** – исходящая очередь, тип HFSC, количество классов, общая полоса пропускания (передано пакетов/отброшено пакетов).

Инкапсуляция HDLC

- **Encapsulation protocol is HDLC** – тип инкапсуляции HDLC;
- **Keepalive time is 10 sec** – время между посылками контрольных пакетов;
- **State is not reliable, last keepalive 0:5:4** - состояние соединения и время последнего приема контрольного пакета.

Инкапсуляция PPP

- **Encapsulation protocol is PPP** – тип инкапсуляции PPP;
- **PPP protocol phase** – фаза состояния протокола PPP;
- **MRU is 1500, MTU is 1500, multilink not allowed, reorders allowed** – максимальные размеры принимаемого и передаваемого пакетов, режим работы multilink;
- **0 ipackets 0 opackets 0 bad packets** – количество входящих\исходящих и плохих пакетов;
- **0 iocets 0 oocets 0 bad octets** – количество принятых\переданных и плохих октетов;
- **CP's queue (max\drops\len)** – параметры очереди команд PPP драйвера;
- **Last PPP NP frame received\sent:** - время (в секундах) со времени посылки\приема последнего сетевого пакета (внутри PPP фрейма);
- **0 protocols UP, 0 in OPEN state, conn. lifetime is 0, idle is 0** – количество протоколов, согласованных и использующихся в текущем соединении, время жизни соединения до разрыва (в случае сервера), время до разрыва соединения в случае простоя соединения.

Инкапсуляция Frame Relay

- **Encapsulation protocol is Frame Relay** – тип инкапсуляции Frame Relay;
- **Port type is QUICC Serial, mode Generic HDLC** – тип и режим работы порта;
- **Receive errors: 0 CRC, 0 overrun, 0 carrier lost** – ошибки приемника: несовпадение контрольной суммы, переполнение FIFO приёмника, потерян сигнал несущей;
- **0 frame violation, 0 non octet, 0 abort sequence** – размер фрейма больше допустимого значения, кол-во принятых битов не кратно 8;
- **0 dp11 error** – ошибка DPLL (цифровой цикл с фазовой синхронизацией);

- **Send errors:** 0 underruns, 0 CTS lost – ошибки передатчика: недогрузка FIFO передатчика, потерян сигнал CTS во время передачи;
- **DCD = up, DSR = up, DTR = up, RTS = up, CTS = up** – состояние сигналов в линии, up – высокий уровень, down – низкий уровень.

1.9 Статистика интерфейса Ethernet

Cons>show interface ethernet 0

Описание полей:

- **Interface ethernet0** – имя интерфейса;
- **Index 2, MTU = 1500, Metric 0** – индекс, максимальный размер передаваемого блока, метрика интерфейса;
- **Status (admin/proto/port) = (up/up/up)** – состояние интерфейса: административное, инкапсуляция, порт; up – рабочее состояние, down – нерабочее состояние;
- **UP BROADCAST RUNNING** – установленные флаги;
- **Input 0 packets, 0 bytes, 0 errors** – кол-во входящих пакетов, байт, ошибок;
- **Output 23 packets, 460 bytes, 0 errors** – кол-во исходящих пакетов, байт, ошибок;
- **IP address 192.168.11.242/24 broadcast 192.168.11.255** – сетевой адрес / маска подсети, широковещательный адрес интерфейса;
- **Input queue is fifo: 0 drops, 0 "no proto"** – входящая очередь, тип FIFO, кол-во отброшенных пакетов, кол-во пакетов с неопределенным протоколом;

Исходящая очередь FIFO

- **Output queue is fifo: 0/50/0 (size/max/drops)** – исходящая очередь, тип FIFO, текущий размер/максимальный размер/отброшено пакетов;

Исходящая очередь FIFOQ

- **Output queue is fifoq: 0/50/0/0 (size/max/pkts/drops)** – исходящая очередь, тип FIFOQ, текущий размер/максимальный размер/передано пакетов/отброшено пакетов;

