Министерство науки и высшего образования Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

Е.В.Красавин, В.О.Трешневская

Лабораторный практикум по дисциплине «Компьютерные сети и интернет технологии»: учебное пособие

Калуга – 2024

УДК 004.62
ББК 32.972.1
Б435

Рецензенты:

Доцент кафедры «Проектирование и технология производства электронных приборов» КФ МГТУ им. Н.Э.Баумана канд. техн.наук, доц. С.В.Полпудников

Заведующий отделом «Численного анализа пассивной безопасности» Центра «ЧАиВВ» ФГУП «НАМИ», канд. техн.наук Д.Ю.Солопов

Утверждено Методической комиссией КФ МГТУ им.Н.Э.Баумана (протокол №_ от __.__2024 г., рег. Номер _____)

Красавин Е.В., Трешневская В.О.

Лабораторный практикум по дисциплине «Компьютерные сети и интернет технологии»: учебное пособие / Е.В.Красавин, В.О.Трешневская – Калуга: КФ МГТУ им. Н.Э.Баумана, 2024. -241 с.

В учебном пособии приведены теоретические сведения и характеристика исходных данных для выполнения лабораторных работ, рекомендации по их выполнению, требования к оформлению, рекомендуемые источники информации.

Учебное пособие предназначено для студентов КФ МГТУ им. Н.Э.Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

Оглавление

ВВЕДЕНИЕ

Настоящий лабораторный практикум составлен в соответствии с программой проведения лабораторных работ по дисциплине «Компьютерные сети и интернет технологии» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета информатики и управления Калужского филиала МГТУ им. Н.Э. Баумана.

Лабораторный практикум предназначен для студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия» и содержит цели и задачи лабораторных работ, основные теоретические сведения, дается описание порядка выполнения и методические указания, приведены контрольные вопросы и формы отчетов по лабораторным работам.

Выполнение лабораторного практикума позволит студентам получить и закрепить знания, умения и навыки, достижения которых является результатом освоения дисциплины «Компьютерные сети и интернет технологии».

ПРАВИЛА БЕЗОПАСНОСТИ ПРИ ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ

При выполнении лабораторных работ необходимо руководствоваться требованиями Инструкции по охране труда для пользователей персональных компьютеров (ПК) ИОТ 020-2018.

ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

FreeBSD – это операционная система, подобная UNIX, которая свободно доступна в Интернете. Она широко применяется в компаниях-провайдерах услуг Интернета, во встроенных устройствах и в любом другом месте, где важна надежность. Операционная система FreeBSD – это результат непрерывного, в течение более тридцати лет, процесса разработки, исследований и доводки. FreeBSD основана на 4.4BSD-Lite и предназначена для компьютеров Intel (x86 и Itanium®), AMD64, Alpha™ и Sun UltraSPARC®. Ведется работа по портированию и на другие архитектуры

FreeBSD используется в качестве платформы на некоторых крупнейших сайтах в интернете.

Вебсайт FreeBSD (http://www.freebsd.org) содержит массу разнообразной информации по вопросам установки и администрирования FreeBSD. Наиболее важными частями являются Справочник (Handbook), сборник FAQ (Frequently Asked Question, часто задаваемые вопросы) и архивы почтовых рассылок, однако здесь же вы найдете огромное число статей на самые разные темы. В дополнение к документации о FreeBSD на вебсайте также имеется большой объем информации о внутреннем руководстве проектом FreeBSD и о состоянии различных частей проекта. Если основной вебсайт работает слишком медленно, то можно воспользоваться зеркалом сайта. На основном сайте имеется раскрывающийся список национальных сайтов-зеркал, кроме того можно попробовать ввести адрес в формате http://www.<код_страны>.freebsd.org. Практически во всех странах существуют свои сайты, дублирующие вебсайт FreeBSD.

Cisco Packet Tracer – это симулятор телекоммуникационных сетей, он позволяет строить работоспособные модели сети, настраивать маршрутизаторы и коммутаторы (преимущественно производства фирмы Cisco Systems), в произвольных топологиях с поддержкой разных про-

токолов. В симуляторе реализованы серии маршрутизаторов Cisco 800, 1800, 1900, 2600, 2800, 2900 и коммутаторов Cisco Catalyst 2950, 2960, 3560, а также межсетевой экран ASA 5505. Беспроводные устройства представлены маршрутизатором Linksys WRT300N, точками доступа и сотовыми вышками. Кроме того, есть серверы DHCP, HTTP, TFTP, FTP, DNS, AAA, SYSLOG, NTP и EMAIL, рабочие станции, различные модули к компьютерам и маршрутизаторам, IP-фоны, смартфоны, хабы, а также облако, эмулирующее глобальные сети. Объединять сетевые устройства можно с помощью различных типов кабелей, таких как прямые и обратные патч-корды, оптические и коаксиальные кабели, последовательные кабели и телефонные пары.

Cisco Packet Tracer позволяет создавать довольно сложные макеты сетей, что зачастую нереально сделать на реальном оборудовании, проверять на работоспособность топологии. Однако, реализованная функциональность устройств ограничена и не предоставляет всех возможностей реального оборудования, но зато приспособлена для понимания основных концепций устройства вычислительных сетей.

ЛАБОРАТОРНАЯ РАБОТА №1
АРХИТЕКТУРА ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

Целью выполнения лабораторной работы является формирование практических навыков работы с сетевыми адаптерами.

Основными задачами выполнения лабораторной работы являются:

1. Выяснить основные функции сетевых адаптеров.
2. Ознакомиться с основными типами кабелей, розеток и разъемов;
3. Изготовить и протестировать патч-корд согласно заданию.
4. Создать простейшую компьютерную сеть при помощи программного продукта Cisco Packet Tracer.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Сетевой адаптер (Network Interface Card, NIC) вместе со своим драйвером реализует второй, канальный уровень модели открытых систем в конечном узле сети— компьютере. Более точно, в сетевой операционной системе пара адаптер и драйвер выполняет только функции физического и MAC-уровней, в то время как LLC-уровень обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров. Собственно, так оно и должно быть в соответствии с моделью стека протоколов IEEE 802. Например, в ОС Windows NT уровень LLC реализуется в модуле NDIS, общем для всех драйверов сетевых адаптеров, независимо от того, какую технологию поддерживает драйвер.

Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра.

Передача кадра из компьютера в кабель состоит из перечисленных ниже этапов (некоторые могут отсутствовать, в зависимости от принятых методов кодирования):

- Прием кадра данных LLC через межуровневый интерфейс вместе с адресной информацией MAC-уровня. Обычно взаимодействие между протоколами внутри компьютера происходит через буферы, расположенные в оперативной памяти. Данные для передачи в сеть помещаются в эти буферы протоколами верхних уровней, которые извлекают их из дисковой памяти либо из файлового кэша с помощью подсистемы ввода/вывода операционной системы.
- Оформление кадра данных MAC-уровня, в который инкапсулируется кадр LLC (с отброшенными флагами 01111110).
- Заполнение адресов назначения и источника, вычисление контрольной суммы.
- Формирование символов кодов при использовании избыточных кодов типа 4B/5B.
- Скрэмблирование кодов для получения более равномерного спектра сигналов. Этот этап используется не во всех протоколах — например, технология Ethernet 10 Мбит/с обходится без него.
- Выдача сигналов в кабель в соответствии с принятым линейным кодом — манчестерским, NRZI, MLT-3 и т. п.

Прием кадра из кабеля в компьютер включает следующие действия:

- Прием из кабеля сигналов, кодирующих битовый поток.
- Выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или сигнальные процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком.
- Если данные перед отправкой в кабель подвергались скрэмблированию, то они пропускаются через дескрэмблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком.
- Проверка контрольной суммы кадра. Если она неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу

LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC-кадра извлекается кадр LLC и передается через межуровневый интерфейс наверх, протоколу LLC. Кадр LLC помещается в буфер оперативной памяти.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для клиентских компьютеров и адаптеры для серверов.

В адаптерах для клиентских компьютеров значительная часть работы перекладывается на драйвер, тем самым адаптер оказывается проще и дешевле. Недостатком такого подхода является высокая степень загрузки центрального процессора компьютера рутинными работами по передаче кадров из оперативной памяти компьютера в сеть. Центральный процессор вынужден заниматься этой работой вместо выполнения прикладных задач пользователя.

Поэтому адаптеры, предназначенные для серверов, обычно снабжаются собственными процессорами, которые самостоятельно выполняют большую часть работы по передаче кадров из оперативной памяти в сеть и в обратном направлении. Примером такого адаптера может служить сетевой адаптер SMS EtherPower со встроенным процессором Intel i960.

В зависимости от того, какой протокол реализует адаптер, адаптеры делятся на Ethemet-адаптеры, Token Ring-адаптеры, FDDI-адаптеры и т. д. Так как протокол Fast Ethernet позволяет за счет процедуры автопереговоров автоматически выбрать скорость работы сетевого адаптера в зависимости от возможностей концентратора, то многие адаптеры Ethernet сегодня поддерживают две скорости работы и имеют в своем названии приставку 10/100.

Классификация сетевых адаптеров

В качестве примера классификации адаптеров используем подход фирмы 3Com, имеющей репутацию лидера в области адаптеров Ethernet. Фирма 3Com считает, что сетевые адаптеры Ethernet прошли в своем развитии три поколения.

Адаптеры **первого поколения** были выполнены на дискретных логических микросхемах, в результате чего обладали низкой надежностью. Они имели буферную память только на один кадр, что приводило к низкой производительности адаптера, так как все кадры передавались из компьютера в сеть или из сети в компьютер последовательно. Кроме этого, задание конфигурации адаптера первого поколения происходило вручную, с помощью перемычек. Для каждого типа адаптеров использовался свой драйвер, причем интерфейс между драйвером и сетевой операционной системой не был стандартизирован.

В сетевых адаптерах **второго поколения** для повышения производительности стали применять метод многокадровой буферизации. При этом следующий кадр загружается из памяти компьютера в буфер адаптера одновременно с передачей предыдущего кадра в сеть. В режиме приема, после того как адаптер полностью принял один кадр, он может начать передавать этот кадр из буфера в память компьютера одновременно с приемом другого кадра из сети.

В сетевых адаптерах второго поколения широко используются микросхемы с высокой степенью интеграции, что повышает надежность адаптеров. Кроме того, драйверы этих адаптеров основаны на стандартных спецификациях. Адаптеры второго поколения обычно поставляются с драйверами, работающими как в стандарте NDIS (спецификация интерфейса сетевого драйвера), разработанном фирмами 3Com и Microsoft и одобренном IBM, так и в стандарте ODI (интерфейс открытого драйвера), разработанном фирмой Novell.

В сетевых адаптерах **третьего поколения** осуществляется конвейерная схема обработки кадров. Она заключается в том, что процессы приема кадра из оперативной памяти компьютера и передачи его в сеть совмещаются во времени. Таким образом, после приема нескольких первых байт кадра начинается их передача. Это существенно (на 25-55 %) повышает производительность цепочки оперативная память — адаптер — физический канал — адаптер — оперативная память. Такая схема очень чувствительна к порогу начала передачи, то есть к количеству байт кадра, которое загружается в буфер адаптера перед началом передачи в сеть.

Сетевой адаптер третьего поколения осуществляет самонастройку этого параметра путем анализа рабочей среды, а также методом расчета, без участия администратора сети. Самонастройка обеспечивает максимально возможную производительность для конкретного сочетания производительности внутренней шины компьютера, его системы прерываний и системы прямого доступа к памяти.

Адаптеры третьего поколения базируются на специализированных интегральных схемах (ASIC), что повышает производительность и надежность адаптера при одновременном снижении его стоимости. Повышение производительности канала «адаптер-память» очень важно для повышения производительности сети в целом, так как производительность сложного маршрута обработки кадров, включающего, например, концентраторы, коммутаторы, маршрутизаторы, глобальные каналы связи и т. п., всегда определяется производительностью самого медленного элемента этого маршрута. Следовательно, если сетевой адаптер сервера или клиентского компьютера работает медленно, никакие быстрые коммутаторы не смогут повысить скорость работы сети.

Выпускаемые сегодня сетевые адаптеры можно отнести к четвертому поколению. В эти адаптеры обязательно входит ASIC, выполняющая функции MAC-уровня, а также большое количество высокоуровневых функций. В набор таких функций может входить поддержка агента удаленного мониторинга RMON, схема приоритезации кадров, функции дистанционного управления компьютером и т. п. В серверных вариантах адаптеров почти обязательно наличие мощного процессора, разгружающего центральный процессор.

Сетевой адаптер OfficeConnect Fast Ethernet

Простой, надежный доступ к сети для предприятий малого бизнеса. Этот сетевой адаптер Fast Ethernet (модель 3CSOHO100-TX), созданный специально для сетей малого бизнеса, является идеальным решением для подключения к сети компьютеров класса Pentium с внутренними 32-битовыми слотами расширения PCI. При подключении адаптера к порту концентратора через интерфейс RJ-45 он автоматически

настраивается на текущую скорость передачи (10 или 100 Мбит/с). Это означает, что вы можете использовать адаптер в имеющейся сети Ethernet 10BASE-T, а в последующем перейти на технологию Fast Ethernet 100BASE-TX, не приобретая новый адаптер.

Подобно всем адаптерам и PC-картам 3Com, данная плата Fast Ethernet обеспечивает максимальную производительность, позволяющую быстро и эффективно работать с файлами, обмениваться электронной почтой и управлять сетевыми приложениями. Фирменная технология 3Com Parallel Tasking® выполняет одновременно обработку нескольких задач, обеспечивая максимально возможную скорость передачи данных. Благодаря этому вы будете тратить меньше времени на ожидание при приеме и передаче информации. С адаптером поставляется Windows-приложение для диагностики и настройки, обеспечивающее простую инсталляцию и обнаружение неисправностей.

Светодиодные индикаторы активности канала и состояния сети гарантируют постоянный контроль за работоспособностью.

Таблица 1.1. Особенности и преимущества сетевых плат 3Com Etherlink 10/100/1000 PCI-X

Особенности	Преимущества
Производительность	
Скорость передачи данных 1000 Мбит/с	Позволяет снизить загруженность сети и улучшить время отклика приложений, обеспечивая самое быстрое соединение с серве-
Разгрузка процессора при вычислении контрольных сумм TCP/IP/UDP	Освобождает процессор от интенсивных вычислений контрольных сумм, выполняя их в самом сетевом адаптере, повышая тем самым производительность системы и время
Объединение прерываний	Позволяет группировать несколько полученных пакетов, генерируя при их получении только одно прерывание хост- компьютера. Оптимизирует вычислительную эффективность хост- компьютера, сокращая число прерываний и максимально освобождая процессорные ресурсы для работы приложений.
Разгрузка процессора при восстановлении сегментированных пакетов TCP	Снижает загрузку центрального процессора и сокращает число прерываний, повышая производительность сети и ее масштабируемость
64-бит 100 МГц Bus Mastering DMA	Обеспечивает более эффективный обмен данными для снижения загрузки ЦП
Большой объем буферной памяти	Предохраняет от потерь пакетов внутри сетевого адаптера, используя буферную память 96 КБ.
Управление потоком в соответствии со стандартом 802.3x	Позволяет сократить число потерянных пакетов и повторных передач между адаптером и коммутатором, достигая за счет этого оптимальной производительности.

Продолжение таблицы 1.1.

Особенности	Преимущества
Надежность и масштабируемость	
Двунаправленное выравнивание нагрузки независимо от коммутатора	Позволяет объединить несколько каналов сервера в один логический канал. Выравнивает входящий и исходящий трафик сервера при подключении к коммутаторам Fast Ethernet любых производителей.
Соответствие стандарту IEEE 802.3ad	Совместимая снизу вверх поддержка агрегирования соединений и каналов для всех коммутаторов, IEEE 802^d- совместимых коммутаторов и установленных сетевых соединений
Восстановление серверных связей (Resilient Server Links)	Позволяет резервным сетевым платам (или сетевому адаптеру на материнской плате компьютера) резервировать активные сетевые адаптеры, включая адаптеры сторонних производителей
Способность восстановления после сбоев	Когда вышедшая из строя плата вновь становится исправной и возвращается к активному состоянию, программное обеспечение снова назначает ее главным сетевым интерфейсом. Определенное пороговое время позволяет избежать беспрерывного переключения между режимом работы при отказе/режимом восстановления.
Горячее подключение PC1	Позволяет заменять и добавлять сетевые платы без отключения или перезагрузки сервера, увеличивая время его непрерывной работы.

Продолжение таблицы 1.1.

Особенности	Преимущества
Управляемость	
Автоматический выбор скорости передачи данных (10/100/1000 Мбит/с)	Позволяет автоматически определять скорость и конфигурировать плату для работы на выбранной скорости (10,100 или 1000 Мбит/с) с концентратором или коммутатором
Приоритизация трафика/ расширенное качество обслуживания (QoS)	Функционирует в соответствии со стандартом 802.1p, регламентирующим назначение приоритета критичному ко времени передачи трафику, позволяя увеличить производительность мультимедийных приложений, приложений VOIP, и приложений с критическими для бизнеса задачами.
Соответствие PXE 2.0	Позволяет осуществлять инсталляцию, модернизацию и восстановление до загрузки операционной системы. Обеспечивает возможность удаленного управления при запуске программ, даже в случае, если компьютер не может загрузиться— это приводит к снижению общей стоимости владения
Совместимость со спецификацией Wired for Management (WfM)	Поддерживает отраслевой стандарт удаленного управления WfM— сокращает потребление энергии и позволяет переключаться в спящие режимы с низким потреблением энергии.

