Сервер доступа к каналам E1
PACS-E1

Конфигурация и управление
Версия 1.0
02.04.2010

Новосибирск 2010

Разработчик и производитель: ООО «Парабел»
630090, Новосибирск-90, а/я 126
http://www.parabel.ru
Email: info@parabel.ru
Тел/факс: +7-383-2138707

Содержание

1. Введение

В данном документе приводится описание программного обеспечения (ПО) сервера PACS-E1, рассматриваются особенности управления и конфигурации. Программное обеспечение представляет собой пакет GNU/Linux с возможностью автономного старта и загрузки с USB flash памяти.

ПО сервера поддерживает следующие протоколы и сервисы:

Протоколы канального уровня на интерфейсах E1

Cisco HDLC, Cisco HDLC bridge, Синхронный PPP, Frame Relay

Маршрутизация

BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, RIPng

Аутентификация пользователей

RADIUS

Безопасность

IP firewall, NAT

Статистика

IP accounting

Управление и конфигурация

Локальная консоль (SVGA, Kbd)

ssh, telnet, ftp, nfs

Диагностика

traceroute, dig, tcpdump, netcat

С помощью менеджера пакетов дополнительно может быть установлена любая утилита из стандартного репозитория Debian.

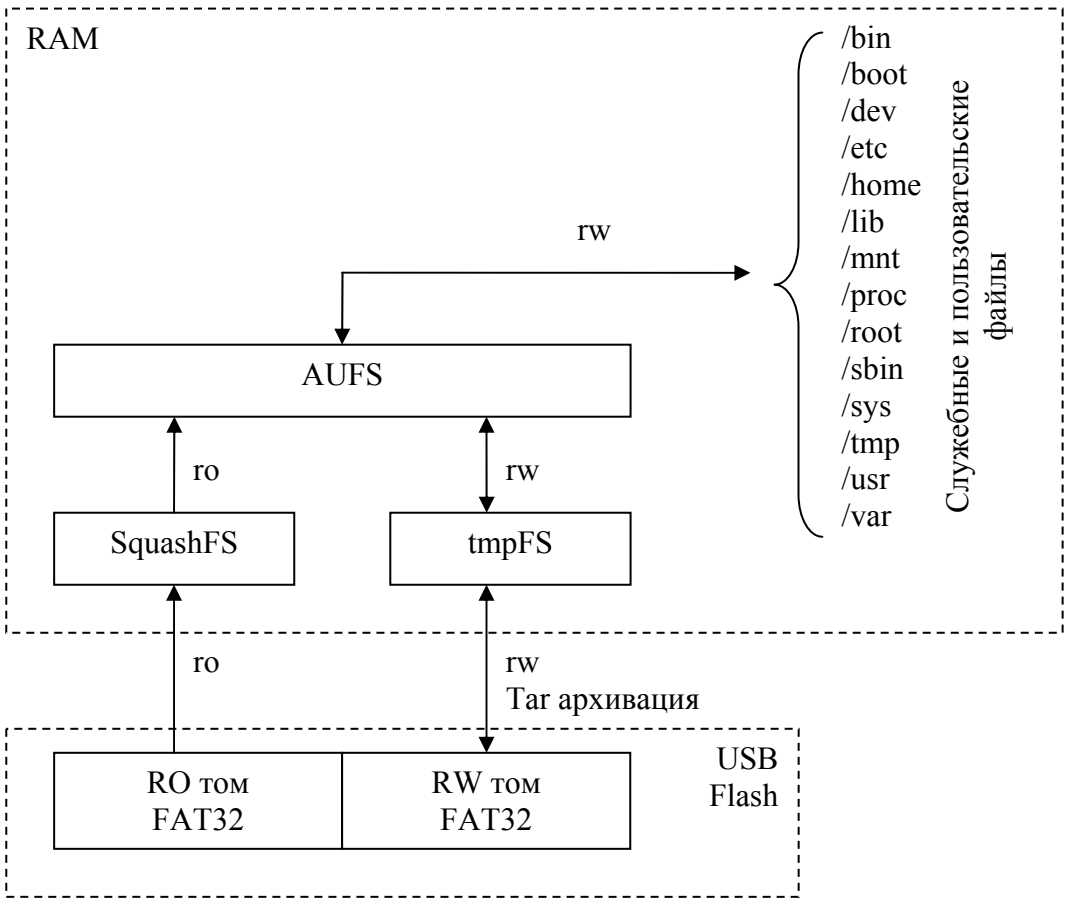
2. Лицензионное соглашение

Рассмотренное в данном документе программное обеспечение не является предметом продажи или каких-либо иных сделок с коммерческой направленностью, а поставляется производителем бесплатно. Стоимость ПО не заложена в стоимость оборудования. Производитель не налагает авторских прав на данное ПО и допускает его дополнение пользователями.