Исходящая очередь RED

- **Output queue is red: 0/40/0/0 (size/max/pkts/drops)** – исходящая очередь, тип RED текущий размер/максимальный размер/передано пакетов/отброшено пакетов;

Исходящая очередь RIO

- **Output queue is rio: high 0/30/0/0 medium 0/0/0/30 low 0/0/0/0 (size/max/pkts/drops)** – исходящая очередь, тип RIO, три очереди RED с разным приоритетом, текущий размер/максимальный размер/передано пакетов/отброшено пакетов;

Исходящая очередь WFQ

- **Output queue is wfq: 32 queues size of 24000 bytes** – исходящая очередь, тип WFQ, количество очередей и их размер;

Исходящая очередь PRIQ

- **Output queue is priq: 2 classes** – исходящая очередь, тип PRIQ, количество классов;

Исходящая очередь CBQ

- **Output queue is cbq: 3 classes 100000bps 0/0** – исходящая очередь, тип CBQ, количество классов, общая полоса пропускания;

Исходящая очередь HFSC

- **Output queue is hfsc: 3 classes 99992bps 0/0 (pkts/drops)** – исходящая очередь, тип HFSC, количество классов, общая полоса пропускания (передано пакетов/отброшено пакетов);
- **Encapsulation protocol is ARPA** – тип инкапсуляции ARPA;
- **address is 0001.0011.2233** – физический MAC адрес;
- **ARP cache TTL 1200, down-dont-send time 20,** – время жизни ARP записи в кэше, время, в течение которого хост считается выключенным и запросы к нему не отсылаются;
- **prune interval 300, retries number 5** – интервал обхода таблицы ARP для удаления устаревших записей, кол-во попыток определения адреса;
- **Port type is QUICC Fast Ethernet, mode Generic Ethernet** – тип и режим работы порта;
- **Receive errors: 0 short, 0 giant, 0 non-octet** – ошибки приемника: кол-во коротких пакетов, кол-во длинных пакетов, кол-во принятых битов не кратно 8;
- **0 CRC, 0 overrun, 0 collision** – несовпадение контрольной суммы, переполнение FIFO приёмника, коллизия в линии, отброшенные при приеме пакеты;
- **Send errors: 0 underrun, 0 late collisions, 0 deferred** – ошибки передатчика: недогрузка FIFO передатчика, после передачи 64 бит пакета, коллизия до окончания передачи пакета;
- **lost carrier, 0 heartbeat** – потеря несущей, качество сигнала;
- **0 retries, 0 retransmit limit** – кол-во повторных передач, кол-во повторных передач превысило предел.

1.10 Статистика очереди, выводимая командой `"show queue {ethernet / serial}"`

Команда `show queue {ethernet | serial}` выводит информацию о типе выходной очереди, её статистике и параметрах. В зависимости от установленного типа очереди на интерфейсе статистика будет различаться. При использовании иерархических или других типов очередей, состоящих из нескольких классов (очередей), информация выводится для каждого класса (очереди). Примеры вывода статистики возможных очередей приведены ниже.

1.10.1 Вывод статистики при типе выходной очереди FIFOQ

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is fifoq:
```

```
qlen:0 qmax:40 period:529
```

```
xmit:3388 pkts (1498536 bytes) drop:141 pkts (211778 bytes)
```

```
throughput: 3278016bps, 271pps
```

Описание полей:

- `Output queue is fifo` – означает, что в данный момент типом исходящей очереди интерфейса является тип FIFO;
- `qlen` – текущая длина очереди;
- `qmax` – максимальная длина очереди;
- `period` – сколько раз очередь становилась пустой;
- `xmit` – передано пакетов (байт);
- `drop` – отброшено пакетов (байт);
- `throughput` – пропускная способность (бит/сек, число пакетов/сек).

1.10.2 Вывод статистики при типе выходной очереди RED

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is red:
```

```
weight:64 inv_pmax:20 qthresh:(10,30)
```

```
qlen:14 (avg: 17), qmax:50
```

```
xmit:517 pkts, drop:149 pkts (forced: 142, early: 7)
```

```
throughput: 1935360bps
```