Продолженипе таблицы 1.1.

Особенности	Преимущества
Desktop Management Interface (DMI)	Позволяет использовать программное обеспечение сетевого управления для удаленного получения информации о ПК без непосредственного доступа к компьютеру.
Соответствие ACPI	Сокращает потребление энергии, а также позволяет выполнять удаленное включение ПК через шину Pci (только для компьютеров, совместимых с PC12.2)
Тактовые импульсные сигналы	Позволяет станциям удаленного управления проверять наличие специального периодического сигнала, отсутствие которого может означать отключение или кражу ПК.
Поддержка нескольких виртуальных локальных сетей (mVLAN)	Поддерживает до 64 виртуальных локальных сетей, соответствующих стандарту IEEE 802.1Q через коммутируемые соединения уровня 2, обеспечивая лучшую производительность ЛВС и позволяя сократить число узких мест в сети
Эффективный многоадресный контроль	Использует стандарт IEEE 802.1p в комбинации с коммутаторами, совместимыми с 802.1 для контроля переполнения сети многоадресными пакетами и повышения производительности в коммутируемых локальных сетях

Продолжение таблицы 1.1.

Особенности	Преимущества
Совместимость и отказоустойчивость	
Соответствие стандарту IEEE 802.3	Поддерживаются международные стандарты; обеспечивается полная обратная совместимость с сетями 802.3 Ethernet и 802.U Fast Ethernet
Совместимость	Поддерживает системы с шиной 32/64 bit 33/66 Mhz PCI
Совместимость с pci-x 1.0	Поддерживает системы с шиной 64 bit 66/100 Mhz PCI-x
Ограниченная гарантия на весь срок эксплуатации	Обеспечивается лучшим в отрасли обслуживанием и поддержкой компании 3Com.
Компакт-диск EtherCD™	Содержит простой в применении графический интерфейс пользователя, средства установки драйверов методом «укажи и нажми», расширенные возможности диагностики, утилиты, поддерживающие технологию DynamicAccess, руководство пользователя, утилиты для создания дискет, облегчающие установку и конфигурирование.

PCI Bus Gigabit Ethernet Adapters (GNIC-2000)

GNIC-2000 - это новый высокоскоростной адаптер для сети Gigabit Ethernet со скоростями 1000/100/10 Мбит/с. Данная модель имеет встроенную функцию Auto-Negotiation, которая автоматически поддерживает передачу данных со скоростями от 10Мбит/с до 1000 Мбит/с. Она плавно переключает драйверы между скоростями 1000 Мбит/с,100 Мбит/с и 10 Мбит/с, а также между режимами Full-duplex и Half-duplex. Адаптер работает в 32-битном режиме bus master. GNIC-2000 поставляется с драйверами для всех основных операционных систем. При установке адаптер полностью поддерживает самую передо-

вую технологию plug-&-play.

Основные особенности:

- VLAN, поддержка длинных кадров. Включение метки- идентификатора VLAN в передающийся пакет.
- Удаление метки-идентификатора VLAN из полученных пакетов.
- Поддержка функции управления потоком для полнодуплексных операций согласно IEEE 802.3x.
- Генерирование контрольных сумм IPv.4 для IP, TCP и UDP заголовков.
- Поддержка очередей приоритета согласно IEEE 802.1D и 802.1Q.
- Поддержка нескольких очередей приоритета для приема и передачи
- Автоматическая функция "crossover" для разных типов подключения.

USB-Bus Fast Ethernet Adapters (CNUE-01)

Внешний USB адаптер позволяет пользователям избежать трудностей при установке, заключающихся в необходимости открывать компьютер, и в то же время обеспечивает эксплуатационную гибкость благодаря переносимости адаптера.

Адаптер имеет стандартный разъем USB тип B и разъем RJ-45 для подключения кабеля типа TP (витая пара). Питание адаптера осуществляется от USB порта персонального компьютера, следовательно, нет никакого внешнего источника питания. Светодиодные индикаторы Link/Activity (Канал/Активность) и скорости 100 Мбит/с удобно расположены, чтобы пользователи могли мгновенно посмотреть состояние устройства.

Эта технология “универсального” порта полностью заменит существующее сегодня множество технологий портов. USB порты уже встроены во многие персональные компьютеры и периферийные устройства, существующие сегодня. USB отвечает технологическим требованиям обеспечения единого порта и типа соединителя для постоянно растущего количества периферийных устройств, таких, как сканеры, цифровые камеры, приводы компакт-дисков и т.д. USB поддерживает технологию plug-and-play и горячей замены устройств. Эта воз-

можность позволяет добавлять, удалять или заменять устройства без выключения персонального компьютера или другого сетевого устройства. CNUE-01 поставляется с USB кабелем для подключения к адаптеру.

Основные особенности:

- 100/10 Мбит / Fast Ethernet адаптер для шины USB.
- Возможность работы в дуплексном и полудуплексном режимах.
- Не требуется внешний источник питания.
- Поддержка автоматического согласования скоростей передачи данных 100/10 Мбит/с.
- Многофункциональные светодиодные индикаторы.
- В комплект поставки входит USB кабель.

PCMCIA Ethernet/Fast Ethernet Adapters (CNF401, CNF301, CN40BC)

PCMCIA - адаптеры для сети Ethernet/Fast Ethernet являются универсальными моделями размером с кредитную карточку и предназначены для использования в портативных компьютерах. Данные адаптеры позволяют просто и быстро подключить к сети большинство существующих laptop (notebook).

Адаптеры CN40/CNF301/CNF401 состоят их двух частей: непосредственно PC платы и сменного переходного устройства.

Основная плата вмещает все необходимые сетевые аппаратные средства, такие как Ethernet контроллер и буфер данных 16 Кб. Плата изготовлена из прочной нержавеющей стали и закрыта специальной термопластичной пленкой. Переходник для CN40BC снабжен двумя разъемами: UTP и BNC для соединения с сетью. Модель CN40BT снабжена переходником с одним разъемом UTP.

Быстрая обработка 32-битных данных при частоте 33 МГц позволяет увеличить пропускную способность до 90 Мбит/с. С адаптером CNF401 ваш портативный компьютер будет общаться с сетью в три - четыре раза быстрее, чем при использовании традиционного 16-битного PCMCIA адаптера. В то же время адаптер имеет пониженное энергопотребление (3.3В), что существенно увеличивает время работы

компьютера от батарей. Пониженное энергопотребление также способствует и меньшему нагреву, что благоприятно сказывается на надежности вашей системы. CNF401 использует новейший высокоэффективный стандарт интерфейса для портативных компьютеров - PC Card Bus. Несмотря на свою новизну, адаптер выглядит как обычная интерфейсная карта со стандартным разъемом, используемым во всех портативных компьютерах. Поставляемая в комплекте MS-DOS утилита “Card Enabling” позволяет легко использовать CNF401, не заботясь о совместимости со специальным программным программным обеспечением (Card and Socket Services). Таким образом, адаптер может использоваться в любом портативном компьютере, поддерживающим стандарт CardBus и имеющим разъемы Type II или Type III. CNF401 поддерживает режим “горячей замены".

Адаптер имеет стальной корпус, что делает его пригодным для ежедневного (частого) использования и не предъявляет особых требований к хранению. 68-штырьковый разъем позволяет легко подключать адаптер к портативному компьютеру. Дополнительный 15-штырьковый разъем обеспечивает подключение стандартного переходника на UTP (RJ-45) разъем. Через этот UTP разъем портативный компьютер и подключается к локальной вычислительной сети. Два световых индикатора отображают состояние сети (Link/Activity) и передачу данных со скоростью 100 Мбит/с. CNF401 комплектуется драйверами для Windows 95/98/NT, Novell NetWare 2x,3x,4x, и широкого ряда других операционных систем.

Основные особенности:

- пропускная способность до 90 Мбит/с
- 32-битная передача данных
- макс. частота 33 Мгц
- режим “горячей" замены
- утилита установки под MS-DOS
- режим auto-negotiation (авто-определение скорости передачи данных10/100)
- малое энергопотребление 3.3 V

CNF301

CNF301 - сетевое решение для современных ноутбуков. Эта небольшая плата размером с кредитную карточку может быть легко подключена к портативному компьютеру везде, где Вы путешествуете. При необходимости подключения к сети Вы можете установить эту карточку в режиме горячего подключения (при работающем компьютере) в любой ноутбук, оснащенный слотом типа II или типа III. Прочный корпус из нержавеющей стали гарантирует надежное подключение и работу этой карточки всякий раз, когда это нужно.

Этот PC адаптер оснащен двумя диагностическими светодиодами. Первый светодиод Link предназначен для визуального контроля за правильностью установленной связи. Второй индикатор Activity загорается всякий раз, когда CNF301 посылает или получает информацию. Это позволяет легко диагностировать работу компьютера в сети. Программное обеспечение входит в комплект каждого адаптера. В состав программного обеспечения была включена утилита MS-DOS Card Enabling Utility, так что потребители могут использовать CNF301, не беспокоясь о вопросах совместимости. В комплект также входят драйверы для поддержки Windows 95/98/NT и других популярных операционных систем.

CNF301 полностью поддерживает сети 100Мбит/с и 10Мбит/с. Единственный порт оснащен разъемом RJ-45, который обеспечивает присоединение простым щелчком. Настройка на скорость сети производится автоматически благодаря функции Auto-Negotiation.

PowerNIC CN40 - серия PCMCIA-адаптеров для сети Ethernet, предназначенных для использования в портативных компьютерах. Эти универсальные модели, размером с кредитную карточку, позволят просто и быстро подключить к сети большинство существующих laptop (notebook).

Сетевые топологии

Топология - физическая или электрическая конфигурация кабельного хозяйства и соединений сети.

Топология – это скелет сети. Существует несколько основных типов:

- Общая шина (Bus)
- Звезда (Star)
- Кольцо (Ring)
- Древовидная (Tree)
- Топология, когда все элементы напрямую соединены друг с другом (Mesh)

Выбор используемой топологии зависит от ваших условий, задач и возможностей. Или же определяется стандартом используемой сети.

Свои компьютеры и другие устройства вы можете соединить любым наиболее подходящим для вас способом, но в этом случае вам придется использовать вполне определенный стандарт, поддерживающий эту топологию.

Если вам удобно, вы даже можете часть компьютеров соединить в сеть с одной топологией, а часть в сеть с другой топологией, затем соединить сети между собой, при помощи какого-либо еще способа.

Сетевая топология «Общая шина»

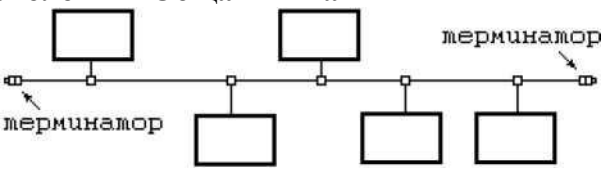


Рис.1.1. Расположение терминатора

Все компьютеры подключаются к одному кабелю (шине данных). На концах кабеля устанавливаются терминаторы. Их наличие для сетей Ethernet обязательно. По такой топологии строятся 10 Мегабитные сети 10Base-2 и 10Base-5. В качестве кабеля используется коаксиальный кабель. Повреждение общего кабеля или любого из двух терминаторов приводит к выходу из строя участка сети между этими терминаторами (сегмента сети). Отключение любого из подключенных устройств на работу сети никакого влияния не оказывает. Для сети 10Base-2 это будет выглядеть одним из следующих способов:

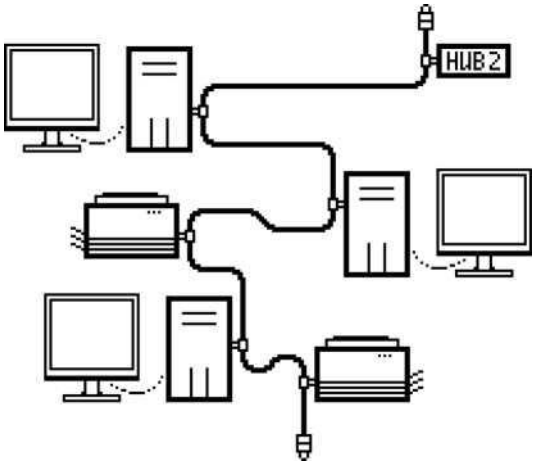


Рис.1.2. Схема топологии «Общая шина»

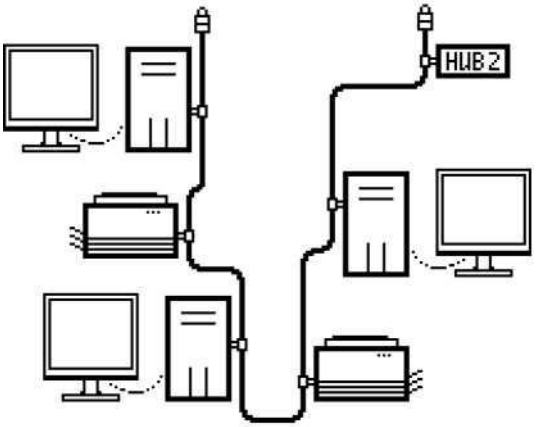


Рис.1.3. Схема топологии «Общая шина»

Данные способы абсолютно одинаковы с точки зрения топологии, но могут оказаться удобнее при прокладке.

В 100Мбитных сетях такая топология не применяется, а используется "Звезда".

Сетевая топология "Звезда"

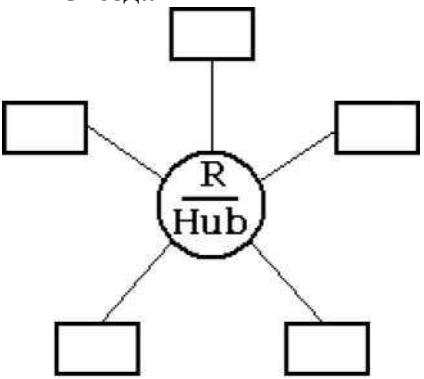


Рис.1.4. Схема топологии «Звезда»

Каждый компьютер подключен отдельным проводом к отдельному порту устройства, называемого концентратором или повторителем (ре-питер) или хабом (Hub).

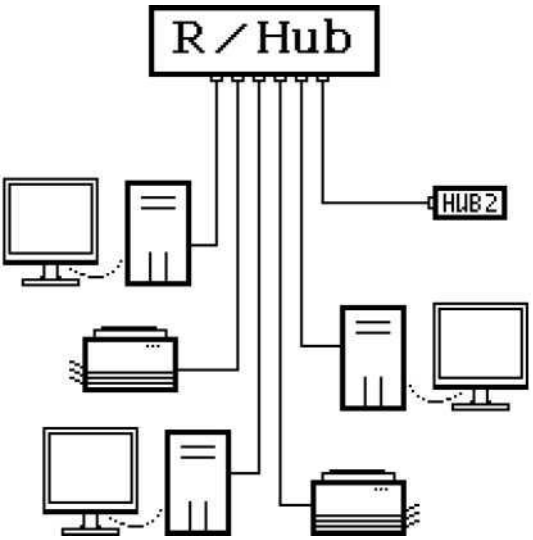


Рис.1.5. Схема соединения между устройствами и хабом

Концентраторы могут быть как активные, так и пассивные. Если между устройством и концентратором происходит разрыв соединения,

то вся остальная сеть продолжает работать. Правда, если этим устройством был единственный сервер, то работа будет несколько затруднена. При выходе из строя концентратора сеть перестанет работать.

Данная сетевая топология наиболее удобна при поиске повреждений сетевых элементов: кабеля, сетевых адаптеров или разъемов. При добавлении новых устройств "звезда" также удобней по сравнению с топологией общая шина. Также можно принять во внимание, что 100 и 1000 Мбитные сети строятся по топологии "Звезда".

Ethernet & IEEE 802.3

Стандарт Ethernet был разработан в 70-х годах в исследовательском центре PARC корпорации XEROX. В некоторых работах отмечается, что "Ethernet" - марка, зарегистрированная XEROX. Затем он был доработан совместно DEC, Intel и XEROX (отсюда идет сокращение DIX) и впервые опубликован как " Blue Book Standart" для Ethernet1 в 1980 г. Этот стандарт получил дальнейшее развитие и в 1985 г. вышел новый - Ethernet2 (известный также как DIX).

IEEE 802.3 был одобрен в 1985 году для стандартизации комитетом по LAN IEEE (Institute of Electrical and Electronics Engineers) и вышел под заголовком: "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications." Этот стандарт устанавливает общие правила по передаче данных в локальных сетях.

Ethernet и IEEE802.3 описывают схожие технологии. Обе являются CSMA/CD локальными сетями. Обе технологии являются широковещательными технологиями. Другими словами, все станции видят все фреймы (frame), даже если они предназначены не для этой станции. Каждая станция должна проверять полученный фрейм для определения, является ли она, эта станция, пунктом назначения. Если это так, то фрейм передается протоколу более высокого уровня для соответствующей обработки. Обе и Ethernet и IEEE 802.3 встроены в железо (hardware).

IEEE 802.3 определяет несколько различных физических уровней, в то время как Ethernet - один.

Каждый физический уровень IEEE 802.3 имеет название, которое отражает его характеристики.