Пакет ПО собран из открытых источников в Интернете и поставляется «как есть». Производитель сервера PACS-E1 не несет ответственности за возможные ошибки или несоответствия документации, допущенные разработчиками открытого кода. Тем не менее, производитель налагает на себя обязанность по технической поддержке пользователей сервера PACS-E1 в порядке консультативной помощи. Производитель также гарантирует совместимость поставляемого ПО с аппаратурой сервера.

3. Особенности файловой системы

Структура файловой системы сервера PACS-E1 изображена на рисунке. После загрузки, в рабочем режиме ПО сервера работает **только** в оперативной памяти, без обращения к flash диску. Как видно из рисунка, рабочая файловая система состоит из двух частей – только для чтения (ro) и только для записи (rw).



Ro часть содержит ядро Linux, системные утилиты и заводскую конфигурацию. Rw часть содержит все изменения, внесенные пользователем во время работы. Файловые системы ro и rw объединяются в одно целое с помощью файловой системы AUFS, которая предоставляет для пользователя стандартную корневую файловую систему Linux, доступную для записи. Таким образом, отслеживание изменений и записей в файлы происходит незаметно для пользователя. Во время работы сервера содержимое ro и rw файловых систем можно посмотреть в директориях /.pb/ro/ и /.pb/rw/ соответственно.

Инициализация обеих частей файловой системы происходит из разных томов USB flash носителя. Так, для ro части используется том размером около 200 Мб с файловой

системой FAT32. Здесь же содержится MBR и загрузчик Grub. Для rw системы отводится весь оставшийся объем flash носителя, который также организован в формате FAT32. Содержимое rw системы записывается в этот том в виде TAR архивов. Запись содержимого rw системы инициируется только пользователем и происходит в трех случаях – при перезапуске системы (командой reboot), при останове системы (команда halt) и при записи конфигурации во flash (команда writeflash).

При старте системы происходит обратный процесс – образ rw системы распаковывается из tar архивов, расположенных в RW томе flash диска и записывается в оперативную память сервера.

Рассмотренное строение файловой системы обеспечивает удобство конфигурации сервера, а также обеспечивает высокую отказоустойчивость.

4. Конфигурация Linux в целом

Изменить конфигурацию сервера можно двумя способами – оперативно и перманентно, через конфигурационные файлы. Оперативные изменения конфигурации происходят с помощью консольных команд и наступают сразу, без перезагрузки сервера. Например, изменить IP адрес можно консольной командой **ifconfig**:

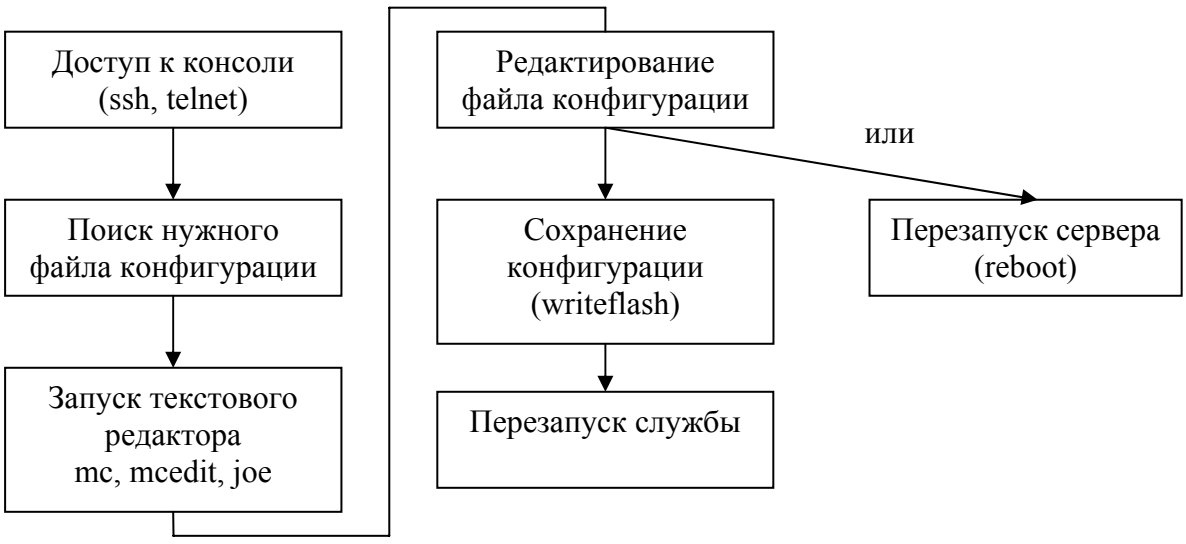
 # ifconfig eth0 192.168.1.1

Адрес **192.168.1.1** присвоится интерфейсу **eth0** сразу же. Оперативные изменения конфигурации будут действительны до первой перезагрузки системы. Их невозможно сохранить во flash память.

Для перманентного изменения конфигурации необходимо редактировать конфигурационные файлы. Конфигурационные файлы в большинстве случаев имеют текстовый формат и редактируются встроенными редакторами, входящими в состав ПО сервера. Например, изменение IP адреса может быть сделано путем редактирования файла **/etc/network/interfaces**. Сделать это можно с помощью файлового менеджера **mc** или редактора **joe**. После изменения конфигурационного файла, конфигурацию системы в целом нужно сохранить командой **writeflash**. Чтобы новые параметры стали актуальными, необходимо или перезапустить весь сервер или соответствующий сервис. В нашем случае достаточно перезапустить сетевую подсистему:

 # /etc/init.d/networking restart

Таким образом, при изменении конфигурации, действия пользователя укладываются в следующую диаграмму:



5. Базовые навыки

5.1. Как изменить IP адрес

При загрузке ОС, параметры для сетевых интерфейсов берутся из файла "/etc/network/interfaces".

- "auto lo eth0 eth1" указывает ОС, что надо сконфигурировать интерфейсы eth0, eth1.
- "iface eth0 inet dhcp" указывает, что параметры интерфейса eth0 надо получить по протоколу dhcp.
- Следующие строки описывают интерфейс eth1 с статическим адресом.
  "iface eth1 inet static
  address 192.168.1.1
  netmask 255.255.0.0
  name Private network"

При загрузке, ОС интерпретирует файл "/etc/network/interfaces" и вызывает команду "*ifconfig*" с указанными параметрами. Изменить адрес интерфейса без перезагрузки и сохранения можно командой "*ifconfig*".

Пример:

"*ifconfig* eth1 192.168.1.1 netmask 255.255.255.0" установит интерфейсу eth1 адрес 192.168.1.1 с маской сети 255.255.255.0 .

5.2. Как изменить имя системы

Имя системы хранится в файле /etc/hostname , и считывается операционной системой (ОС) единожды (при загрузке). В зависимости от настройки, сервисы (демоны) могут использовать это имя, или хранить свое имя.

5.3. Изменение системного времени и даты

Сервер имеет часы реального времени (RTC), работающие даже при выключенном питании. Настройка часов реального времени осуществляется командой "*hwclock*".

Запустившись, операционная система берет время RTC, добавляет к нему (если необходимо) смещение временной зоны, и в дальнейшем обслуживает время самостоятельно. Управление временем ОС занимается команда "*date*".

Однако часы сервера имеют конечную точность, и могут уходить. Чтобы это избежать, рекомендуется настроить сервер времени (openntpd). Сервер openntpd подключается к мировым серверам времени, и плавно подстраивает локальные часы (шаг подстройки примерно 0.1 сек за 60 секунд).
Настройка openntpd осуществляется в файле /etc/openntpd/ntpd.conf .

5.4. Как поменять пароль

Поменять пароль пользователя user1 можно командой "*passwd user1*".
Обязательно смените пароль пользователя root и 5.6. Как сохранить текущую конфигурацию **!**

5.5. Как добавить нового пользователя

Добавить нового пользователя можно командой adduser.
Пример (добавить пользователя user1):
“*adduser user1*”

5.6. Как сохранить текущую конфигурацию

Сохранить директорию /etc/ можно принудительно командой “*writeflash*”.

Для сохранения всех измененных данных, например установленные пакеты и пользовательские данные, необходимо перезагрузить или выключить систему. Системные “скрипты” сохранят все измененные данные в архив.

5.7. Как перезагрузить или выключить систему

Перезагрузить систему можно командой “*reboot*”.
Выключить систему можно командами “*shutdown –h now*”, “*halt*”.