Описание полей:

- **Output queue is red** – означает, что в данный момент типом исходящей очереди интерфейса является тип RED;
- **weight** – обратное значение весового коэффициента экспоненциально взвешенной скользящей средней (EWMA);
- **inv_pmax** – обратное значение максимальной вероятности отбрасывания пакетов;
- **qthresh** – минимальная и максимальная пороговые величины длины очереди;
- **qlen** – текущая длина очереди;
- **avg** – средняя длина очереди;
- **qmax** – максимальная длина очереди;
- **xmit** – количество переданных пакетов;
- **drop** – количество отброшенных пакетов;
- **forced** – количество принудительно отброшенных пакетов;
- **early** – количество непринудительно отброшенных пакетов (раннее отбрасывание);
- **marked** – количество промаркированных пакетов (установлен бит предупреждения о перегрузке);
- **throughput** - пропускная способность (бит/сек).

1.10.3 Вывод статистики при типе выходной очереди RIO

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is rio:
```

```
Low DP
```

```
weight:64 inv_pmax:30 qthresh:(20,15)
```

```
qlen:16 (avg: 14), qmax:30
```

```
xmit:1091 pkts, drop:219 pkts (forced: 219, early: 0)
```

```
throughput: 2909264bps
```

```
Medium DP
```

```
weight:64 inv_pmax:20 qthresh:(10,15)
```

```
qlen:16 (avg: 14), qmax:30
```

```
xmit:0 pkts, drop:0 pkts (forced: 0, early: 0)
```

```
throughput: 0bps
```

```
High DP
```

```
weight:64 inv_pmax:10 qthresh:(5,7)
```

```
qlen:15 (avg: 14), qmax:30
```

```
xmit:0 pkts, drop:0 pkts (forced: 0, early: 0)
```

```
throughput: 0bps
```

Описание полей:

- `Output queue is rio` – означает, что в данный момент типом исходящей очереди интерфейса является тип RIO;
- `Low DP` – класс для трафика с низким приоритетом;
- `Medium DP` – класс для трафика со средним приоритетом;
- `High DP` – класс для трафика с высоким приоритетом;
- `weight` – обратное значение весового коэффициента экспоненциально взвешенной скользящей средней (EWMA);
- `inv_pmax` – обратное значение максимальной вероятности отбрасывания пакетов;
- `qthresh` – минимальная и максимальная пороговые величины длины очереди;
- `qlen` – текущая длина очереди;

- **avg** – средняя длина очереди;
- **qmax** – максимальная длина очереди;
- **xmit** – количество переданных пакетов;
- **drop** – количество отброшенных пакетов;
- **forced** – количество принудительно отброшенных пакетов;
- **early** – количество непринудительно отброшенных пакетов (раннее отбрасывание);
- **marked** – количество промаркированных пакетов (установлен бит предупреждения о перегрузке);
- **throughput** - пропускная способность (бит/сек).

1.10.4 Вывод статистики при типе выходной очереди WFQ

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is wfq: 32 queues size of 8000 bytes, classifier by all fields
```

[QID]	WEIGHT	QSIZE (KB)	SENT (pkts)	(KB)	DROP (pkts)	(KB)	bps
[12]	100	0	9627	9796	1897	1930	0
[2]	100	0	22	2	0	0	0
[0]	100	0	5	0	0	0	0
[1]	100	0	1	0	0	0	0
[3]	100	0	1	0	0	0	0
[4]	100	0	1	0	0	0	0
[5]	100	0	1	0	0	0	0
[6]	100	0	1	0	0	0	0
[7]	100	0	1	0	0	0	0
[14]	100	0	1	0	0	0	0

Описание полей:

- **Output queue is wfq** – означает, что в данный момент типом исходящей очереди интерфейса является тип WFQ;
- **32 queues size of 8000 bytes** – кол-во используемых очередей и их размер в байтах;
- **classifier by all fields** – классификация по адресам и номерам портов источника и назначения и протоколу;
- **classifier by destination address** – классификация по адресу назначения;
- **classifier by source port** – классификация по номеру порта источника;
- **QID** – идентификатор очереди;
- **WEIGHT** – вес очереди;
- **QSIZE** – размер очереди (кбайт);
- **SENT** – количество отправленных пакетов (кбайт);
- **DROP** – количество отброшенных пакетов (кбайт);
- **bps** – количество отправленных бит за последнюю секунду.