Например: 10Base5

10 - скорость локальной сети в Мегабитах в секунду

Base = baseband или Broad = broadband

5 - длина сегмента в сотнях метров (в данном случае 500)

Таблица 1.2. Физические характеристики двух стандартов

Характеристика	Ether net	IEEE 802.3				
		10 Base 5	10 Base 2	10 Base 5	10 Base T	10 Broad 36
Скорость передачи (Mbps)	10	10	10	1	10	10
Метод передачи сигнала	Base-band	Base-band	Baseband	Baseband	Baseband	Baseband
Максимальная длина сегмента (м)	500	500	185	250	100	3600
Сетевая среда (кабель)	50-Ом коаксиальный (толстый)	50-Ом коаксиальный (толстый)	50-Ом коаксиальный (тонкий)	Неэкранированная витая пара (UTP)	Неэкранированная витая пара (UTP)	75-Ом коаксиальный
Топология	Шина	Шина	Шина	Звезда	Звезда	Шина

Таблица 1.3. 10Base2 или Тонкий Ethernet

Основная используемая топология	Общая шина
Используемый провод	коаксиальный кабель 50 Ом, тонкий
Максимальная длина сегмента	185 метров
Минимальное расстояние между точками подключения	0,5 метра
Максимальное количество точек подключения к сегменту	30
Максимальное количество сегментов в сети	5

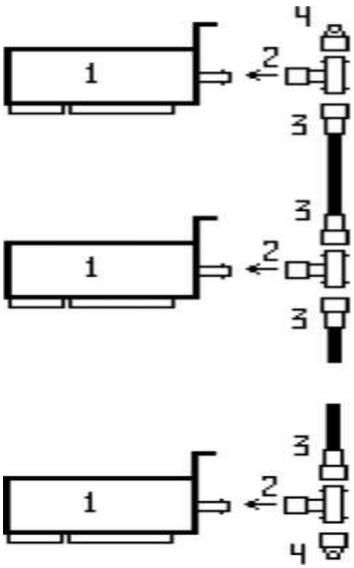


Рис. 1.6. Схема подключения кабеля к сетевой карте.

Здесь:

1 - сетевая карта, установленная в компьютере
2 - Т-коннектор
3 - разъемы на концах кабеля
4 - терминатор

Таблица 1.4. Base-T или Ethernet на витой паре

Основная используемая топология	**Звезда**
В центре звезды расположено устройство	HUB
Используется провод типа	витая пара (категории 3 или лучше)
Тип соединителя	RJ-45
Максимальное расстояние между устройствами	100метров (возможно использование другого ограничения: максимальное затухание сигнала на пути от источника до приемника не более 11,5 дб)

Для соединения устройств стандарт 10 Base-T предусматривает использование провода, имеющего две пары: одну для передачи, другую - для приема. Используются две возможные разводки кабеля в порту. MDI для DTE (Data Terminal Equipment) устройств (компьтеры, принтеры и т.д.) и MDI-X для хабов.

При подключении MDI порта к MDI-X порту используется прямая разводка кабеля. А при соединении одинаковых портов MDI и MDI или MDI-X и MDI-X используется "перевернутая" (crossover) разводка кабеля. При этом "передача" соответственно соединяется с "приемом".

Для расширения сети хабы могут каскадно соединяться друг с другом, образуя древовидную топологию с единственным хабом в вершине. Максимальное количество пользователей - 1024.

Репитеры

Сети Ethernet могут быть расширены при использовании устройства, называемого репитер (repeater-повторитель). Репитер Ethernet – это устройство, физически расположеное в сети, с двумя или более Ethernet портами. Эти порты могут быть любого типа: AUI, BNC, RJ-45 или fiber-optic, а также в любой комбинации. Основная функция репитера -

получив данные на одном из портов, немедленно перенаправить (forward) их на другие порты. Данные (сигнал) в процессе передачи на другие порты формируются заново, чтобы исключить любые отклонения, которые могли возникнуть во время движения сигнала от источника. Репитеры так же могут выполнять функцию, называемую "разделение". Если репитер определяет большое количество коллизий, происходящих на одном из портов, он делает вывод, что произошла авария где-то на этом сегменте, и изолирует его от остальной сети. Эта функция была сделана для предотвращения распространения ошибок одного сегмента на всю сеть.

У репитеров имеется отрицательная черта, заключающаяся в том, что он вносит задержку в распространение сигнала по сети. Все сети Ethernet используют протокол доступа, называемый CSMA/CD ("Carrier Sense Multiple Access, with Collision Detection"). Чтобы этот протокол работал нормально, ему необходимо иметь возможность определять возникновение коллизии. CSMA/CD определяет это возникновение, сравнивая данные, находящиеся в сети, с тем, что должны были отправить в сеть. Если определяется любое отличие, то это означает, что произошла коллизия (одновременная передача двумя устройствами) и передача немедленно прекращается. CSMA/CD затем ждет случайный отрезок времени и повторяет попытку передачи. Существует изъян в CSMA/CD, который ограничивает размер сети.

Посылаемые биты не попадают мгновенно во все точки сети, необходим некоторый небольшой отрезок времени, для того чтобы сигнал прошел по проводам и через каждый репитер в сети. Это время может быть измерено, и оно называется "задержкой распространения" ("Propagation Delay"). Если "задержка распространения" между источником сигнала и наиболее удаленным источником сети больше, чем половина размера наименьшего пакета (frame), который может существовать, тогда CSMA/CD не сможет правильно определить коллизию, и данные в сети могут быть потеряны или искажены.

Согласно проведенным разработчиками Ethernet вычислениям и измерениям, на пути сигнала в сети не может быть более 4-х репитеров и не более 5-ти сегментов, причем только к трем из них могут быть

подключены устройства. Эти выводы обычно выражаются в виде правила "5-4-3".

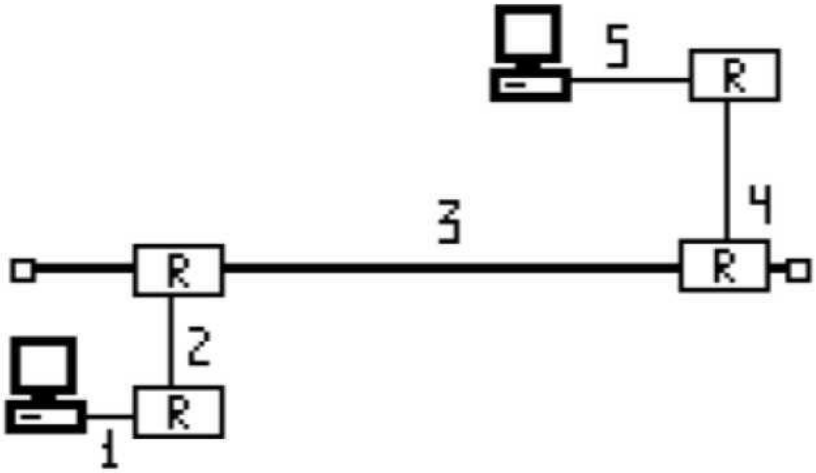


Рис.1.7. Схема правила "5-4-3"

Причем, в целом в сети может быть больше 4-х репитеров, но нас интересует только их количество между двумя любыми точками.

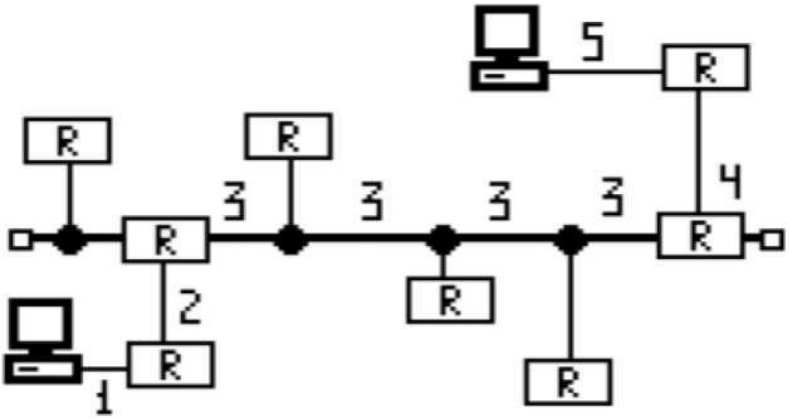


Рис.1.8. Схема правила "5-4-3"

Трансиверы

Название "Transceiver" происходит от английских слов transmiter (передатчик) и receiver (приемник). Трансивер позволяет станции передавать в и получать из общей сетевой среды передачи.

Дополнительно, трансиверы Ethernet определяют коллизии в среде и обеспечивают электрическую изоляцию между станциями. 10Base2 и 10Base5 трансиверы подключаются напрямую к среде передачи (кабель) общая шина. Хотя первый стандарт обычно использует внутренний трансивер, встроенный в схему контроллера и Т-коннектор для подключения к кабелю, а второй (10Base5) использует отдельный внешний трансивер и AUI-кабель или трансиверный кабель для подключения к контроллеру. 10BaseF, 10BaseT, FOIRL также обычно используют внутренние трансиверы.

Рис.1.9. AUI разъем (Attachment Unit interface)

Коаксиальный кабель

Коаксиальный кабель (от латинского со - совместно и axis - ось), представляет собой два соосных гибких металлических цилиндра, разделенных диэлектриком.

1 - центральный провод (жила)
2 - изолятор центрального провода
3 - экранирующий проводник (экран)
4 - внешний изолятор и защитная оболочка

Рис.1.10. Схема коаксиального кабеля

Разъемы Thin Ethernet

Разъем, расположенный на сетевой карте:

Разъем на коаксиальный кабель выглядит следующим образом:

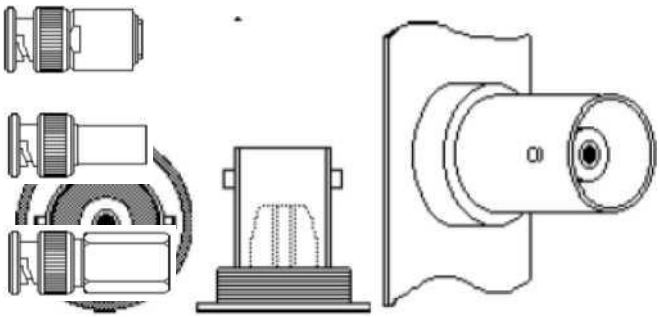


Рис.1.11. Розетка (ма-

Разъем под пайку.(типа СР150-74-ПВ)

Разъем на кабель обжимной.Требуется специальный инструмент (crimping tool)

Разъем на кабель навинчивающийся (twist- on). Инструмент для установки не требуется

Рис.1.12. Вилка прямая (папа)на коаксиальный кабель

Терминатор

Это разъем (папа) с запаяным в нем, между центральным и внешним контактами, резистором. Сопротивление резистора должно равняться волновому сопротивлению кабеля. Для сетей типа 10Base-2 или

тонкий Ethernet эта величина составляет 50 Ом. Только один терминатор в сегменте 10Base2 может быть заземлен. Для заземления используется терминатор с цепочкой и контактом на ее конце. Для 10Base5 заземление одного и только одного из терминаторов (точнее, одной из точек сегмента) обязательно.

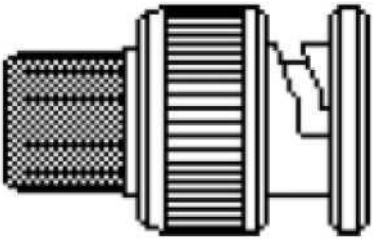


Рис.1.13. Терминатор

Hub (хаб)

Hub или концентратор - это многопортовый репитер. Наиболее распространенное применение - сети на основе витой пары 10Base-T или 100Base-TX/T4. Но бывают также хабы для сетей 10Base-2 на основе коаксиального кабеля и для сетей 10Base-F на основе волоконной-оптики.

Многие 10Mbit хабы имеют разъемы как под витую пару, обычно называемый (RJ-45), так и под коаксиальный кабель (BNC) или AUI. Что позволяет использовать сегменты коаксиального или оптического кабеля в качестве главной магистрали (Backbone) между хабами.

В хабах под витую пару используются порты MDI-X типа, что позволяет подключать компьютеры напрямую. Для соединения хабов между собой один из его портов имеет разводку MDI. Этот порт каким- либо образом выделен на корпусе устройства.

Применяются различные названия: "Cascading" или "In", или "Cross-over", или "Uplink". Нередко имеется переключатель, позволяющий переключать режим порта из MDI в MDI-X и наоборот, что позволяет использовать этот порт не для каскадирования, а для подключения обычных компьютеров. Если на вашем хабе отсутствует переключа-

тель режима порта (MDI - MDI-X), а все другие порты заняты и вам необходимо подключить еще один компьютер, то вы легко можете это сделать, просто использовав для этого "cross-over" кабель.

Такой кабель применяется для соединения двух компьютеров напрямую без хаба. Но учтите, что часто этот порт является просто cross-over вариантом одного из обычных портов, в таком случае одновременное подключение к разъемам этих портов недопустимо. Для соединения хабов по кабелю "витая пара" между собой провод (не cross-over) включается в обычный разъем (MDI-X) на одном хабе и в разъем для каскадирования на другом.

Восьмиконтактный модульный соединитель (Вилка (Plug))

Народное название "RJ-45". Вилка "RJ-45" похожа на вилку от импортных телефонов, только немного большего размера и имеет восемь контактов. Вилки делятся на экранированные и неэкранированные, со вставкой и без, для круглого и для плоского кабеля, для одножильного и для многожильного кабеля, с двумя и с тремя зубцами. Полезно вместе с вилкой на кабель устанавливать защитный колпачок.

Расплетенные и расположенные в соответствии с выбранным вами способом провода кабеля заводятся во вставку до упора, лишнее обрезается, затем полученная конструкция вставляется в вилку. Вилка обжимается. При данном способе монтажа длина расплетения получается минимальной, монтаж проще и быстрее, чем при использовании обычной вилки без вставки. Такая вилка несколько дороже чем обычная.

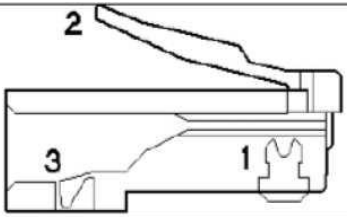
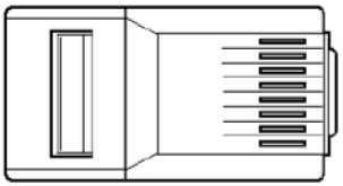
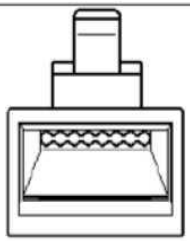
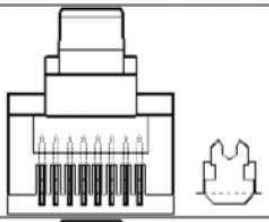
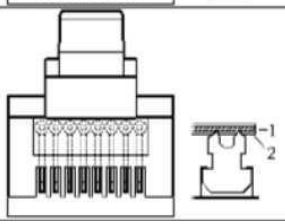
1- контакты 8 шт. 2- фиксатор разъема 3- фиксатор прово-	
Вид со стороны контактов Контакт 1 2 3 4 5 6 7 Контакт 8	
Вид со стороны кабеля	
Вид спереди На новой, неиспользованной вилке, контакты выходят за пределы корпуса.	
В процессе обжима они будут утоплены внутрь корпуса, прорежут изоляцию (2) провода и воткнутся в жнлу(1).	

Рис.1.14. Вилка "RJ-

Монтаж вилки RJ-45 на кабель

Монтаж производится одинаковым способом (568A или 568B) с обеих сторон кабеля. За исключением случая, когда вы делаете "cross-over" кабель для соединения двух компьютеров напрямую без хаба.

1. Удалите внешнюю оболочку кабеля на длину 12,5 мм (1/2 дюйма). Для этого используйте обжимной инструмент. Он позволяет обрезать кабель, удалить внешнюю оболочку и обжать вилку RJ-45. В обжимном инструменте имеется специальный нож и ограничитель для этой операции. Провода зачищать не надо.

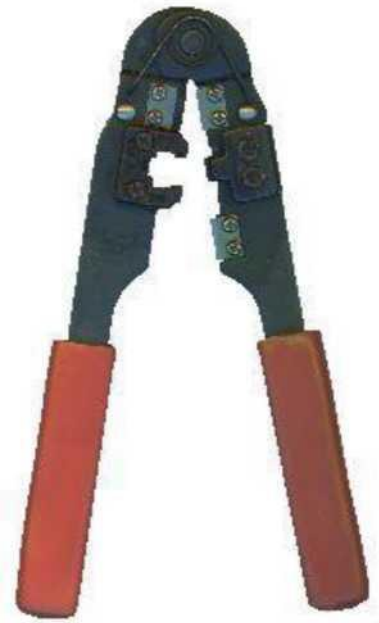


Рис. 1.15. Обжимной инструмент

Рис. 1.16. Зачистка кабеля

2. Расплетите кабель и расположите провода в соответствии с выбранной вами схемой заделки, причем длина расплетения не должна превышать 12,5 мм.

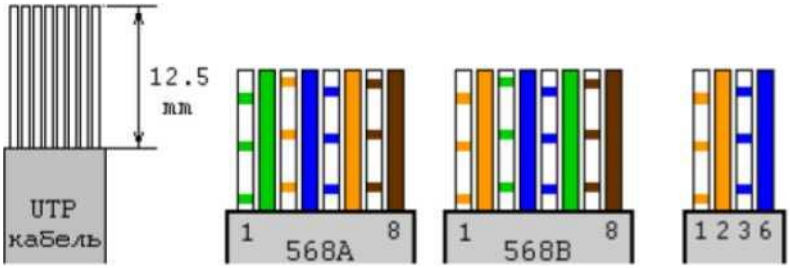


Рис. 1.17. Вариант расположения проводов в витой паре

3. Поверните вилку контактами к себе, как на рисунке, и аккуратно надвиньте на кабель до упора, чтобы провода прошли под контактами.