5.8. Как узнать конфигурацию сетевых интерфейсов

Конфигурацию сетевых интерфейсов можно запросить командой “*ifconfig*”.
Конфигурацию выбранного интерфейса можно запросить, добавив имя интерфейса параметром команды “*ifconfig*”.
Команда “*ifconfig*” показывает только настроенные интерфейсы (статус UP).
Список всех интерфейсов и “сырую” статистику можно запросить командой “*cat /proc/net/dev*”

5.9. Как посмотреть таблицу маршрутизации

Текущую таблицу маршрутизации можно посмотреть командой “*route*”.

5.10. Как посмотреть загрузку системы

Текущую загрузку системы можно посмотреть командами “*top*” и “*htop*”.
Поля имеют следующие значения (проценты, относительно всех процессоров):

- us = user, загрузка процессоров непосредственно приложениями.
- sy = system, загрузка процессоров ядром операционной системы и драйверами.
- id = idle, простой процессоров.

Команда “uptime” в поле “load average” отобразит среднюю загрузку за 1, 5 , 15 минут.

5.11. Как посмотреть список процессов

Список можно запросить командой ps.
Также можно отфильтровать процессы командой “grep” или пролистать их командой “less”.

Пример:
“ps aufx | grep init” отобразит все строки выданные “ps” отфильтровав их по строке “init”.

5.12. Как удалить процесс

Удалить процесс можно командой "*kill* PID", где PID – идентификатор процесса. Убить процесс (завершить аварийно) можно добавив параметр -9 к команде. "kill -9" рекомендуется использовать если "kill" без параметра не удалила процесс.

Удалить все процессы по имени можно командой killall (используйте с осторожностью).

5.13. Где хранятся сообщения системы

Типично все сообщения служб и системы хранятся в директории /var/log/ . Сообщения ядра ОС попадают в файл kern.log, сообщения загрузки системы в файл dmesg . Сообщения ядра и основных служб попадают в файл messages.

Посмотреть сообщения можно командой "*less*".
Смотреть поступающие сообщения можно командой "*tail* -f".

Примеры:

"*less* /var/log/messages" покажет сообщения в файле "messages"

"*tail* –f /var/log/messages" покажет все поступающие сообщения ядра и базовых служб.

5.14. Как настроить доступ к файлам Windows

Подключить сетевой диск Windows можно командой mount.

- "mount -t smbfs -o username=user1 //192.168.1.2/folder1 /mnt/folder1/" подключит сетевую директорию folder1 с компьютера IP=192.168.1.2 в директорию /mnt/folder1 используя имя пользователя user1 и пароль, который пользователь введет по запросу команды mount.

Получить доступ к файлам сервера можно по протоколу SSH (SFTP/SCP). Для этого можно использовать программу winscp, или установить дополнение winscp для far или total commander. Подключаться к серверу нужно используя имеющие учетные записи.

5.15. Работа с пакетами Debian

Пакетами Debian управляет программа dpkg.
Ее задача отображать, устанавливать и удалять пакеты, а также отслеживать зависимости пакетов.

Поверх программы работает программа apt. Ее задача устанавливать пакеты из источников (CD, Internet,…), проверять зависимости через dpkg и устанавливать все пакеты, необходимые для удовлетворения зависимостей.

Примеры:

- "apt-cache search | grep asterisk" отобразит все доступные пакеты, содержащие слово "asterisk".
- "dpkg –l | grep asterisk" отобразит все установленные пакеты, содержащие слово "asterisk".
- "apt-get install asterisk" установит все пакеты, и все пакеты необходимые для пакета "asterisk".

Список источников пакетов содержится в файле “/etc/apt/sources.list”.

5.16. Список основных предустановленных пакетов Debian

acl	
apache	Web сервер Apache2
asterisk	IP АТС Asterisk
dahdi	Драйвер для периферийного оборудования Asterisk
fuse	Драйвер файловой системы FUSE
htop	Отображение загрузки ОС и памяти процессами
ifenslave	Объединение сетевых интерфейсов (bonding)
iotop	Отображение файловой загрузки процессами
ipcad	Сбор статистики соединений
joe	Редактор
mc	Файловый коммандер
minicom	Терминал
netcat	Утилита для создания трубы через tcp,udp. Для отладочных целей.
nfs-kernel-server	NFS сервер
openntpd	Сервер и клиент времени
openssh	Ssh сервер и клиент
openswan	Ipsec сервер и клиент
openvpn	VPN сервер поверх TCP,UDP
perl	Интерпретатор perl5
php5	Интерпретатор PHP5
pppd	Сервер и клиент PPP
pppoed	PPP через Ethernet (PPPoE)
quagga	Сервера динамической маршрутизации rip, ripng, ospf, bgp
samba	Сервер и клиент SMB
tcpdump	Sniffer
Top	Отображение загрузки ОС и памяти процессами