1.10.5 Вывод статистики при типе выходной очереди PRIQ

Пример вывода статистики:

```
Cons>show queue ethernet 0

Output queue is priq: 4 classes

Class: udp
measured: 601920bps qlen: 0 period:339
packets:2031 (3057336 bytes) drops:0
RED q_avg:5 xmit:2031 (forced:0 early:0 marked:0)

Class: tcp
measured: 2117504bps qlen: 9 period:12663
packets:86038 (24131596 bytes) drops:3238
      RED q_avg:6 xmit:86047 (forced:2564 early:674 marked:0)

Class: control
measured: 0bps qlen: 0 period:0
packets:0 (0 bytes) drops:0

Class: default
measured: 0bps qlen: 0 period:4
packets:4 (240 bytes) drops:0
RED q_avg:0 xmit:4 (forced:0 early:0 marked:0)
```

Описание полей:

- `Output queue is priq` – означает, что в данный момент типом исходящей очереди интерфейса является тип PRIQ;
- `xx classes` – количество классов;
- `class` – имя класса;
- `measured` – количество бит, переданных за последнюю секунду;
- `qlen` – текущий размер очереди класса;
- `period` – число раз, когда очередь становилась пустой;
- `packets` – количество переданных пакетов;
- `drops` – количество отброшенных пакетов;

- **RED** – для данного класса используется очередь RED вместо FIFO;
- **q_avg** – средний размер очереди;
- **xmit** – количество переданных байт;
- **forced** – количество принудительно отброшенных пакетов;
- **early** – количество непринудительно отброшенных пакетов (раннее отбрасывание);
- **marked** – количество промаркированных пакетов (установлен бит предупреждения о перегрузке).