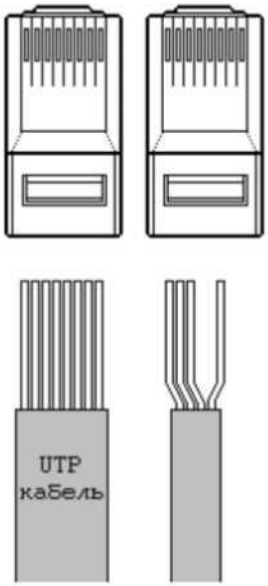


Рис.1.18. Вставление проводов в вилку

Рис. 1.19. Вилка с кабелем внутри

4. Обожмите вилку. На обжимном инструменте имеется специальное гнездо, в которое вставляется вилка с проводами и нажатием на ручки инструмента обжимается. При этом контакты будут утоплены внутрь корпуса и прорежут изоляцию проводов. Фиксатор провода также должен быть утоплен в корпус.

Рис. 1.20. Закрепление контактов в вилке.

Если у вас нет обжимного инструмента, то попробуйте обжать разъем RJ-45 тонкой отверткой. Поочередно утапливая контакты (1) 8шт. в корпус, а также фиксатор провода (3). Подложите что-нибудь под разъем, чтобы не сломать его фиксатор (2). Это не очень надежный способ монтажа, но вполне применимый.

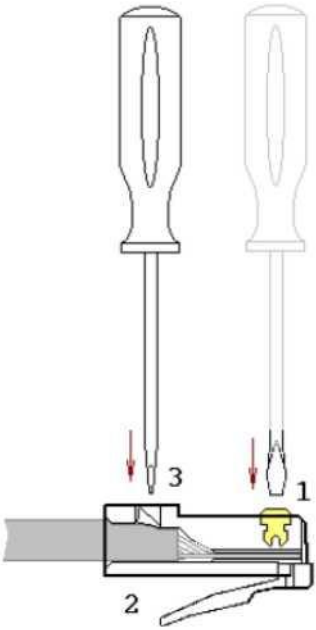


Рис.1.21. Монтаж без обжимного инструмента

Варианты заделки проводов (разводка проводов витая пара)

Кабель разделывается одинаково с обеих сторон. Если кабель содержит только две пары:

Таблица 1.5. 10Base-T/100Base-TX

Одна сторона	Цвет провода	Другая сторона
1	бело/оранж	1
2	оранж/белый	2
3	бело/синий	3
6	сине/белый	6

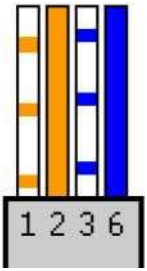


Рис. 1.22. Двухпроводный кабель

Для восьмижильного кабеля (четыре пары). Выбор варианта заделки 568A или 568B зависит исключительно от принятого в вашей сети. Оба этих варианта эквивалентны. Рекомендуется использовать первый.

Таблица 1.6. EIA/TIA-568A

Одна сторона	Цвет провода	Другая сторона
1	бело/зеленый	1
2	зелен/белый	2
3	бело/оранж	3
4	сине/белый	4
5	бело/синий	5
6	оранж/белый	6
7	бело/коричн.	7
8	коричн./белый	8

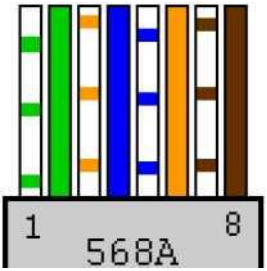


Рис. 1.23. 568A

Таблица 1.7. EIA/TIA-568B, AT&T 258A

Одна сторона	Цвет провода	Другая сторона
1	бело/оранж	1
2	оранж/белый	2
3	бело/зеленый	3
4	сине/белый	4
5	бело/синий	5
6	зелен/белый	6
7	бело/коричн.	7
8	коричн./белый	8

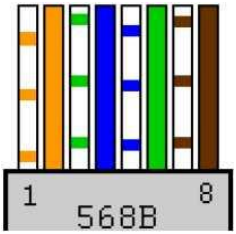


Рис. 1.24. 568B

Хорош своей надежностью, наболее современен, допускает соединение компьютеров на скорости до 100 Мбит. Но не позволяет без покупки специального устройства HUB (хаб) расширить сеть даже до трех компьютеров. Для подключения к хабу используется перевернутая развестка.

Таблица 1.8. "Cross-over" ("нуль-хабный") кабель

Одна сторона	Цвет провода	Другая сторона
1	бело/оранж	3
2	оранж/белый	6
3	бело/синий	1
6	сине/белый	2

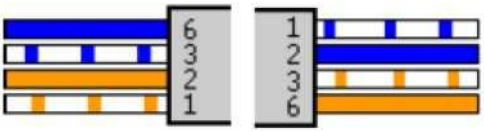


Рис. 1.25. Двухпроводный кабель

Таблица 1.9. "нуль-хабный" кабель

Одна сторона	Цвет провода	Другая сторона
1	бело/зеленый	3
2	зеленый	6
3	бело/оранж	1
4	синий	4
5	бело/синий	5
6	оранжевый	2
7	бело/коричн.	7
8	коричневый	8

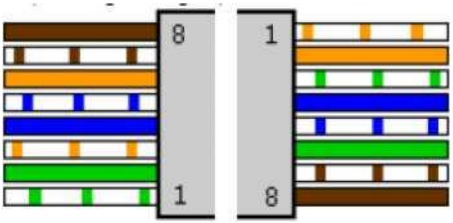


Рис. 1.26. Восьмипроводный кабель

Разводка кабеля витая пара для соединения двух компьютеров напрямую.

Кабель витая пара может быть как четырехпроводный, так и восьмипроводный. Для монтажа на кабель используются вилки RJ-45. Монтаж вилки на кабель должен осуществляться при помощи специального инструмента. Для восьмипроводного кабеля возможен как вариант показанный на рис.26, так и приведенный ниже.

Таблица 1.10. "нуль-хабный" кабель

Одна сторона	Цвет провода	Другая сторона
1	бело/зеленый	3
2	зеленый	6
3	бело/оранж	1
4	синий	7
5	бело/синий	8
6	оранжевый	2
7	бело/коричн.	4
8	коричневый	5

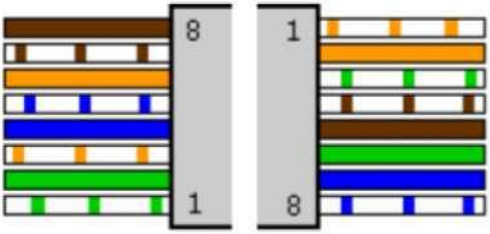


Рис. 1.27. Восьмипроводный кабель

СОЗДАНИЕ ПРОСТЕЙШЕЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ CISCO PACKET TRACER

Для выполнения лабораторной работы необходимо качать и установить Cisco Packet Tracer.

Вы можете официально скачать и использовать Cisco Packet Tracer бесплатно. Вам нужна учетная запись Cisco Network Academy для загрузки и использования Cisco Packet Tracer. Вы можете создать учетную запись Cisco Network Academy бесплатно.

В Packet Tracer 7 добавлена функция аутентификации пользователей. Пользователь Сетевой академии должен выполнить вход при первом запуске Packet Tracer. Пользователи без учетной записи Сетевой академии смогут сохранять топологии не более трех раз. Пользователь без учетной записи Сетевой академии может нажать кнопку гостевого

входа, чтобы записаться на бесплатный курс для самостоятельного изучения «Введение в Packet Tracer» и получить учетную запись netacad.com для полного доступа к Packet Tracer. Курс «Введение в Packet Tracer» поможет вам ознакомиться с основными функциями Packet Tracer.

Чтобы создать учетную запись Cisco Network Academy, перейдите на страницу https://www.netacad.com/ru/courses/packet-tracer/introduction-packet-tracer из любого веб-браузера по вашему выбору, и вы должны увидеть следующую страницу. Теперь нажмите «Зарегистрируйтесь уже сегодня!», чтобы загрузить Packet Tracer.

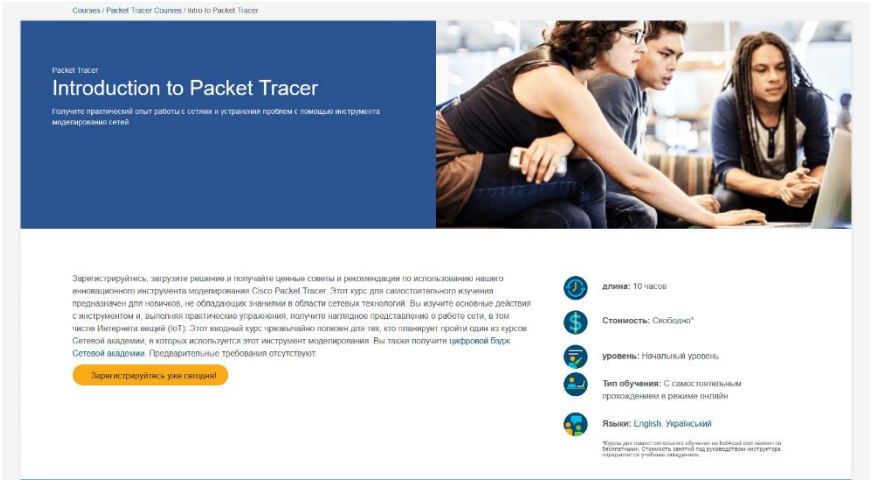


Рис. 1.28. Окно создания учетной записи

В выпадающем меню нужно нажать кнопку English. Должна открыться страница регистрации. Заполните данные и нажмите Отправить, как показано на скриншоте ниже.

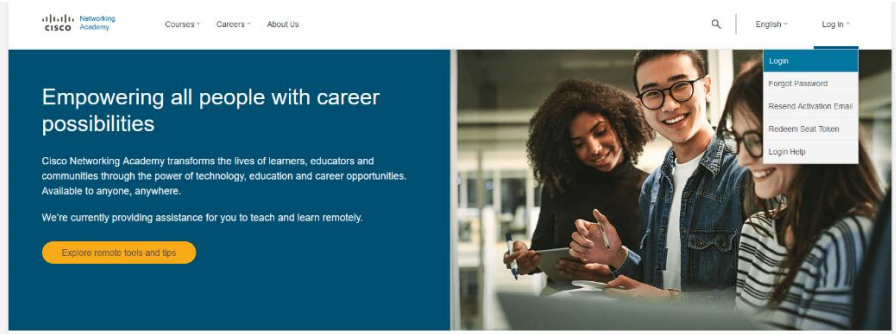


Рис.1.29. Окно создания учетной записи

После того как вы зарегистрировались и подтвердили свою учетнуюзапись, перейдите по адресу https://www.netacad.com/, и вы должны увидеть следующую страницу. Нажмите Log In -> Login, как видно на скриншоте.

После того как вы зашли, нужно нажать в верхнем меню Resource ->Download Packet Tracer.

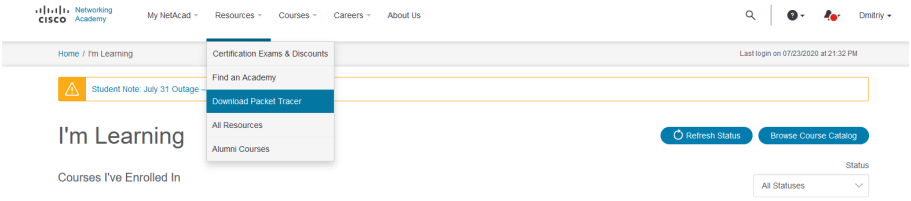


Рис.1.30. Окно загрузки

На этой странице в разделе Downloads нужно выбрать и скачать необходимую версию - для Windows, Linux, MacOS, Android или iOS.

Рис.1.31. Окно выбора версии

Устанавливаем и запускаем. При первом запуске мы увидим окно где нужно еще раз залогиниться под учетной записью netacad. Чтобы войти без учетной записи нужно нажать кнопку Guest Login в правом нижнем углу и подождать окончания таймера, после чего нажать кнопку Confirm Guest.

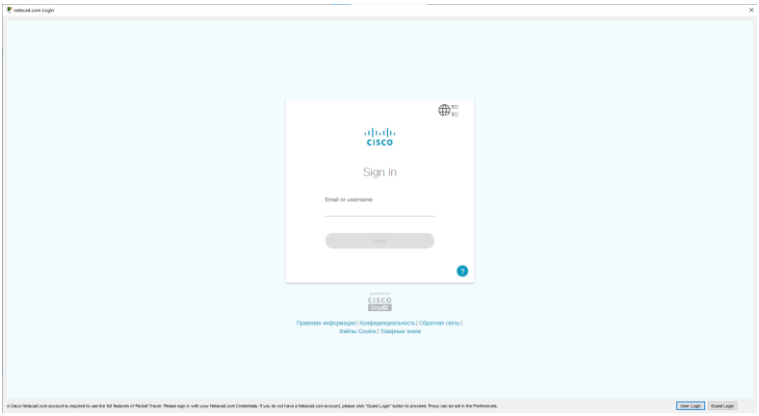


Рис.1.32. Окно авторизации

Можно начинать работать.

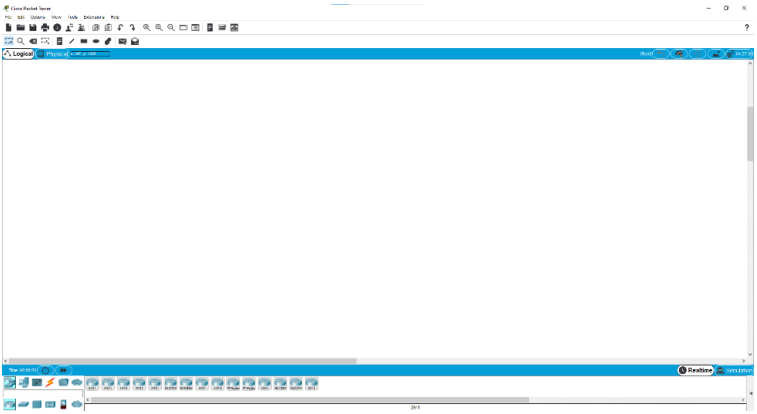


Рис.1.33. Окно программы

Окно программы и его структура представлены ниже.

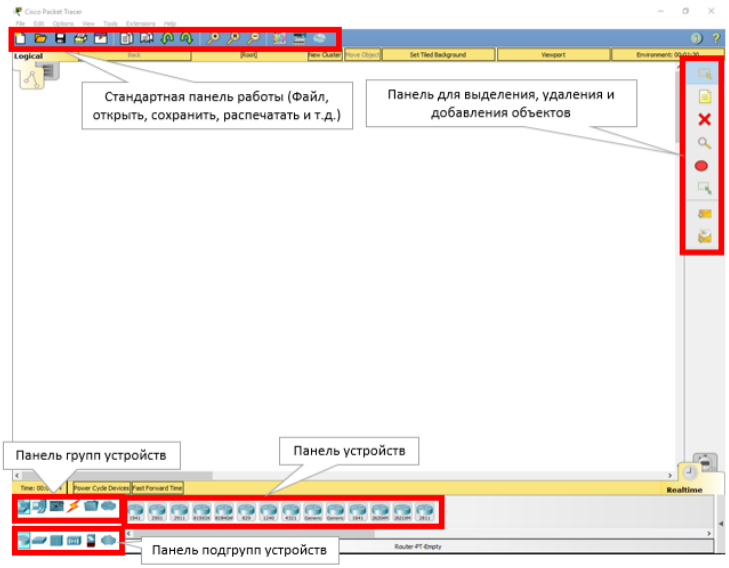


Рис.1.34. Элементы интерфейса

Для организации простейшей сети необходимо:

- Два компьютера;
- Коммутационный кабель (патч-корд).

Коммутационный кабель бывает двух видов:

- Прямой кабель (straight through cable) Для соединения типа компьютер-коммутатор, коммутатор маршрутизатор.
- Перекрестный кабель (crossover cable) Для соединения типа компьютер-компьютер, коммутатор-коммутатор, маршрутизатор-маршрутизатор.

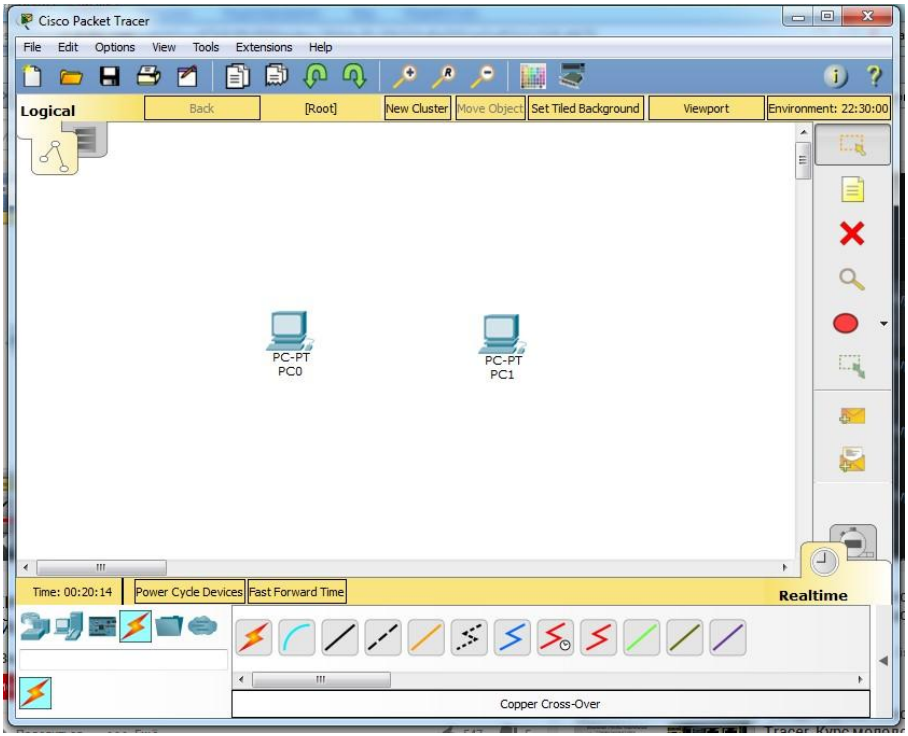


Рис. 1.35. Рабочая область Cisco Packet Tracer

Создание простейшей сети с помощью Cisco Packet Tracer:

1. Открыть Cisco Packet Tracer;
2. Выбрать компьютер и перетащить его на рабочую область

(рис.1);

3. Аналогично выбрать второй компьютер (рис. 1.35);

4. Перейти на вкладку Connections (рис. 1.36). Выбирать тип кабеля (в нашем случае перекрестный). Подключить Fast Ethernet – Fast Ethernet (рис. 1.36).