6. Конфигурация физического уровня E1

Приведенные в данном разделе параметры определяются в конфигурационном файле **/etc/dahdi/system.conf**. После его изменения необходимо перезагрузить систему или загрузить конфигурацию командой "**/etc/init.d/pacs restart**".

6.1. Конфигурация E1 портов

Ключевым словом **span** описываются параметры конкретного порта.

span = <span_num>,<timing>,<LBO>,< framing>,<coding>[,crc4]

где

span_num – номер порта E1 (от 1 до максимального номера порта в плате)

timing – использовать ли порт как источник синхронизации

0 – порт адаптера ведущий по E1

1 и более – порт ведомый по E1 и является одним из источников синхронизации адаптера.

LBO – параметр не используется, ставить 0.

Framing – тип телефонной сигнализации, ставить ccs или cas.

Coding – кодирование в линии, может принимать значения ami или hdb3

Crc4 – разрешить проверку и генерацию crc4 (не обязательный параметр)

Каждому порту E1 должна быть сопоставлена запись span в конфигурационном файле.

6.2. Выбор канальных интервалов для передачи данных

Каждому порту E1 соответствует 31 канальный интервал (TS0 отвечает за формат фрейма и в передаче данных не участвует). Нумерация TS в системе сквозная – для порта 1 соответствуют TS1..TS31, для порта 2 – TS32..TS62 и т.д. Чтобы коммутировать группу TS из E1 в сетевой интерфейс, используется ключевое слово nethdlc:

nethdlc=<**S**>-<**E**>

где

S – номер начального TS,

E – номер конечного TS

Приведем пример.

nethdlc=2-13

В данной конфигурации 12 TS первого порта, начиная со 2 и заканчивая 13-м, будут сконфигурированы как один канал передачи данных.

Диапазон TS может задаваться и через запятую, перечислением. Например, то же самое можно описать как:

nethdlc=2,3-13

Описанная данным образом группа каналов образует в Linux сетевой интерфейс с именем hdlc0. Следующая заявленная команда nethdlc будет соответствовать hdlc1 и т.д.

Кроме имен «по умолчанию» (hdlc0, hdlc1..), через двоеточие можно задать собственное имя интерфейса:

nethdlc=2-13:**ifname**

6.3. Коммутация каналов

Группы канальных интервалов могут коммутироваться между собой. Эта опция задается ключевым словом dacs:

dacs=<**S**>-<**E**>:<**D**>

где

S – номер начального TS группы,

E – номер конечного TS группы,

D – TS назначения

Коммутация каналов и соответствующей им CAS сигнализации задается ключевым словом dacsrbs.

Примеры:

dacs=3-5:10

означает, что канальные интервалы с 3,4,5 коммутируются с 10,11,12 без коммутации CAS сигнализации.

dacsrbs=3-5:10

означает, что канальные интервалы с 3,4,5 коммутируются с 10,11,12 с коммутацией CAS сигнализации.

6.4. Утилита ecfg

Утилита ecfg (emcfg для 1 и 2-х канальной версии сервера) может быть использована как простой анализатор E1. Утилита ecfg запускается со следующими параметрами в командной строке Linux:

ecfg –i N

где

N – номер порта E1, начиная с 0 [0..7]

После запуска ecfg на экране отображается главное меню, где содержится информация о версии прошивки платы E1, а также статус выбранного порта E1.

```
Quasar monitor v.1.14 26/08/2008 Updates: http://parabel.ru/
PMC/chan=0/0, conf. file="/etc/emcfg/quasarm0_0.cfg"
HW/FW/REV version=10/10/e, driver verision=2.0.3

Line status: LOS=On , AIS=Off
Frame status: LOF=On , Sa4..8=00000, RAIS=Off
CAS Multiframe: CAS LOM=Off, XYXX=0000
CRC4 Multiframe: CRC4 err=Off, LOC=On , E bit=On
Err counters: HDB3=0, FAS=0, CRC4=0
ABCD status: 00000000 00000000 00000000 00000000

1. Configuration >>
2. Status >>
3. Test >>
0. Quit
```