1.10.6 Вывод статистики при типе выходной очереди CBQ

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is cbq: 5 classes
```

```
Root Class: root
```

```
    priority: 0 depth: 1 offtime: 0 [us] wrr_allot: 1500 bytes
```

```
    bandwidth: 10000000bps,  measured: 704bps, 1pps
```

```
    pkts: 333,      bytes: 30166
```

```
    drops: 0,      bytes: 0
```

```
    overs: 0,      overactions: 0
```

```
    borrows: 0,    delays: 0
```

```
    qlen: 0,       (qmax: 60)
```

```
    avgidle: 37 [us],      (maxidle: 37 minidle: -37 [us])
```

```
Default Class: default
```

```
    priority: 3 depth: 0 offtime: 45675 [us] wrr_allot: 1523 bytes
```

```
    bandwidth: 256000bps,  measured: 0bps, 0pps
```

```
    pkts: 194,      bytes: 18866
```

```
    drops: 0,      bytes: 0
```

```
    overs: 0,      overactions: 0
```

```
    borrows: 0,    delays: 0
```

```
    qlen: 0,       (qmax: 30)
```

```
    avgidle: 2994 [us],    (maxidle: 2994 minidle: -1464 [us])
```

```
Ctl Class: control
```

priority: 1 depth: 0 offtime: 118800 [us] wrr_allot: 1562 bytes
bandwith: 100000bps, measured: 704bps, 1pps
pkts: 139, bytes: 11300
drops: 0, bytes: 0
overs: 0, overactions: 0
borrows: 0, delays: 0
qlen: 0, (qmax: 30)
avgidle: 7788 [us], (maxidle: 7788 minidle: -3750 [us])

Class 0: data

priority: 7 depth: 0 offtime: 22237 [us] wrr_allot: 1523 bytes
bandwith: 512000bps, measured: 0bps, 0pps
pkts: 0, bytes: 0
drops: 0, bytes: 0
overs: 0, overactions: 0
borrows: 0, delays: 0
RED q_avg:0 xmit:0 (forced:0 early:0 marked:0)
qlen: 0, (qmax: 60)
avgidle: 1457 [us], (maxidle: 1457 minidle: -732 [us])

Class 1: voice

priority: 2 depth: 0 offtime: 186300 [us] wrr_allot: 1600 bytes
bandwith: 64000bps, measured: 0bps, 0pps
pkts: 0, bytes: 0
drops: 0, bytes: 0
overs: 0, overactions: 0
borrows: 0, delays: 0
qlen: 0, (qmax: 30)
avgidle: 12213 [us], (maxidle: 12213 minidle: -5859 [us])

Описание полей:

- **Output queue is cbq** – означает, что в данный момент типом исходящей очереди интерфейса является тип CBQ;
- **xx classes** – количество классов;
- **Root Class** – корневой класс;
- **Default Class** – класс для трафика по умолчанию;
- **Ctl Class** – класс для управляющего трафика;
- **class** – имя класса;
- **priority** – приоритет обслуживания класса;
- **depth** – уровень иерархии класса;
- **offtime** – период приостановки между передачей двух пакетов;
- **wrr_allot** – доля ресурсов во взвешенном алгоритме кругового обслуживания;
- **nsPerByte** – количество наносекунд для передачи одного байта (пропускная способность класса);
- **measured** – количество бит и пакетов, переданных за последнюю секунду;
- **pkts** – количество переданных пакетов;
- **bytes** – количество переданных байт;
- **overs** – число раз превышения классом назначенной полосы пропускания;
- **overactions** – сколько раз производилось задерживание или отбрасывание пакета при превышении классом полосы пропускания;
- **borrow** – сколько раз класс занимал полосу пропускания у родительского класса;
- **delays** – сколько раз производилось задерживание класса;
- **drops** – количество отброшенных пакетов;
- **drop_bytes** – количество отброшенных байт;
- **qlen** – текущий размер очереди класса;
- **qmax** – максимальный размер очереди класса;
- **avgidle** – среднее время ожидания класса;
- **maxidle** – максимальное время ожидания класса;
- **minidle** – минимальное время ожидания класса;
- **RED** – для данного класса используется очередь RED вместо FIFO;

- **q_avg** – средний размер очереди;
- **xmit** – количество переданных байт;
- **forced** – количество принудительно отброшенных пакетов;
- **early** – количество непринудительно отброшенных пакетов (раннее отбрасывание);
- **marked** – количество промаркированных пакетов (установлен бит предупреждения о перегрузке).

1.10.7 Вывод статистики при типе выходной очереди HFSC

Пример вывода статистики:

```
Cons>show queue ethernet 0
```

```
Output queue is hfsc: 4 classes, 4 packets in the tree
```

```
[ 0 root][rt 9999990 0ms 9999990][ls 9999990 0ms 9999990]
```

```
measured: 0bps [rt:0 ls:0] qlen: 0 period:0
```

```
packets:0 (0 bytes) drops:0
```

```
cumul:0 total:0
```

```
[ 1 default][rt 0 0ms 999999][ls 0 0ms 0]
```

```
measured: 560688bps [rt:560688 ls:0] qlen: 0 period:62
```

```
packets:1520 (2264450 bytes) drops:81
```

```
cumul:2264450 total:2264450
```

```
RED q_avg:10 xmit:1520 (forced:4 early:77 marked:0)
```

```
[ 2 control][rt 0 0ms 99992][ls 0 0ms 0]
```

```
measured: 0bps [rt:0 ls:0] qlen: 0 period:0
```

```
packets:0 (0 bytes) drops:0
```

```
cumul:0 total:0
```

```
[ 3 video][rt 0 0ms 0][ls 0 0ms 4999995]
```

```
measured: 3314304bps [rt:0 ls:3314304] qlen: 4 period:3028
```

```
packets:5159 (7800408 bytes) drops:202
```

```
cumul:0 total:7800408
```

Описание полей

- **Output queue is hfsc** – означает, что в данный момент типом исходящей очереди интерфейса является HFSC;
- **xx classes** – количество классов;
- **xx packets in the tree** – количество пакетов в дереве классов;
- **rt** – кривая обслуживания, определяющая гарантированную полосу пропускания;
- **ls** – кривая обслуживания, определяющая негарантированную полосу пропускания;
- **xx yy ms zz** – параметры кривой обслуживания: xx – полоса пропускания первого сегмента, yy – время (мсек) начала второго сегмента, zz - полоса пропускания второго сегмента;
- **measured** – количество бит, переданных за последнюю секунду;
- **qlen** – длина очереди класса;
- **period** – сколько раз очередь становилась пустой;
- **packets** – количество переданных пакетов (байт) класса;
- **drops** – количество отброшенных пакетов класса;
- **cumul** – количество переданных байт с гарантированной полосой пропускания;
- **total** – количество переданных байт;
- **RED** – для данного класса используется очередь RED вместо FIFO;
- **q_avg** – средний размер очереди;
- **xmit** – количество переданных байт;
- **forced** – количество принудительно отброшенных пакетов;
- **early** – количество не принудительно отброшенных пакетов (раннее отбрасывание);
- **marked** – количество промаркированных пакетов (установлен бит предупреждения о перегрузке).

1.11 Команды просмотра статистики и состояния объектов *frame relay*.

1.11.1 Вывод состояния LMI интерфейса

```
show frame-relay lmi interface интерфейс
```

Команда осуществляет вывод состояния LMI указанного интерфейса.

Пример вывода параметров LMI представлен на Рис. 5

```
Cons#show frame-relay lmi interface serial 2
Interface type is "UNI network", LMI type "ITU-T Annex A".
LMI DLCI 0, DCE status UP, Summary status is UP.
Timers and counters:
  DCE: T392 = 15 sec, N392 = 3, N393 = 4
LMI packets statistics:
  "STATUS" pkts rcvd           0 "STATUS" pkts sent           5510
  "STATUS ENQUIRY" pkts rcvd  5510 "STATUS ENQUIRY" pkts sent  0
  bad pkts rcvd               0 sync lost pkts rcvd       5301
```

Рис. 5 Пример вывода параметров LMI

Выведенная информация показывает тип интерфейса, тип LMI, номер DLCI, который используется для обмена сообщениями LMI, состояния DCE (тип интерфейса DCE и NNI), DTE (тип интерфейса DTE и NNI), и суммарный статус. Кроме того, показаны счетчики LMI и статистика.

Описание полей:

- | | |
|------------------------------|--|
| "STATUS" pkts rcvd - | количество принятых пакетов типа "STATUS" |
| "STATUS" pkts sent - | количество отправленных пакетов типа "STATUS" |
| "STATUS ENQUIRY" pkts rcvd - | количество принятых пакетов типа "STATUS ENQUIRY" |
| "STATUS ENQUIRY" pkts sent - | количество отправленных пакетов типа "STATUS ENQUIRY" |
| bad pkts rcvd - | количество принятых пакетов неправильной структуры |
| sync lost pkts rcvd - | количество потерь синхронизации при приеме пакетов LMI |

1.11.2 Вывод состояния ARP таблицы интерфейса

```
show frame-relay map interface интерфейс
```

Команда осуществляет вывод состояния ARP таблицы указанного интерфейса. Кроме того, выводится список DLCI мультимедийной инкапсуляции данного интерфейса и их параметры.

Пример вывода параметров ARP таблицы представлен на **Ошибка! Источник ссылки не найден.**