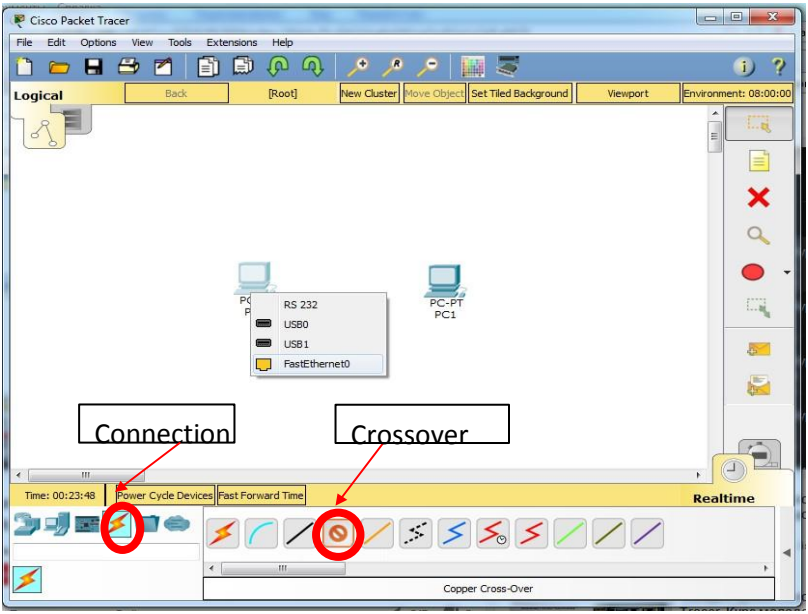


Рис. 1.36. Простейшая сеть

5. Перейти к настройке компьютеров. Перейти во вкладку Desktop-IP Configuration и ввести IP адрес (например, 192.168.1.1) (рис. 1.37).

Аналогичные действия провести со вторым компьютером.

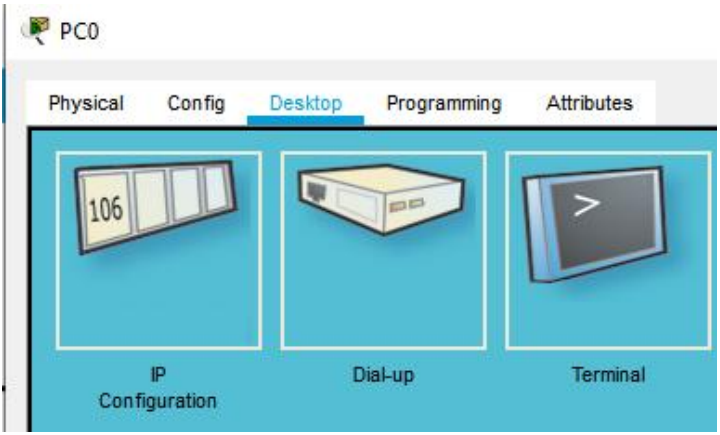


Рис.1.37. Выбор Desktop-IP Configuration

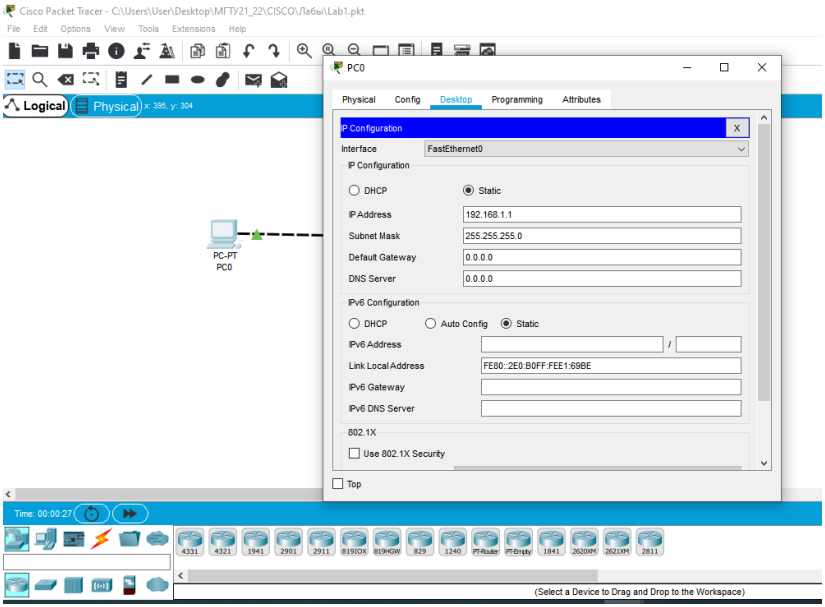


Рис. 1.38. Действия со вторым компьютером

6. Проверить соединение. Выбрать Desktop -Command Prompt.

Ввести в нашем случае ping 192.168.1.1. Результат приведен на рис. 1.39.

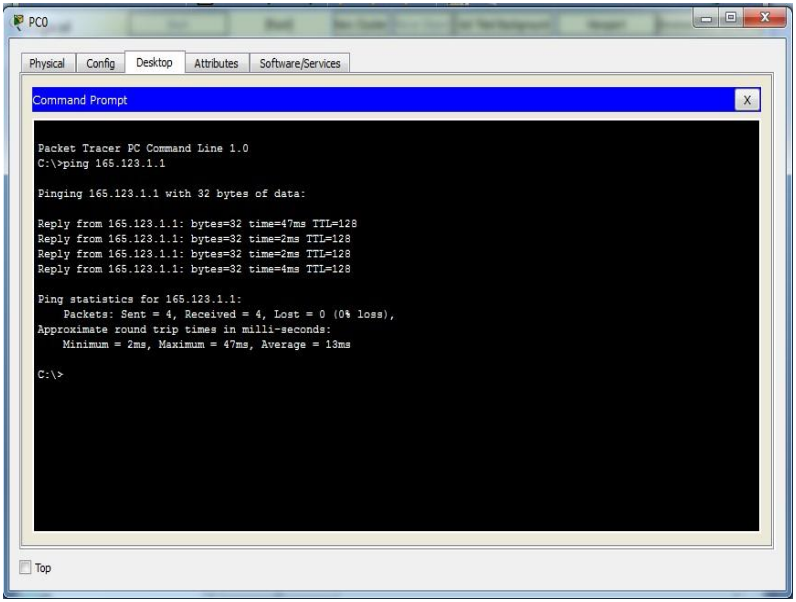


Рис. 1.39. Проверка соединения

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Под руководством преподавателя самостоятельно изготовить несколько вариантов патч-кордов и протестировать их работоспособность. Для этого необходимо:

1. С помощью обжимного инструмента подготовить (отрезать, снять изоляцию) кабель.
2. Расположить проводники в правильном порядке по цветам изоляции согласно схеме обжима (EIA/TIA-568A) для прямой или cross-over разводки.
3. Вставить проводники в модульный соединитель и закрепить обжимным инструментом.
4. Протестировать работоспособность изготовленного патч-корда

с помощью тестера.

5. Убедиться в работоспособности изготовленного патч-корда, соединив им компьютер с розеткой (для прямого соединения) и с другим компьютером напрямую (для cross-over разводки).
6. С помощью Cisco Packet Tracer создать простейшую сеть.
7. Ответить на контрольные вопросы и оформить отчет.

ФОРМА ОТЧЕТНОСТИ О ВЫПОЛНЕННОЙ РАБОТЕ, ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ

Результатами работы являются:

1. Изготовленный и протестированный патч-корд.
2. Сохраненные в файлах результаты создания сети в формате Cisco Packet Tracer (с расширением pkt)
3. Подготовленный отчет.

Отчет на защиту предоставляется в электронном или печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы со скриншотами, выводы.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите этапы передачи и приема кадра.
2. Перечислите основные отличия сетевых адаптеров серверов от клиентских компьютеров.
3. Опишите назначение процедуры автопереговоров.
4. Перечислите особенности сетевых адаптеров различных поколений.
5. Назовите преимущества использования адаптера CNUE-01.
6. Опишите основные особенности использования сетевых адаптеров PCMCIA.

7. Перечислите основные сетевые топологии.
8. Изложите концепцию построения топологии сети 10Base-2 и 10Base-5.
9. Приведите пример схемы топологии «общая шина».
10. Перечислите физические характеристики стандартов 10Base-5, 10Base-2 и 10Base-T.
11. Раскройте область применения прямой и перевернутой разводки кабелей стандарта 10Base-T.
12. Дайте определение и раскройте основные задачи репитера.
13. Изложите концепцию правила «5-4-3».
14. Раскройте значение термина трансивер.
15. Изобразите и опишите структуру коаксиального кабеля.
16. Дайте определение термину терминатор.
17. Опишите роль восьмиконтактного модульного соединителя.
18. Раскройте значение термина патч-корд.
19. Перечислите основные стандарты обжима кабеля типа витая пара.

ЛАБОРАТОРНАЯ РАБОТА №2
НАСТРОЙКА И ИСПОЛЬЗОВАНИЕ СЕРВЕРОВ В ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью выполнения лабораторной работы является получение практические навыки по работе и настройке прокси-сервера Squid под ОС FreeBSD, по настройке системы DNS в ОС FreeBSD.

Основными задачами выполнения лабораторной работы являются: научиться получать и устанавливать прокси-сервер Squidпод ОС FreeBSD, научиться настраивать и управлять прокси-сервером Squidпод ОС FreeBSD, научиться настраивать DNS-клиент и DNS-сервер в ОС FreeBSD.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Лабораторная работа выполняется под ОС FreeBSD.

Установка FreeBSD

Перед установкой надо определиться, какой образ скачать. Ниже список с описанием всех типов образов Freebsd для платформы x64, которые можно скачать.

- Bootonly - Минимальный образ по размеру. Чтобы установить с него систему, необходимо подключение к интернету во время установки. bootonly.iso 285M
- disc1 - Основная система и базовый набор программ есть на диске. Можно установить без подключения к сети. disc1.iso 656M
- dvd1 - Максимальный образ. В него входят помимо системы, пакеты программ. dvd1.iso 3G
- memstick - Стандартный образ для установки с флешки, аналог disc1. memstick.img 700M

Стандартному загрузчику FreeBSD необходим первичный раздел (MBR) или GPT раздел. Если все первичные или GPT разделы уже за-

действованы, то для FreeBSD один из них необходимо будет освободить.

Минимальная установка FreeBSD занимает около 1 ГБ дискового пространства. Однако, это очень минимальная установка, практически не оставляющая свободного места. Более реалистичным минимумом является 3 ГБ без графической подсистемы, а если будет использоваться графическая подсистема, то 5 ГБ или более. Свободное пространство также потребуется приложениям от третьих лиц.

Наиболее удобный и универсальный образ disc1, его рекомендуется использовать.

Установка FreeBSD начинается с загрузки компьютера с установочного носителя, будь то CD, DVD или USB флеш-накопитель.

Вставляем iso образ в автозагрузку и загружаемся с диска. Появляется традиционное окно приветствия с тремя вариантами продолжения:

1. **Install** - начать установку системы.
2. **Shell** - перейти в консоль.
3. **Live CD** - загрузиться в режиме Live CD.

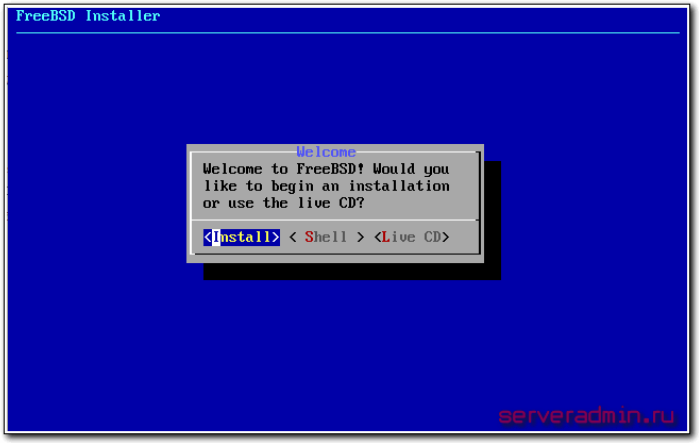


Рис.2.1. Окно приветствия

Выбираем установку. На следующем этапе будет предложено выбрать раскладку. Чаще всего достаточно стандартной, поэтому ничего не меняем, а идем дальше со стандартной раскладкой.

Дальше нужно будет указать имя новой freebsd системы. Назвать можете как угодно, это не принципиально. В случае необходимости, это имя можно будет сменить после установки.

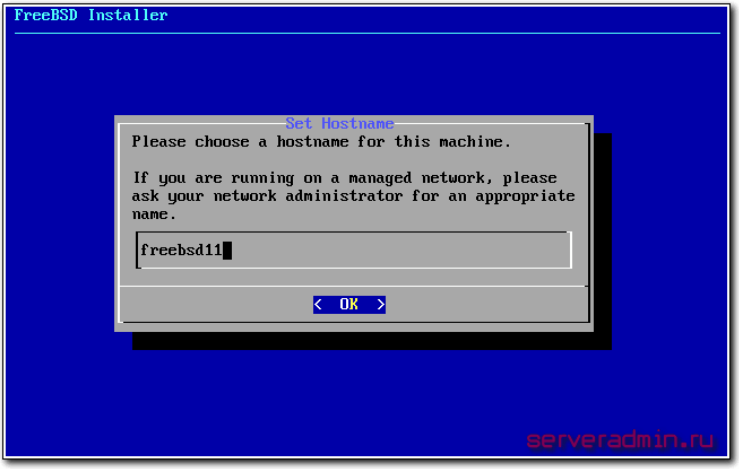


Рис.2.2. Ввод имени новой freebsd системы

Теперь выбираем компоненты, которые будут установлены. Для наших целей не надо ничего устанавливать, кроме lib32. Все, что нужно, можно потом установить последней версии из интернета.

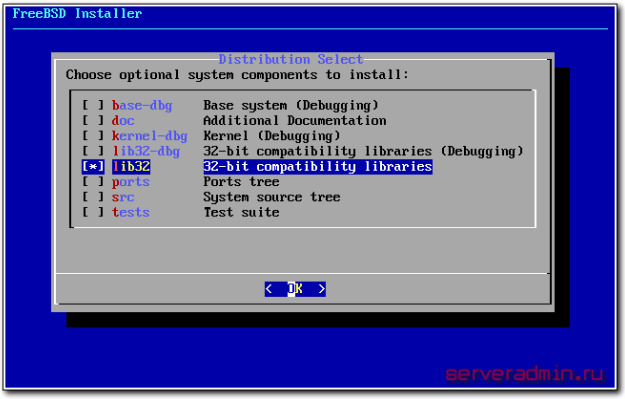


Рис.2.3. Выбор компонентов системы

На следующем этапе выбираем разбивку жесткого диска. Можно вручную указать все необходимые разделы, выбрать размер и т.д. Достаточно все установить на одном корневом разделе. Наиболее частая

рекомендация - вынести в отдельный раздел все логи, чтобы случайно заполнив все свободное место они не повесили сервер, но можно этого не делать , а следить за ротацией логов и не допускать их роста до больших размеров. Так что выбираем первый пункт - Auto (UFS).

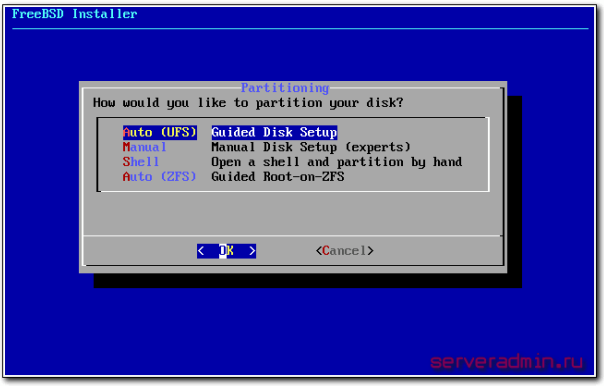


Рис.2.4. Выбор разбивки жесткого диска

Дальше у вас спросят, хотите ли вы занять все свободное место жесткого диска под систему. Если это так, то соглашайтесь. На следующем этапе указываем таблицу разделов. Рекомендуется GPT:

Рис.2.5. Ввод таблицы разделов

Проверяйте предложенную схему разбивки диска. Если все устраивает, то жмите Finish и согласитесь с применением изменений, начнется установка базовой системы.