7. Конфигурация канального уровня E1

Для каждого сконфигурированного в пункте **6.** Конфигурация физического уровня E1 HDLC интерфейса необходимо дополнительно задать протокол канального уровня. Это делается командой **sethdlc**. Для нее необходимо задать имя интерфейса и протокол:

sethdlc ifname proto

где

ifname – имя HDLC интерфейса, например, hdlc0

proto – один из протоколов:

- hdlc – передача IP пакетов в HDLC формате
- hdlc-eth – передача пакетов с Ethernet заголовками (мост)
- cisco – Cisco HDLC протокол
- cisco-bridge – Cisco HDLC bridge (инкапсулирует кадры Ethernet, мост)
- fr – frame relay
- ppp – синхронный PPP без аутентификации

После того, как канальный протокол задан, HDLC интерфейс представляет собой обычный сетевой интерфейс Linux. На него можно ссылаться при задании маршрутов.

Вообще говоря, вызов команды **sethdlc** может осуществляться с консоли или из любого удобного для пользователя системы скрипта. Тем не менее, удобнее воспользоваться специально созданным для этого командным файлом **/etc/init.d/pacs**, который вызывается автоматически при старте системы.

В секции start этого файла приведен пример создания bridge-группы на двух E1 портах, а также создание двух VPN соединений.

8. Конфигурация IP интерфейсов (ifconfig)

Назначение и удаление IP адресов на интерфейсах передачи данных осуществляется командой **ifconfig**.

ifconfig без параметров выводит список интерфейсов с указанием IP адресов, масок и ряда других параметров, а также выводит статистику пакетов по интерфейсам.

Для назначения IP адресов используется следующий синтаксис:

ifconfig <interface> [address] [options]

где,

interface – имя интерфейса (eth0, hdlc1 и т.д.)

address – IP адрес интерфейса (например, 100.0.0.1)

Опции:

[netmask <address>] – установить IP маску

[broadcast <address>] – установить широковещательный адрес интерфейса

[pointopoint <address>] – установить адрес партнера для соединений точка-точка

[up | down] – включить/выключить интерфейс

ПРИМЕР:

ifconfig eth0 192.168.1.1 netmask 255.255.255.0

Подробная информация по конфигурации IP интерфейсов приведена здесь.

9. Конфигурация VLAN интерфейсов (vconfig)

Команда **vconfig** используется для конфигурации vlan (IEEE802.1Q) интерфейсов. Добавление виртуальных интерфейсов возможно только для интерфейса Ethernet.

Использование:

add <ifname> <vlan_id> - добавить виртуальный интерфейс с идентификатором **vlan_id** на реальный интерфейс **ifname**. Имя виртуального интерфейса будет выглядеть как **eth0.vlan_id**, где **vlan_id** – десятичное число 0..4095.

rem <vlan_name> - удалить виртуальный интерфейс с именем **vlan_name**.

Пример:

vconfig add eth0 45

ifconfig eth0.45 192.168.45.1 netmask 255.255.255.0

10. Управление мостом (brctl)

Команда **brctl** используется для добавления и удаления моста (bridge) в систему, назначения интерфейсов, подключенных к мосту и мониторинга работы моста. Интерфейсы передачи данных, подключенные к мосту, образуют группу моста и напрямую недоступны для маршрутизации. В пределах этой группы между интерфейсами происходит передача данных по протоколам моста. Маршрутизация может использоваться только между группой как целым и остальными интерфейсами, не включенными в группу моста. Группе интерфейсов, включенных в мост, приписывается виртуальный интерфейс передачи данных, которому может быть назначен свой IP адрес и маска.

Подробная информация по работе моста и протокола STP приведена здесь.

Использование:

brctl <command> [parameters]

команды могут быть следующими:

addbr <brname> – добавить новый мост в систему с именем **brname**. Имя указывается администратором и может быть произвольным, например, **br0**, **br1** и т.д. Далее указанное имя может использоваться для присвоения IP адреса и маски командой **ifconfig**.

delbr <brname> - удалить мост с указанным именем из системы

addif <brname> <ifname> - добавить в группу моста с именем **brname** интерфейс **ifname**. После этого интерфейс становится недоступным для маршрутизации.

delif <brname> <ifname> - удалить интерфейс **ifname** из группы моста **brname**.

brctl stp <on|off> - включить (**on**) / выключить (**off**) поддержку протокола Spanning Tree. Протокол STP предотвращает образование петель и обеспечивает выбор наилучшей топологии сети.

brctl show – вывести на консоль список мостов, существующих в системе

brctl showmacs <brname> - вывести на консоль список MAC адресов, обнаруженных в сегменте моста **brname**.

brctl showstp <brname> - вывести статистику протокола STP для моста **brname**.