```
Cons#show frame-relay map interface serial 2.  
Connection type is point-to-point  
Total DLCI on interface 1  
100: IETF, Inv. ARP on ( IP ),map entries 0.  
Total map entries 2  
1: DLCI = 21, IP (10.0.0.2), < STATIC >.  
2: DLCI = 20, IP (10.0.0.1), < STATIC >.
```

Приведенная информация означает, что интерфейс serial 2 имеет один DLCI. Номер DLCI равен 100. На нем разрешена процедура инверсного ARP. Тип инкапсуляции данных сетевого уровня во frame relay IETF (RFC1490). Количество записей таблицы ARP, ссылающихся на данный DLCI, равно 0. Всего количество записей таблицы ARP интерфейса serial 2 равно двум. Это соответствие IP адреса 10.0.0.2 DLCI 21 и IP адреса 10.0.0.1 DLCI 20. Обе записи статические. Об этом свидетельствует флаг «STATIC». Возможны также следующие флаги: «DYNAMIC» - запись получена в результате процедуры инверсного ARP, «DLCI» - DLCI с указанным в записи номером существует.

1.11.3 Вывод состояния PVC

```
show frame-relay pvc interface интерфейс
```

```
show frame-relay pvc номер-DLCI
```

```
show frame-relay pvc
```

Команда осуществляет вывод информации о DLCI или PVC (Permanent Virtual Channel) указанного интерфейса или всех интерфейсов. Выводимая информация включает в себя параметры настройки, а также статистику по пакетам. Для основных интерфейсов показывается его собственные DLCI мультипротокольной инкапсуляции, а также все DLCI коммутации. Для субинтерфейсов показывается их DLCI мультипротокольной инкапсуляции.

Пример вывода параметров PVC представлен на Рис. 6

```
Cons#show frame-relay pvc

INTERFACE serial2:
DLCI 100: DLCI usage "Frame Relay Multiprotocol Encapsulation"
STATUS: local = ACTIVE, remote = ACTIVE;
Quality of Service class test1
QoS parameters: priority normal, {Bc,Be,Tc} = {100000,16000000,10}
Output queue type is FIFO, (size/pkts/drops) = 8192/0/0
"FRF.12 end to end fragmentation" enabled, fragment size 50
Q.922 packets statistics:
input packets           0 output packets       0
input bytes             0 output bytes         0
input FECN packets     0 output FECN packets 0
input BECN packets     0 output BECN packets 0
input DE packets       0 output DE packets   0
input drops             0 output drops         0
FRF.12 UNI/NNI fragmentation statistics (fragment size 50):
input pkts reassembled  0 output pkts fragmented 0
input fragments        0 output fragments       0
input too long         0 input sync lost       0
FRF.12 end to end fragmentation statistics (fragment size 50):
input pkts reassembled  0 output pkts fragmented 0
input fragments        0 output fragments       0
input too long         0 input sync lost       0
Cons#
```

Рис. 6 Пример вывода параметров PVC

Из представленной выше информации видно, что интерфейс serial 2 имеет один DLCI мультипротокольной инкапсуляции. Перечисленные далее параметры означают следующее:

STATUS. Состояние DLCI. Общий статус DLCI является результатом двух статусов – локального и удаленного. Локальный статус отражает состояние объектов, которые связаны с данным DLCI. Для DLCI мультипротокольной инкапсуляции это административный статус логического интерфейса, ассоциированного с данным DLCI. Для DLCI коммутации это наличие активного удаленного статуса DLCI на интерфейсе назначения. Локальный статус передается удаленной стороне в сообщениях LMI. Удаленный статус формируется на основании сообщений LMI принимаемых от удаленной стороны соединения. В режимах `po-lmi` и `dce` он принудительно выставляется активным.

Quality of Service class. Данная строка выводит информацию о классе качества обслуживания, ассоциированном с данным DLCI. В данном случае класс определен и называется `test1`.

QoS parameters. Параметры качества обслуживания трафика данного DLCI. Они включают в себя приоритет и набор `{Bc,Be,Tc}`. Параметры отображаются только в том случае, когда они заданы.

Output queue type. Информация об очереди передачи DLCI. Указываются тип, размер (size), количество пакетов (pkts) и количество отброшенных пакетов (drops).

Информация о фрагментации. Она включает в себя тип фрагментации и размер фрагмента.

Q922 packets statistics. Статистика пакетов канального уровня данного DLCI. Включает в себя следующие элементы:

<code>input packets</code> -	количество принятых пакетов;
<code>output packets</code> -	количество отправленных пакетов;
<code>input bytes</code> -	количество принятых байтов;
<code>output bytes</code> -	количество отправленных байтов;
<code>input FECN packets</code> -	количество принятых пакетов с флагом FECN;
<code>output FECN packets</code> -	количество отправленных пакетов с флагом FECN;
<code>input BECN packets</code> -	количество принятых пакетов с флагом BECN;
<code>output BECN packets</code> -	количество отправленных пакетов с флагом BECN;
<code>input DE packets</code> -	количество принятых пакетов с флагом DE;
<code>output DE packets</code> -	количество отправленных пакетов с флагом DE;
<code>input drops</code> -	количество пакетов, отброшенных при приеме;
<code>output drops</code> -	количество пакетов, отброшенных при передаче.

Статистика фрагментации. Выводится, если включена фрагментация на интерфейсе или DLCI. В данном случае включены обе фрагментации. Статистика фрагментации включает в себя следующие поля:

<code>input pkts reassembled</code> -	количество дефрагментированных пакетов при приеме;
<code>output pkts reassembled</code> -	количество дефрагментированных пакетов при передаче;
<code>input fragments</code> -	количество принятых фрагментов;
<code>output fragments</code> -	количество отправленных фрагментов;
<code>input too long</code> -	количество ошибок при приеме из-за превышения дефрагментируемым пакетом максимальной длины – 8192 байт;
<code>input sync lost</code> -	количество потерь синхронизации при дефрагментации.

1.11.4 Вывод таблицы коммутации пакетов frame relay