Рис.2.6. Схема разбивки диска

Длится она буквально несколько минут. Чистая система ставится очень быстро. По ходу дела будет предложено указать пароль для root. Сделайте это. Далее нужно будет выбрать сетевой интерфейс для настройки.

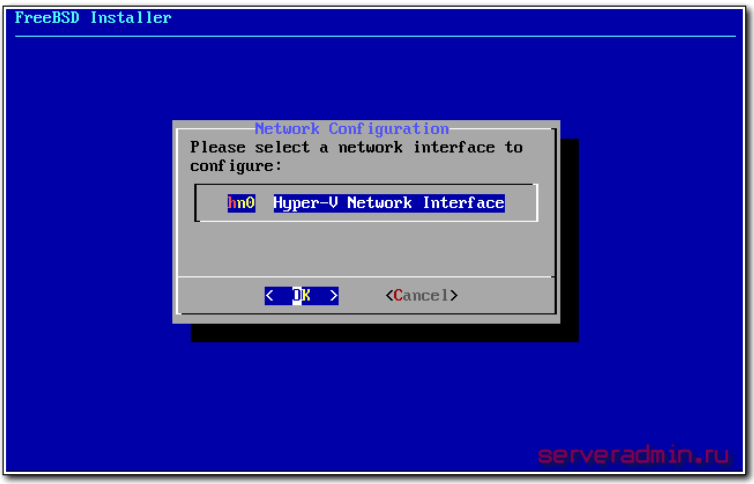


Рис.2.7. Выбор сетевого интерфейса

Для выполнения данной работы пропускаем этот пункт, нажав [Cancel].

Теперь выбираем часовой пояс, дату и время. Ничего сложного нет,

скриншоты приводить не буду. Если дата и время указаны верно, то просто выбирайте **Skip**, если есть расхождения, вручную укажите правильные. Это нововведение в 11-й версии. Раньше такого календаря и часов не было.

У нас локальное время не совпадает с временем по Гринвичу, поэтому на соответствующий вопрос отвечаем **No**.

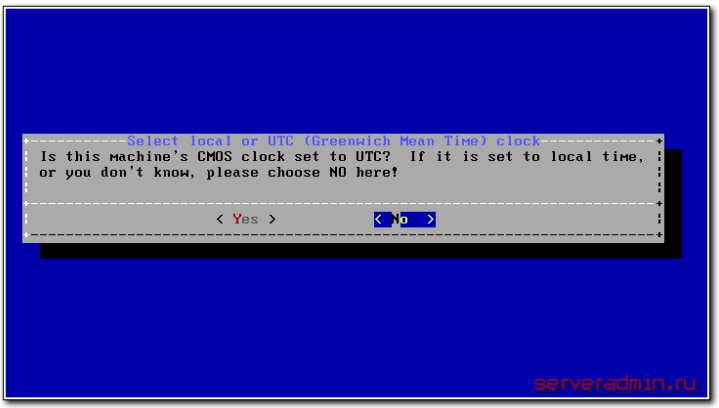


Рис.2.8. Установка времени

Выбираем регион расположения своего сервера.

Рис.2.9. Выбор региона

После выбора региона указываем часовой пояс.

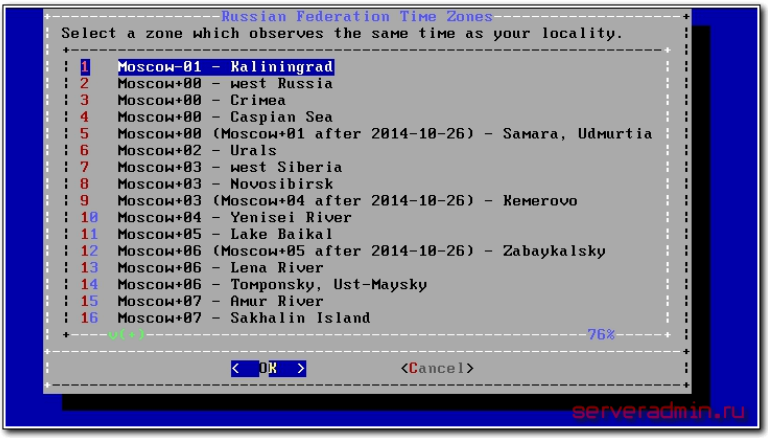


Рис.2.10. Выбор часового пояса

Установка движется к завершению. Нужно указать, какие службы вы хотите запускать автоматически при загрузке системы. Обязательно укажите **sshd**, чтобы подключаться к серверу удаленно, еще **ntpd** не помешает. Остальное на ваше усмотрение, я больше ничего не указываю. dumped стоит по-умолчанию, пусть останется.

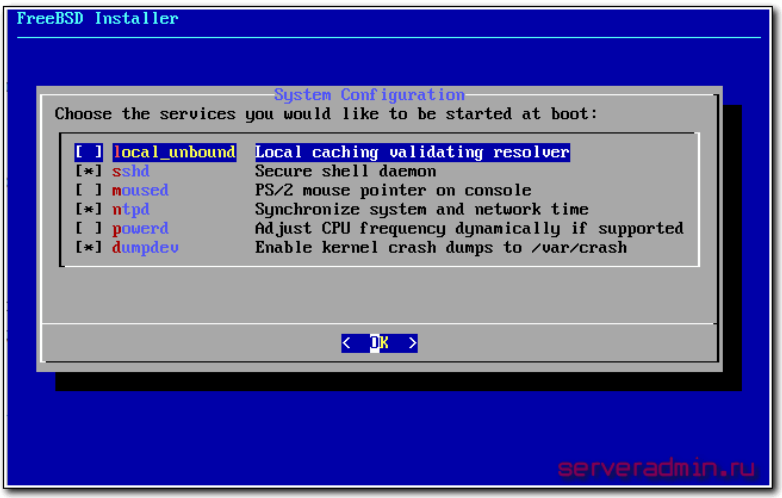


Рис.2.11. Выбор автозапуска служб

На следующем этапе нам предлагается выбрать некоторые пара-

метры безопасности. Все эти настройки можно и позже сделать.

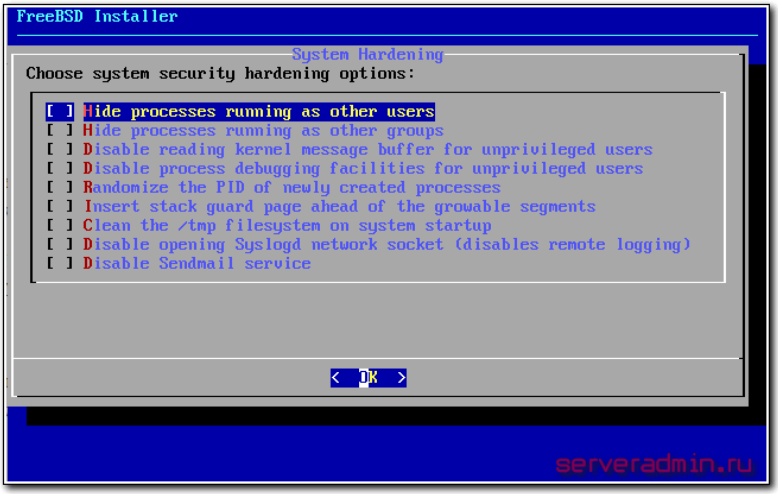


Рис.2.12. Параметры безопасности

На заключительном этапе вам будет предложено добавить пользователей в систему. Если вы этого не сделаете, то не сможете подключиться по ssh к серверу. По-умолчанию в freebsd пользователю root запрещено подключаться по ssh. Это можно исправить только зайдя локально рутом и отредактировав настройки ssh. Так что создайте хотя бы одного пользователя и добавьте его в группу wheel, чтобы можно было подключиться по ssh и сделать su для получения root доступа.

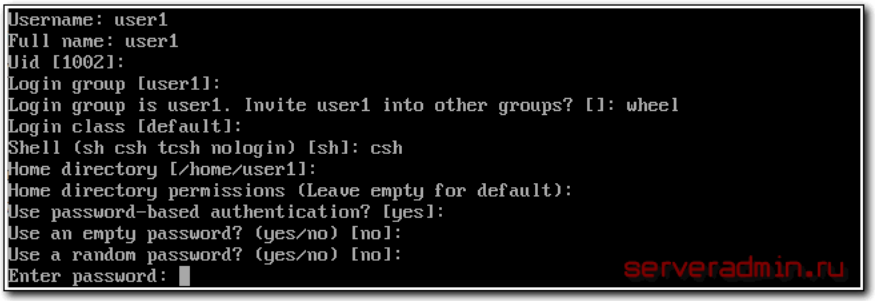


Рис.2.13. Настройка пользователя

Дальше выбирайте **Exit**, перезагружайте систему и вынимайте загрузочный диск.

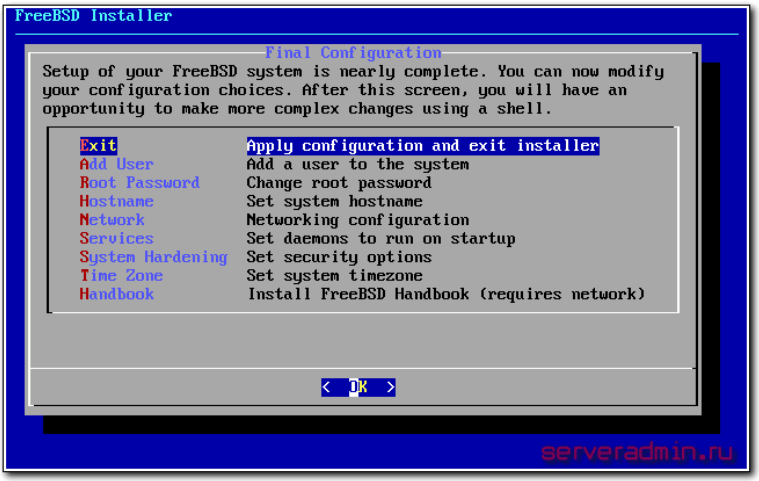


Рис.2.14. Завершение настройки

Вы должны загрузиться в свежеустановленной системе Freebsd. На этом базовая установка закончена.

НАСТРОЙКА ПРОКСИ-СЕРВЕРА SQUID И СИСТЕМЫ DNS ПОД ОС FREE BSD.

1. Настройка сетевых интерфейсов

В наши дни мы не представляем себе компьютера без сетевого подключения. Добавление и настройка сетевой карты это обычная задача любого администратора FreeBSD.

1.1. Поиск подходящего драйвера

В первую очередь определите тип используемой карты (PCI или ISA), модель карты и используемый в ней чип. FreeBSD поддерживает многие PCI и ISA карты. Обратитесь к Списку поддерживаемого оборудования вашего релиза чтобы узнать, поддерживается ли карта.

Как только вы убедились, что карта поддерживается, потребуется определить подходящий драйвер. В файлах /usr/src/sys/conf/NOTES и /usr/src/sys/arch/conf/NOTES находится список драйверов сетевых интерфейсов с информацией о поддерживаемых чипсетах/картах. Если вы сомневаетесь в том, какой драйвер подойдет, прочтите страницу справочника к драйверу. Страница справочника содержит больше

информации о поддерживаемом оборудовании и даже о проблемах, которые могут возникнуть.

Если ваша карта широко распространена, вам скорее всего не потребуется долго искать драйвер. Драйверы для широко распространенных карт представлены в ядре GENERIC, так что ваша карта должна определиться при загрузке, примерно так:

dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38 000ff irq 15 at device 11.0 on pci0

dc0: Ethernet address: 00:a0:cc:da:da:da

miibus0: <MII bus> on dc0

ukphy0: <Generic IEEE 802.3u media interface> on miibus0

ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto

dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30 000ff irq 11 at device 12.0 on pci0

dc1: Ethernet address: 00:a0:cc:da:da:db

miibus1: <MII bus> on dc1

ukphy1: <Generic IEEE 802.3u media interface> on miibus1

ukphy1: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto

В этом примере две карты используют имеющийся в системе драйвер dc.

Если драйвер вашей сетевой карты отсутствует в GENERIC, для ее использования потребуется загрузить подходящий драйвер. Это может быть сделано одним из двух способов:

Простейший способ — просто загрузить модуль ядра сетевой карты с помощью kldload. Не все драйверы доступны в виде модулей; например, модули отсутствуют для ISA карт.

Вместо этого, вы можете статически включить поддержку карты, скомпилировав собственное ядро. Информацию о том, какие параметры нужно включать в ядро, можно

получить *из /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES* и страницы справочника драйвера сетевой карты. Если карта была обнаружена вашим ядром (GENERIC) во время загрузки, собирать ядро не потребуется.

1.2. Настройка сетевой карты

Как только для сетевой карты загружен подходящий драйвер, ее потребуется настроить. Как и многое другое, сетевая карта может быть настроена во время установки с помощью bsdinstall.

Для вывода информации о настройке сетевых интерфейсов системы, введите следующую команду:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500 inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255 ether 00:a0:cc:da:da:da media: Ethernet autoselect (100baseTX <full-duplex>) status: active
dc1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500 inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255 ether 00:a0:cc:da:da:db media: Ethernet 10baseT/UTP status: no carrier
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
inet 127.0.0.1 netmask 0xff000000
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

Примечание

Старые версии FreeBSD могут потребовать запуска ifconfig с параметром -a, за более подробным описанием синтаксиса ifconfig обращайтесь к странице справочника. Учтите также, что строки, относящиеся к IPv6 (inet6 и т.п.) убраны из этого примера.

В этом примере были показаны следующие устройства:

- dc0: первый Ethernet интерфейс
- dc1: второй Ethernet интерфейс
- lp0: интерфейс параллельного порта
- lo0: устройство loopback
- tun0: туннельное устройство, используемое ppp

Для присвоения имени сетевой карте FreeBSD использует имя драйвера и порядковый номер, в котором карта обнаруживается при инициализации устройств. Например, sis2 это третья сетевая карта, использующая драйвер sis.

В этом примере, устройство dc0 включено и работает. Ключевые признаки таковы:

1. UP означает, что карта настроена и готова.
2. У карты есть интернет (inet) адрес (в данном случае 192.168.1.3).
3. Установлена маска подсети (netmask; 0xffffff00, то же, что и 255.255.255.0).
4. Широковещательный адрес (в данном случае, 192.168.1.255).
5. Значение MAC адреса карты (ether) 00:a0:cc:da:da:da
6. Выбор физической среды передачи данных в режиме автовыбора (media: Ethernet autoselect (100baseTX <full-duplex>)). Мы видим, что dc1 была настроена для работы с 10baseT/UTP. За более подробной информацией о доступных драйверу типах среды обращайтесь к странице справочника.
7. Статус соединения (status) active, т.е. несущая обнаружена. Для dc1, мы видим status: no carrier. Это нормально, когда Ethernet кабель не подключен к карте.

Если ifconfig показывает примерно следующее:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
ether 00:a0:cc:da:da:da
```

это означает, что карта не была настроена.

Для настройки карты вам потребуются привилегии пользователя root. Настройка сетевой карты может быть выполнена из командной строки с помощью ifconfig, но вам потребуется делать это после каждой перезагрузки системы. Подходящее место для настройки сетевых карт это файл /etc/rc.conf.

Откройте /etc/rc.conf в текстовом редакторе. Вам потребуется добавить строку для каждой сетевой карты, имеющейся в системе, например, в нашем случае, было добавлено две строки:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Замените dc0, dc1, и так далее на соответствующие имена ваших карт, подставьте соответствующие адреса. Обратитесь к страницам справочника сетевой карты и ifconfig, за подробной информацией о доступных опциях и к странице справочника rc.conf за дополнительной информацией о синтаксисе /etc/rc.conf.

Если вы настроили сетевую карту в процессе установки системы, некоторые строки, касающиеся сетевой карты, могут уже присутствовать. Внимательно проверьте /etc/rc.conf перед добавлением каких-либо строк.

Отредактируйте также файл /etc/hosts для добавления имен и IP адресов различных компьютеров сети, если их еще там нет. За дополнительной информацией обращайтесь к man.hosts.5; и к /usr/share/examples/etc/hosts.

1.3. Тестирование и решение проблем

Как только вы внесете необходимые изменения в /etc/rc.conf, перегрузите компьютер. Изменения настроек интерфейсов будут применены, кроме того будет проверена правильность настроек.

Как только система перезагрузится, проверьте сетевые интерфейсы.

1.3.1. Проверка Ethernet карты

Для проверки правильности настройки сетевой карты, попробуйте выполнить ping для самого интерфейса, а затем для другой машины в локальной сети.

Сначала проверьте локальный интерфейс:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms
--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0%
packet loss round-trip min/avg/max/stddev
=0.074/0.083/0.108/0.013 ms
```

Затем проверьте другую машину в локальной сети:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
```