После добавления нового моста в систему и подключения к нему интерфейсов передачи данных, мосту может быть назначен IP адрес командой ifconfig.

Примечание:

Мост может включать интерфейсы “выглядящие” как Ethernet (у них должны быть 6-байтные адреса отправителя и получателя) и интерфейсы должны иметь одинаковые MTU.
Для hdlc интерфейсов, на канальном уровне могут использоваться протоколы hdlc-eth, cisco-bridge (см. “7. Конфигурация канального уровня E1”).
Также можно включать в мост виртуальные интерфейсы, порожденные туннелями, например vtund, openvpn, …

Пример конфигурации моста и присвоения ему IP адреса:

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 hdlc0
brctl stp on
ifconfig br0 192.168.1.54 netmask 255.255.255.0
```

11. Статическая маршрутизация (route)

С помощью команды **route** можно вручную добавлять и удалять маршруты из таблицы маршрутизации системы. **route** без параметров выводит известные системе маршруты.

ИСПОЛЬЗОВАНИЕ:

add [-net | -host] IP [netmask NM] [gw GW] [metric N] [dev IF] – добавить маршрут

del [-net | -host] IP [netmask NM] [gw GW] [metric N] [dev IF] – удалить маршрут

где,

IP – ip адрес сети или хоста, на которые указывает маршрут. В случае указания маршрута на сеть необходимо задать параметр **netmask**.

NM – сетевая маска, например 255.255.255.0.

Для указания маршрута «по умолчанию», вместо **IP** и **NM** можно использовать ключевое слово **default**.

GW – ip адрес шлюза (если необходимо)

metric N – метрика маршрута, где N – десятичное число 0..15. Метрика используется сервисом динамической маршрутизации и должна соответствовать числу промежуточных IP шлюзов до указанной подсети (хоста). Сети, доступные из маршрутизатора напрямую, должны иметь метрику 0, доступные через два шлюза – метрику 2.

IF – имя интерфейса, через который должны отправляться пакеты (eth0, hdlc0 и т.д.).

Примеры:

route add default gw 100.0.0.1

route add –net 192.168.1.0 netmask 255.255.255.0 gw 100.0.0.1

Подробная информация по таблице маршрутизации и использованию команды route приведена здесь или здесь.

12. Сетевая статистика (netstat)

Команда **netstat** полезна для получения информации о состоянии сетевой системы Linux. Она позволяет выводить информацию о маршрутах, сетевых интерфейсах и соединениях.

Описание команды приводится здесь.

13. Динамическая маршрутизация

Динамическая маршрутизация в сервере PACS-E1 реализована на пакете Quagga.

Quagga - пакет программ, реализующих протоколы маршрутизации, основанных на TCP/IP и поддерживает такие протоколы как RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4, и BGP-4+. Quagga также поддерживает BGP Route Reflector и Route Server behavior. В дополнение к традиционному протоколу IPv4 Quagga также поддерживает протоколы маршрутизации для IPv6.

Подробное описание пакета и его конфигурация описаны здесь.

По умолчанию, демоны динамической маршрутизации не запущены. Примеры конфигурационных файлов по каждому протоколу маршрутизации можно найти в директории /usr/share/doc/quagga/examples/.

14. Фильтрация и NAT (iptables)

Управление встроенным фильтром ОС Linux осуществляется командой **iptables**. Конфигурирование фильтра сводится к заданию правил распознавания пакетов и задания действий, в случае обнаружения нужного пакета. Правила разбиты на цепочки. Цепочки правил разбиты на основные группы – таблицы. Наиболее часто используемыми таблицами являются **nat** и **filter**. При манипуляции с цепочками правил администратор должен указывать таблицу, которой они принадлежат.

В каждой таблице существует предопределенный набор цепочек:

Таблица **filter**:

INPUT - все пакеты, идущие непосредственно на IP адрес маршрутизатора

FORWARD – все маршрутизируемые пакеты

OUTPUT – все пакеты, созданные и отправленные самим маршрутизатором

Таблица **nat**:

PREROUTING – пакеты до маршрутизации

OUTPUT - все пакеты, созданные и отправленные самим маршрутизатором

POSTROUTING – пакеты после маршрутизации

Использование iptables:

iptables –[AD] chain rule [options]