```
show frame-relay route
```

Пример вывода таблицы коммутации пакетов frame relay представлен на Рис. 7

```
Cons#show frame-relay route
Src Interface      Src DLCI  Dst Interface      Dst DLCI  Status
*serial2          100      serial1             200        INACTIVE
*serial2          101      serial1             300        INACTIVE
```

Рис. 7 Пример вывода коммутации пакетов frame relay

Столбик «Src Interface» содержит название интерфейса источника, «Src DLCI» - номер коммутируемого DLCI интерфейса источника, «Dst Interface» содержит название интерфейса назначения, «Dst DLCI» - номер коммутируемого DLCI интерфейса назначения. Звездочка слева означает, что объект существует. Столбик «Status» показывает текущее состояние записи. Состояние становится активным, если интерфейс источника, интерфейс назначения, указанные DLCI созданы, и DLCI назначения имеет активный удаленный статус.

ПЕРЕЧЕНЬ РИСУНКОВ

Рис. 1 Вывод информации о протоколах маршрутизации.....	14
Рис. 2. Список маршрутов, распространяемых службой RIP.....	17
Рис. 3. Вывод таблицы маршрутизации.....	18
Рис. 4. Вывод таблицы маршрутизации ядра маршрутизатора.....	19
Рис. 5 Пример вывода параметров LMI	36
Рис. 7 Пример вывода параметров PVC	38
Рис. 8 Пример вывода коммутации пакетов frame relay	40

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

clear icmp-stats	11	show ip protocols	14
clear ip-stats	5	show ip rip	17
clear tcp-stats	8	show ip routes	18
clear udp-stats	13	show ip routes kernel	19
show frame-relay lmi interface	35	show ip statistics	5, 8, 11
show frame-relay map interface	36	show queue {ethernet serial}	24
show frame-relay pvc	37	show queue ethernet 0	24, 25, 26, 28, 29, 30, 33
show frame-relay pvc interface	37	show tcp statistics	8
show frame-relay route	39	show udp statistics	13
show icmp statistics	11, 13		
show interface ethernet 0	22		
show interface serial 0	20		