```
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0%
packet loss round-trip min/avg/max/stddev =
                                    0.700/0.729/0.766
/0.025 ms
```

Вы можете также использовать имя машины вместо 192.168.1.2, если настроен файл /etc/hosts.

2. Настройка виртуальных серверов

Очень часто FreeBSD используется для размещения сайтов, когда один сервер работает в сети как несколько серверов. Это достигается присвоением нескольких сетевых адресов одному интерфейсу.

У сетевого интерфейса всегда есть один "настоящий" адрес, хотя он может иметь любое количество "синонимов" (alias). Эти синонимы обычно добавляются путём помещения соответствующих записей в /etc/rc.conf.

Синоним для интерфейса fxp0 выглядит следующим образом:

ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"

Заметьте, что записи синонимов должны начинаться с alias0 и идти далее в определенном порядке (например, _alias1, _alias2, и т.д.). Конфигурационный процесс остановится на первом по порядку отсутствующем числе.

Определение маски подсети для синонима очень важно, но к счастью, так же просто. Для каждого интерфейса должен быть один адрес с истинной маской подсети. Любой другой адрес в сети должен иметь маску подсети, состоящую из всех единичек (что выражается как 255.255.255.255 или как 0xffffffff).

Например, рассмотрим случай, когда интерфейс fxp0 подключён к двум сетям, к сети 10.1.1.0 с маской подсети 255.255.255.0 и к сети 202.0.75.16 с маской 255.255.255.240. Мы хотим, чтобы система была видна по IP, начиная с 10.1.1.1 по 10.1.1.5 и с 202.0.75.17 по 202.0.75.20. Как было сказано выше, только первый адрес в заданном диапазоне (в данном случае, 10.0.1.1 и 202.0.75.17) должен иметь реальную маску сети; все остальные (с 10.1.1.2 по 10.1.1.5 и с

202.0.75.18 по 202.0.75.20) должны быть сконфигурированы с маской сети 255.255.255.255.

Для этого в файл /etc/rc.conf должны быть внесены следующие записи:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

3. Автоматическая настройка сети (DHCP)

3.1. Что такое DHCP?

DHCP, или Dynamic Host Configuration Protocol (Протокол Динамической Конфигурации Хостов), описывает порядок, по которому система может подключиться к сети и получить необходимую информацию для работы в ней. Во FreeBSD используется dhclient , импортированный из OpenBSD 3.7. Вся информация здесь, относительно dhclient относится либо к ISC, либо к DHCP клиентам.DHCP сервер включён в ISC дистрибутив.

3.2. Как это работает

Когда на клиентской машине выполняется программа dhclient, являющаяся клиентом DHCP, она начинает широковещательную рассылку запросов на получение настроечной информации. По умолчанию эти запросы делаются на 68 порт UDP. Сервер отвечает на UDP 67, выдавая клиенту адрес IP и другую необходимую информацию, такую, как сетевую маску, маршрутизатор и серверы DNS. Вся эта информация даётся в форме "аренды" DHCP и верна только определенное время (что настраивается администратором сервера DHCP). При таком подходе устаревшие адреса IP тех клиентов, которые больше не подключены к сети, могут автоматически использоваться повторно.

Клиенты DHCP могут получить от сервера очень много информации. Подробный список находится в странице Справочника dhcp-options.

3.3. Интеграция с FreeBSD

DHCP клиент от OpenBSD, dhclient, полностью интегрирован во FreeBSD. Поддержка клиента DHCP есть как в программе установки, так и в самой системе, что исключает необходимость в знании подробностей конфигурации сети в любой сети, имеющей сервер DHCP. Утилита dhclient включена во все версии FreeBSD, начиная с 3.2.

DHCP поддерживается утилитой sysinstall. При настройке сетевого интерфейса из программы sysinstall второй вопрос, который вам задается: "Do you want to try DHCP configuration of the interface?" ("Хотите ли вы попробовать настроить этот интерфейс через DHCP?"). Утвердительный ответ приведёт к запуску программы dhclient, и при удачном его выполнении к автоматическому заданию информации для настройки интерфейса.

Есть две вещи, которые вы должны сделать для того, чтобы ваша система использовала DHCP при загрузке:

- Убедитесь, что устройство bpf включено в компиляцию вашего ядра. Чтобы это сделать, добавьте строчку device bpf в конфигурационный файл ядра и перестройте ядро.

Устройство bpf уже является частью ядра GENERIC, которое поставляется вместе с FreeBSD, так что, если вы не используете другое ядро, то вам и не нужно его делать для того, чтобы работал DHCP.

Примечание

Те, кто беспокоится о безопасности, должны иметь в виду, что устройство bpf является также тем самым устройством, которое позволяет работать программам- снифферам пакетов (хотя для этого они должны быть запущены пользователем root). Наличие устройства bpf необходимо для использования DHCP, но если вы чересчур беспокоитесь о безопасности, то вам нельзя добавлять устройство bpf в ядро только для того, чтобы в неопределённом будущем использовать DHCP.

• По умолчанию, конфигурирование FreeBSD по протоколу DHCP выполняется

фоновым процессом, или асинхронно. Остальные стартовые скрипты продолжают работу не ожидая завершения процесса конфигурирования, тем самым ускоряя загрузку системы.

Фоновое конфигурирование не создает проблем в случае, если сервер DHCP быстро отвечает на запросы, и процесс конфигурирования происходит быстро. Однако, в некоторых случаях настройка по DHCP может длиться значительное время. При этом запуск сетевых сервисов может потерпеть неудачу, если будет произведен ранее завершения конфигурирования по DHCP. Запуск DHCP в синхронном режиме предотвращает проблему, откладывая выполнение остальных стартовых скриптов до момента завершения конфигурирования по DHCP.

Для осуществления фонового конфигурирования по DHCP (асинхронный режим), используйте значение «DHCP» в /etc/rc.conf :

```
ifconfig_fxp0="DHCP"
```

Для откладывания запуска стартовых скриптов до завершения конфигурирования по DHCP (синхронный режим), укажите значение «SYNCDHCP »:

```
ifconfig_fxp0="SYNCDHCP"
```

Примечание

Обязательно замените fxp0 на имя интерфейса, который вы хотите настраивать динамически.

Если dhclient в вашей системе находится в другом месте или если вы хотите задать дополнительные параметры для dhclient, то также укажите следующее (изменив так, как вам нужно):

```
dhcp_program="/sbin/dhclient"
dhcp_flags=""
```

Сервер DHCP, dhcpd, включён как часть порта net/isc-dhcp3-server в коллекцию портов. Этот порт содержит DHCP-сервер от ISC и документацию.

3.4. Файлы

• `/etc/dhclient.conf`

dhclient требует наличия конфигурационного файла, /etc/dhclient.conf. Как правило, файл содержит только комментарии, а настройки по умолчанию достаточно хороши. Этот настроечный файл описан на страницах справочной системы по dhclient.conf.

- `/sbin/dhclient`

dhclient скомпонован статически и находится в каталоге /sbin. На страница Справочника dhclient дается более подробная информация о dhclient.

- `/sbin/dhclient-script`

dhclient-script является специфичным для FreeBSD скриптом настройки клиента DHCP. Он описан в dhclient-script, но для нормального функционирования никаких модификаций со стороны пользователя не требуется.

- `/var/db/dhclient.leases`

В этом файле клиент DHCP хранит базу данных выданных к использованию адресов в виде журнала. На странице dhclient.leases дается гораздо более подробное описание.

Полное описание протокола DHCP дается в RFC 2131 (http://www.freesoft.org/CIE/RFC/2131/). Кроме того, дополнительная информация есть на сервере http://www.dhcp.org/.

3.5. Установка и настройка сервера DHCP

Серверная часть пакета не поставляется как часть FreeBSD, так что вам потребуется установить порт net/isc-dhcp3-relay для получения этого сервиса.

3.5.1. Установка сервера DHCP

Для того, чтобы настроить систему FreeBSD на работу в качестве сервера DHCP, вам необходимо обеспечить присутствие устройства bpf, вкомпилированного в ядро. Для этого добавьте строку device bpf в файл конфигурации вашего ядра.

Устройство bpf уже входит в состав ядра GENERIC, поставляемого с FreeBSD, так что вам не нужно создавать собственное ядро для обеспечения работы DHCP.

Примечание

Те, кто обращает особое внимание на вопросы безопасности, должны заметить, что bpf является тем устройством, что позволяет нормально работать снифферам пакетов (хотя таким программам требуются привилегированный доступ). Наличие устройства bpf обязательно для использования DHCP, но если вы очень обеспокоены безопасностью, наверное, вам не нужно включать bpf в ваше ядро только потому, что в отдалённом будущем вы собираетесь использовать DHCP.

Следующим действием, которое вам нужно выполнить, является редактирование примерного dhcpd.conf, который устанавливается в составе порта net/isc-dhcp3-server. По умолчанию это файл /usr/local/etc/dhcpd.conf.sample, и вы должны скопировать его в файл /usr/local/etc/dhcpd.conf перед тем, как его редактировать.

3.5.2. Настройка сервера DHCP

dhcpd.conf состоит из деклараций относительно подсетей и хостов, и проще всего описывается на примере:

```
option    domain-name "example.com";
option domain-name-servers 192.168.4.100;
option    subnet-mask 255.255.255.0;
default-lease-time 3600;
max-lease-time 86400; ddns-update-style none;
subnet 192.168.4.0 netmask 255.255.255.0     {
range 192.168.4.129  192.168.4.254;
option routers 192.168.4.1;
}
host mailhost {
hardware ethernet 02:03:04:05:06:07;
fixed-address  mailhost.example.com;
}
```

1. Этот параметр задаёт домен, который будет выдаваться клиентам в качестве домена, используемого по умолчанию при поиске. Обратитесь к страницам справочной системы по resolv.conf для получения дополнительной информации о том, что это значит.

2. Этот параметр задаёт список разделённых запятыми серверов DNS, которые должен использовать клиент.

3. Маска сети, которая будет выдаваться клиентам.

4. Клиент может запросить определённое время, которое будет действовать выданная информация. В противном случае сервер выдаст настройки с этим сроком (в секундах).

5. Это максимальное время, на которое сервер будет выдавать конфигурацию. Если клиент запросит больший срок, он будет подтверждён, но будет действовать только max-lease-time секунд.

6. Этот параметр задаёт, будет ли сервер DHCP пытаться обновить DNS при выдаче или освобождении конфигурационной информации. В реализации ISC этот параметр является обязательным.

7. Это определение того, какие IP-адреса должны использоваться в качестве резерва для выдачи клиентам. IP-адреса между и включая границы, будут выдаваться клиентам.

8. Объявление маршрутизатора, используемого по умолчанию, который будет выдаваться клиентам.

9. Аппаратный MAC-адрес хоста (чтобы сервер DHCP мог распознать хост, когда тот делает запрос).

10. Определение того, что хосту всегда будет выдаваться один и тот же IP-адрес. Заметьте, что указание здесь имени хоста корректно, так как сервер DHCP будет разрешать имя хоста самостоятельно до того, как выдать конфигурационную информацию.

Когда вы закончите составлять свой dhcpd.conf, нужно разрешить запуск сервера DHCP в файле /etc/rc.conf, добавив в него строки

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Замените dc0 именем интерфейса (или именами интерфейсов, разделяя их пробелами), на котором(ых) сервер DHCP должен принимать запросы от клиентов.

Затем вы можете стартовать сервер DHCP при помощи команды