iptables –I chain [rulenum] rule [options]

iptables –R chain rulenum rule [options]

iptables –D chain rulenum [options]

iptables –[LFZ] [chain] [options]

iptables –N chain

iptables –X chain

iptables –P chain target [options]

где,

chain – имя цепочки

rule – спецификация правила поиска пакета, см. задание правил

rulenum – номер правила в цепочке

target – с ключом **–P** означает действие по умолчанию (политика цепочки по умолчанию)

14.1.Команды

-N – создать новую цепочку

-X – удалить цепочку

-A – добавить правило к выбранной цепочке

-D – удалить правило из цепочки (указать номер правила или его спецификацию)

-R – заменить правило с заданным номером

-I – вставить правило перед правилом с номером rulenum

-L – вывести список всех правил указанной цепочки

-F – стереть все правила в цепочке (если цепочка не указана, стереть все)

-Z – стереть счетчики пакетов во всех правилах цепочки

14.2. Задание правил обнаружения пакетов (поле rule)

-p protocol – протокол (**tcp**, **udp**, **icmp**, **all**)

-s addr/[mask] – IP адрес отправителя пакета

-d addr/[mask] – IP адрес получателя пакета

Для протокола tcp допускаются также условия:

--source-port port[:port] – порт (или диапазон портов) отправителя tcp

--destination-port port[:port] – порт (или диапазон портов) получателя tcp

--syn – обнаруживать все пакеты с установленным флагом SYN и сброшенными флагами ACK, FIN (tcp пакеты, инициирующие соединение)

Для протокола udp допускаются условия:

--source-port port[:port]– порт (или диапазон портов) отправителя udp

--destination-port port[:port] – порт (или диапазон портов) получателя udp

Большинство полей допускает задание с инверсией (символ «!»).Например, **-p !tcp** означает все протоколы, кроме tcp.

14.3. Действия при обнаружении пакета (-j опция)

-j target – действие, которое необходимо выполнять при обнаружении пакета. Поле target может означать имя другой цепочки или одно из предопределенных действий.

Для таблицы **filter** действия задаются ключевыми словами:

ACCEPT - пропустить пакет

DROP - выбросить пакет

RETURN - прекратить просматривать текущую цепочку и вернуться в предыдущую

Для таблицы **nat**, цепочка **POSTROUTING**:

SNAT – преобразовать адрес отправителя, дополнительно указать опцию **--to-source**

--to-source ipaddr[-ipaddr][:port-port] – адрес отправителя после преобразования будет выбран из диапазона **ipaddr-ipaddr**. Если протокол tcp или udp, дополнительно можно указать диапазон портов отправителя.

Для таблицы **nat**, цепочки **PREROUTING** и **OUTPUT**:

DNAT – преобразовать адрес получателя, дополнительно указать опцию --to-destination

--to-destination ipaddr[-ipaddr][:port-port] – адрес получателя после преобразования будет выбран из диапазона **ipaddr-ipaddr**. Если протокол tcp или udp, дополнительно можно указать диапазон портов получателя.

14.4. Дополнительные возможности iptables

В данном описании возможности iptables описаны не полностью, для более полной справки обращаться к документации.

15. Управление трафиком (tc)

Сервер PACS-E1 предоставляет широкие возможности по управлению пропускной способностью для различных видов трафика. Так, IP пакеты могут быть классифицированы по различным признакам и в зависимости от этого помещены в очереди на отправку с разными приоритетами и разной пропускной способностью. Таким образом, клиентам, подключенным к серверу, может быть обеспечен сбалансированный доступ к каналам передачи данных. Управление трафиком осуществляется с помощью утилиты **tc**. Вопросы управления пропускной способностью подробно описаны в документе.

16. Дополнительная литература

На CD с документацией прилагаются следующие статьи в русском переводе:

1. Энциклопедия сетевого администратора Linux, версия 2.0
2. The Linux Networking Overview HOWTO
3. Сетевая поддержка в Линуксе, Linux NET-3-HOWTO.
4. Iptables Tutorial 1.1.19
5. Linux Advanced Routing & Traffic Control HOWTO
6. Quagga - расширенный пакет программ маршрутизации
7. Linux Bridge