```
# /usr/local/etc/rc.d/isc-dhcpd.sh start
```

Если в будущем вам понадобится сделать изменения в настройке вашего сервера, то важно заметить, что посылка сигнала SIGHUP приложению dhcpd не приведёт к перезагрузке настроек, как это бывает для большинства даемонов. Вам нужно послать сигнал

SIGTERM для остановки процесса, а затем перезапустить его при помощи вышеприведённой команды.

3.5.3. Файлы

- /usr/local/sbin/dhcpd

dhcpd скомпонован статически и расположен в каталоге /usr/local/sbin. Страницы справочной системы dhcpd, устанавливаемые портом, содержат более полную информацию о dhcpd.

- /usr/local/etc/dhcpd.conf

dhcpd требует наличия конфигурационного файла, /usr/local/etc/dhcpd.conf, до того, как он будет запущен и начнёт предоставлять сервис клиентам. Необходимо, чтобы этот файл содержал все данные, которая будет выдаваться обслуживаемым клиентам, а также информацию о работе сервера. Этот конфигурационный файл описывается на страницах справочной системы dhcpd.conf, которые устанавливаются портом.

- /var/db/dhcpd.leases

Сервер DHCP ведёт базу данных выданной информации в этом файле, который записывается в виде протокола. Страницы справочной системы dhcpd.leases, устанавливаемые портом, дают гораздо более подробное описание.

- /usr/local/sbin/dhcrelay

dhcrelay используется в сложных ситуациях, когда сервер DHCP пересылает запросы от клиента другому серверу DHCP в отдельной сети. Если вам нужна такая функциональность, то установите порт net/isc-dhcp3-server. На страницах справочной системы dhcrelay, которые устанавливаются портом, даётся более полное описание.

Прокси сервер

1. Использование прокси-сервера

Прокси-сервер (от англ. proxy — «представитель, уполномоченный») — служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-

серверу и запрашивает какойлибо ресурс (например, файл), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кэша (в случаях, если прокси имеет свой кеш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях.

Из многочисленных значений английского слова proxy в данном контексте применимы такие: «доверенное лицо», «полномочный представитель». То есть некто, кто действует от вашего имени по вашему поручению вместо вас. В компьютерах прокси – это программа, которая передает запросы ваших программ (браузеров и других) в интернет, получает ответы и передает их обратно. Необходимость в такой программе возникает обычно, если с пользовательского компьютера невозможно работать в интернете непосредственно напрямую из-за того, что у него нет прямого подключения к интернету (модема, например), но есть на другом компьютере в его сети. Тогда на этом другом компьютере ставят программу прокси, а все остальные компьютеры в локальной сети настраивают таким образом, чтобы работа велась через прокси. Сейчас через прокси умеют работать практически все популярные интернет-программы. Это значит что все пользователи локальной сети могут получить полноценный доступ в интернет, если хотя бы один из них этот доступ уже имеет.

Чаще всего прокси-серверы применяются для следующих целей:

- Обеспечение доступа с компьютеров локальной сети в Интернет.
- Кэширование данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию

на прокси-сервере и выдавать по запросу, снижая тем самым нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации.

- Сжатие данных: прокси-сервер загружает информацию из Интернета и передаёт информацию конечному пользователю в сжатом виде. Такие прокси-серверы используются в основном с целью экономии внешнего трафика.
- Защита локальной сети от внешнего доступа: например, можно настроить проксисервер так, что локальные компьютеры будут обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они «видят» только прокси-сервер). См. также NAT.
- Ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определённым веб-сайтам, ограничить использование интернета каким-то локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и вирусы.
- Анонимность доступа к различным ресурсам. Прокси-сервер может скрывать сведения об источнике запроса или пользователе. В таком случае целевой сервер видит лишь информацию о прокси-сервере, например, IP-адрес, но не имеет возможности определить истинный источник запроса. Существуют также искажающие прокси-серверы, которые передают целевому серверу ложную информацию об истинном пользователе.

2. Работа компьютеров локальной сети в Интернете без прокси

Такая работа требует выполнения определенных дополнительных условий и имеет свои минусы в сравнении с работой через прокси. Какие условия: каждому компьютеру

должен быть выдан персональный IP-адрес в сети Интернет, и построена схема маршрутизации так, что пакеты будут попадать именно на этот компьютер. Это невозможно сделать без участия провайдера. Провайдеры идут на это, но, как правило, за отдельную плату. Например, $5 за каждый IP в месяц. Это хлопотно, дорого и снижает уровень безопасности в целом: каждый компьютер вашей сети станет потенциальной мишенью хакеров, вирусных атак и прочих «прелестей» Интернета. При правильной настройке компьютеров это не очень страшно, но рядовые пользователи не склонны следить за безопасностью своих компьютеров, значит, будут дополнительные хлопоты у администраторов сети. Возможностей контролировать работу пользователей у администратора будет немного, так как система децентрализована. Кстати, при таком способе подключения тоже нужна программа-посредник на том компьютере, который непосредственно подключен к интернету. Но при наличии реальных IP-адресов эта программа – обычный роутер (маршрутизатор) IP-пакетов, он является частью операционной системы. И к этой программе название «прокси» не применяют. Важное отличие маршрутизатора от прокси – при использовании маршрутизатора IP-пакеты остаются без изменений, в них сохраняются исходные адреса компьютеров ЛС. А прокси всегда работает от своего адреса, а адреса клиентов не передаются, т.к. недоступны из интернета. Маршрутизатор, меняющий адреса, уже является прокси (его называют NAT-proxy, NAT (Network Address Translation).

3. Виды прокси

Упомянутый выше NAT-proxy – самый простой вид прокси. Теперь он даже входит в состав Windows 2000 и Windows XP. Там он называется «Общий доступ к подключению интернета» и

включается галочкой в свойствах модемного соединения. Этот прокси работает прозрачно для пользователя, никаких специальных настроек в программах не требуется. Но на этом удобства этого прокси заканчиваются. Влиять на работу «общего доступа» Windows (например, ограничивать список доступных сайтов для отдельных пользователей) вы не сможете. Другие NAT-proxy могут быть более гибкими, но их общая проблема – универсальность. Они не «вникают» в тонкости тех прикладных протоколов, которые через себя пропускают, поэтому и не имеют средств управления ими.

Специализированные прокси (для каждого протокола интернета свой вид прокси) имеют ряд преимуществ и с точки зрения администраторов, и с точки зрения пользователей. Ниже перечислены виды специализированных прокси.

3.1. HTTP-прокси

HTTP-прокси – самый распространенный. Он предназначен для организации работы браузеров и других программ, использующих протокол HTTP. Браузер передает проксисерверу URL ресурса, прокси-сервер получает его с запрашиваемого веб-сервера (или с другого прокси-сервера) и отдает браузеру. У HTTP-прокси широкие возможности при выполнении запросов:

- Можно сохранять полученные файлы на диске сервера. Впоследствии, если запрашиваемый файл уже скачивался, то можно выдать его с диска без обращения в интернет – увеличивается скорость и экономится внешний трафик (который может быть платным). Эта опция называется кэшированием – именно её очень любят администраторы и пользователи – настолько, что считают её главной функцией прокси. Однако приводимые оценки экономии (в описаниях встречалось от 30 до 60%) слишком оптимистичны, не верьте им. На деле получается не более 10-15% – современный интернет очень динамичен, страницы часто меняются, зависят от работающего с ними

пользователя и т.д. – такие данные кэшировать нельзя, веб-серверы обычно вставляют в HTTP-заголовки специальные указания об этом, чтобы браузеры и прокси имели это в виду. Хотя многие прокси-серверы можно настроить так, чтобы эти указания частично игнорировались – например, перечитывать страницу не чаще одного раза в день.

- Можно ограничивать доступ к ресурсам. Например, завести «черный список» сайтов, на которые прокси не будет пускать пользователей (или определенную часть пользователей, или в определенное время и т.д.). Ограничения можно реализовать по-разному. Можно просто не выдавать ресурс – например, выдавая вместо него страницу «запрещено администратором» или «не найдено». Можно спрашивать пароль и авторизованных пользователей допускать к просмотру. Можно, не спрашивая пароля, принимать решение на основании адреса или имени компьютера пользователя.
- Можно выдавать не тот ресурс, который запрашивается браузером. Например, вместо рекламных баннеров и счетчиков показывать пользователям прозрачные картинки, не нарушающие дизайн сайта, но существенно экономящие время и трафик за счет исключения загрузки картинок извне.
- Можно ограничивать скорость работы для отдельных пользователей, групп или ресурсов. Например, установить правило, чтобы файлы *.mp3 качались на скорости не более 1кб/сек, чтобы предотвратить забивание вашего интернетканала трафиком меломанов, но не лишать их полностью этого удовольствия. Эта возможность, к сожалению, есть не во всех прокси.
- Ведутся журналы работы прокси – можно подсчитывать трафик за заданный период, по заданному пользователю, выяснять популярность тех или иных ресурсов и т.д.
- Можно маршрутизировать веб-запросы – например, часть направлять напрямую, часть через другие прокси (прокси

провайдера, спутниковые прокси и т.д.). Это тоже помогает эффективнее управлять стоимостью трафика и скоростью работы прокси в целом.

3.2. FTP-прокси

FTP-прокси бывает двух основных видов в зависимости от протокола работы самого прокси. С ftp-серверами этот прокси, конечно, всегда работает по протоколу FTP. А вот с клиентскими программами – браузерами и ftp-клиентами (CuteFTP, FAR, и др.) прокси может работать как по FTP, так и по HTTP. Второй способ удобнее для браузеров, т.к. исторически является для них «родным». Браузер запрашивает ресурс у прокси, указывая протокол целевого сервера в URL – http или ftp. В зависимости от этого прокси выбирает протокол работы с целевым сервером, а протокол работы с браузером не меняется – HTTP. Поэтому, как правило, функцию работы с FTP-серверами также вставляют в HTTPпрокси, т.е. HTTP-прокси, описанный выше, обычно с одинаковым успехом работает как с HTTP, так и с FTP-серверами. Но при «конвертации» протоколов FTP<->HTTP теряется часть полезных функций протокола FTP. Поэтому специализированные ftpклиенты предпочитают и специальный прокси, работающий с обеими сторонами по FTP. В Eserv и Eproxy мы называем этот прокси FTP-gate, чтобы подчеркнуть отличие от FTPпрокси внутри HTTP-прокси. Также этот прокси называется в некоторых ftp-клиентах. Хотя встречаются и вносящие путаницу названия. Например, в программе CuteFTP FTPgate называют firewall, хотя FireWall в общем случае – это вообще не прокси, а фактически программа обратного назначения – не для подключения к интернету, а для изоляции от него ;) Для прокси в FireWall оставляют специальные «дыры». FTP-gate поддерживают различные способы указания в FTP-протоколе целевого сервера, с которым FTP-клиент хочет работать, в настройке FTP-клиентов обычно предлагается выбор

этого способа, например, USER user@site, OPEN site, и т.д. – способ указания сервера, с которым производится работа. Такое многообразие связано с тем, что нет общепринятого стандарта на этот вид прокси, и применяются такие хитрые добавки к стандартным командам FTP-протокола.

3.3. HTTPS-прокси

HTTPS-прокси – фактически часть HTTP-прокси. S в названии означает «secure», т.е. безопасный. Не смотря на то, что программно эту часть HTTP-прокси, обычно HTTPS выделяют в отдельную категорию (и есть отдельное поле для него в настройке браузеров). Обычно этот протокол – безопасный HTTP – применяют, когда требуется передача секретной информации, например, номеров кредитных карт. При использовании обычного HTTP-прокси всю передаваемую информацию можно перехватить средствами самого прокси (т.е. это под силу администратору ЛС) или на более низком уровне, например, tcpdump (т.е. и администратор провайдера и любого промежуточного узла и вообще любой человек, имеющий физический доступ к маршрутам передачи ваших данных по сети, может при большом желании узнать ваши секреты). Поэтому в таких случаях применяют secure HTTP – всё передаваемое при этом шифруется. Прокси-серверу при этом дается только команда «соединится с таким-то сервером», и после соединения прокси передает в обе стороны шифрованный трафик, не имея возможности узнать подробности (соответственно и многие средства управления доступом – такие как фильтрация картинок – не могут быть реализованы для HTTPS, т.к. прокси в этом случае неизвестно, что именно передается). Собственно в процессе шифрации/дешифрации прокси тоже участия не принимает – это делают клиентская программа и целевой сервер. Наличие команды «соединиться с таким-то сервером» в HTTPS-прокси приводит к интересному и

полезному побочному эффекту, которым все чаще пользуются разработчики клиентских программ. Так как после соединения с указанным сервером HTTPS-прокси лишь пассивно передает данные в обе стороны, не производя никакой обработки этого потока вплоть до отключения клиента или сервера, это позволяет использовать прокси для передачи почти любого TCP-протокола, а не только HTTP. То есть HTTPS-прокси одновременно является и простым POP3-прокси, SMTP-прокси, IMAP-прокси, NNTP-прокси и т.д. – при условии, что соответствующая клиентская программа умеет так эксплуатировать HTTPS-прокси (увы, далеко не все еще это умеют, но есть вспомогательные программы, «заворачивающие» трафик обычных клиентов через HTTPS-прокси). Никаких модификаций целевого сервера не требуется. Фактически HTTPS-прокси является программируемым mapping-proxy, как и Socks-proxy.

3.4. Mapping-прокси

Mapping-прокси – способ заставить работать через прокси те программы, которые умеют работать с интернетом только напрямую. При настройке такого прокси администратор создает как бы «копию» целевого сервера, но доступную через один из портов проксисервера для всех клиентов локальной сети – устанавливает локальное «отображение» заданного сервера. Например, пользователи локальной сети хотят работать с почтовым сервером mail.ru не через браузер, а с использованием почтовой программы Outlook Express или TheBat. Эти программы не умеют работать через прокси (кроме случая, когда Outlook получает почту по HTTP с hotmail.com – тогда он, как и браузер, пользуется HTTP-прокси). Простейший способ работать с mail.ru по POP3 через прокси – установить локальное отображение сервера pop.mail.ru. И в Outlook'ах вместо pop.mail.ru написать имя прокси-сервера и порт отображения. Outlook будет соединяться с прокси-сервером («думая», что это

почтовый сервер), а прокси при этом будет соединяться с pop.mail.ru и прозрачно передавать всю информацию между Outlook и pop.mail.ru, таким образом «превращаясь» на время соединения в POP3-сервер. Неудобство mapping-прокси в том, что для каждого необходимого внешнего сервера нужно вручную устанавливать отдельный порт на прокси. Но зато не требуется модификация ни серверов, ни клиентов. Особенно это помогает в случае необходимости «проксирования» многочисленных

«доморощенных» протоколов, реализованных в играх или финансовых программах. Почему-то они часто игнорируют существование прокси и стандартных протоколов. Такие программы можно «обмануть» и направить через прокси практически всегда, если они не делают другой глупости – передачи клиентского IP-адреса внутри протокола и пытаются с ним соединяться напрямую еще раз (что невозможно, т.к. локальные адреса недоступны извне).

3.5. Socks-прокси

Socks-прокси. SOCKS5 – протокол для прокси-сервера, позволяющий пропускать через прокси почти любой прикладной TCP- или UDP-протокол. SOCKS4 – старая версия протокола, имеет ряд ограничений – в частности, не поддерживается передача имени хоста вместо IP-адреса.

4. Кэш

В интернете неизбежны перегрузки и «заторы», что объясняется наличием следующих проблем: низкие скорости соединения; непредсказуемые технические характеристики; ограничения по полосе частот; Web-сайты, перегруженные заказами. Один из способов решения этих проблем связан с использованием устройств кэширования (от слова cache - тайник, склад, запас).

Средства кэширования включают в состав Web-браузеров, что позволяет запоминать некоторые Web-страницы, к которым обращался пользователь компьютера, для их последующего повторного использования. Точно такой же принцип заложен и в любое устройство Web-кэширования: Web-контент перемещается в некий сетевой кэш поближе к пользователям, нуждающимся в нем, вследствие чего уменьшается число участков маршрутизации или коммутации, через которые он должен пройти. В случае корпоративных пользователей наиболее близкое расположение такого контента - в самой корпоративной сети. Сетевые кэши работают на тех же принципах, что и браузеры, однако они выбирают контент, анализируя активность сотен и тысяч пользователей, а не одного, как в случае с браузером.

RFC 2187 описывает протокол ICP (Internet Cache Protocol), который позволяет осуществлять иерархическое соединение кэшей. Он определяет порядок обмена информацией между кэшами, находящимися в состоянии подчинения.

ICP прежде всего используется в иерархии кэшей для поиска определенных объектов в братских кэшах. Если squid не находит нужного документа, то посылает ICP запрос братским кэшам, которые в свою очередь отвечают ICP ответами «HIT» («попадание») или «MISS» («промах»). Затем кэш использует ответы для выбора, при помощи какого кэша разрешать свои ответы MISS.

5. Прокси и кэш

Кэш-серверы и прокси-серверы - не одно и то же. Кэширование по-прежнему остается одной из функций прокси-серверов. Однако повышение спроса на специализированное кэширование приводит к тому, что кэш-серверы все чаще выпускаются в качестве отдельных продуктов. Так, продукт CacheQube компании Cobalt Networks представляет собой

устройство, которое просто устанавливается между локальной сетью и маршрутизатором для осуществления прозрачного кэширования. Streaming Media Cache фирмы Inktomi и MediaMail производства InfoLibria представляют собой кэши, специально предназначенные для обработки потоковых аудио и видео.

6. Прокси-кэш

Кэш-серверы изучают активность, перехватывая запросы одним из двух способов: путем прозрачного кэширования или прокси-кэширования. Прозрачный кэш-сервер «просеивает» через себя весь проходящий трафик и поэтому не требует модификации установок конечного клиента. Он устанавливается обычно перед маршрутизатором, соединенным с интернетом.

В случае прокси-кэша сетевые администраторы конфигурируют пользовательские браузеры так, чтобы они направляли запросы на контент непосредственно в кэш. Затем прокси-кэш-сервер запрашивает нужный контент от имени пользователя. Это позволяет сетевым администраторам также ограничить число сайтов, на которые могут заходить пользователи. Такой подход более сложен, поскольку требует конфигурирования каждого клиента. Кроме того, если в этом случае прокси-кэш-сервер выйдет из строя, пользователи не смогут обращаться к Web.

Таким образом, прокси-кэш - некое средство в прокси-сервере, которое кэширует поступающие Web-страницы на жестком диске. Если страница, запрашиваемая браузером, уже находится в прокси-кэше, то она отыскивается в нем, а не в интернете. Так случилось, что прокси-кэш-серверами называют практически все устройства кэширования, независимо от их расположения относительно потока информации, а «прозрачное» кэширование стало лишь одним из режимов работы прокси-кэш-сервера.

Существует несколько подходов к реализации архитектуры прокси-кэша, которые принято называть моделями. Выбор той или иной модели определяется размещением прокси-кэша, его главным назначением и природой трафика. Помимо прозрачного кэширования, существуют еще следующие архитектуры (модели) прокси-кэш-сервера:

- Прямой прокси-кэш. При такой конфигурации запросы пользователей на своем пути к Web-серверу проходят через кэш. Если кэш содержит запрашиваемый документ, этот документ отправляется пользователю. В противном случае сервер работает как прокси, извлекая нужный контент из Web-сервера.
- Обратный прокси-кэш, или «серверный ускоритель». Кэш может быть также сконфигурирован как быстрый Web-сервер для ускорения более медленных традиционных Web-серверов. При этом документы, хранящиеся в кэше, обрабатываются с высокой скоростью, в то время как документы, не занесенные в кэш (обычно динамический контент) запрашиваются при необходимость из исходных Web-серверов. Такая кэширующая система располагается перед одним или несколькими Web-серверами, перехватывая запросы и действуя наподобие прокси. Эти прокси-кэш-серверы могут размещаться по всей сети, формируя некую распределенную сеть сайтов для хостирования контента. Дополнительное достоинство данной схемы связано с возможностью балансировки нагрузки и динамического зеркалирования.

7. Продукты

В настоящее время все прокси-кэш-серверы, представленные на рынке, могут быть классифицированы следующим образом:

- Прокси-кэш-серверы со специализированной ОС (CacheFlow, Network Alliance и Cisco Systems). В таких серверах само ядро ОС разработано с учетом требований, предъявляемых к прокси-кэш-серверам реальными условиями их использования

- продолжительные потоки данных, файлы большого размера и длительные сеансы связи.

- Изделия, представляющие собой специализированные программно-аппаратные средства со стандартной ОС (Cobalt Networks).
- Стандартные серверы, несколько оптимизированные для использования в качестве Web-серверов, на которых установлено ПО кэширования. К этой категории относятся прокси-кэш-серверы компании Compaq с ПО Novell Internet Caching System и корпорации Intel с ПО Inktomi Traffic Server Engine. Такие серверы используются обычно в локальных сетях средних размеров. По итогам испытаний, проведенных IRCache (см. «Сетевой журнал», № 1/2000), лучшим был признан продукт именно этого класса, который состоял из аппаратного комплекса Dell и ПО Novell.

CacheFlow специализируется непосредственно на системах кэширования, для чего ею была разработана собственная ОС CachOS, оптимизирующая функции кэширования. В продуктах этой фирмы реализована технология Object Pipelining, которая позволяет организовать быстрый доступ к контенту с первого раза, ликвидируя значительную часть задержек на пути от Web-браузера клиента до удаленного Web-сервера провайдера. Возникновение таких задержек связано с тем, что Web-страницы, как правило, состоят из множества объектов и для каждого такого объекта обычно должна сначала открываться TCP-сессия, после чего уже следует получение HTTP-запроса. Вместо последовательного получения объектов Object Pipelining сразу открывает такое количество TCP-сессий, какое удаленный сервер может позволить, и получает объекты параллельно. После этого объекты доставляются от устройства прямо пользователю настолько быстро, насколько браузер пользователя может их запрашивать.

Также реализован алгоритм адаптивного обновления (Adaptive Refresh), предназначенный для ускорения обработки повторных запросов. Так как содержание Web-серверов постоянно меняется, прокси-кэш-сервер должен содержать временное хранилище контента в актуальном состоянии. В традиционном решении для того, чтобы гарантированно доставить пользователю актуальные данные, прокси-кэш-сервер должен обязательно произвести проверку контента на исходном сервере. С другой стороны, чтобы доставить данные быстро, сервер не должен ждать, пока пользователь запросит контент, перед тем как сервер обновит страницы. Если обновление контента происходит только в тот момент, когда пользователь посылает запрос, последний сталкивается со значительными задержками. Единственный метод доставки Web-страниц быстро и адекватно - это производить освежение контента асинхронно с запросами клиентов. Запатентованный алгоритм адаптивного обновления селективно обновляет содержимое кэша в зависимости от потребности. Для каждого объекта, хранящегося в кэше, формируются две модели - модель изменений и модель использования, из комбинаций которых формируются рациональные сценарии обновления. Эти сценарии динамически изменяются в зависимости от изменения модели.

Компания утверждает, что уменьшение используемой полосы достигает 60%, сокращение времени доставки контента пользователю (или провайдеру) или времени отклика сети доходит до десятикратного, разгрузка Web-серверов составляет до 70% запросов.

Еще одна особенность этих продуктов - хорошая совместимость с сетевым оборудованием Cisco, о которой говорят многие системные интеграторы в России.

Network Appliance - крупнейший поставщик NAS (Network Attached Storage), и, естественно, ее коньком являются технологии предоставления быстрого, надежного доступа по

сети к большим объемам данных. Подходы, отработанные при проектировании устройств доступа, фирма перенесла и на свои прокси-кэш-серверы:

- отказоустойчивая высокопроизводительная архитектура (поддержка RAID, fibre channel, горячая замена блоков питания и вентиляторов);
- оптимизированная под RAID4 файловая система WAFL плюс кэширование запросов на запись в памяти типа NVRAM.
- Продукты используют собственную ОС (NetCache) и поддерживают ICAP -- протокол, по которому прокси-кэш-сервер передает принятые данные на сторонний сервер, осуществляющий их мониторинг (например, дополнительную проверку на вирусы) и возвращающий обратно. Имеется поддержка takeover, когда два сервера работают в паре; если один вышел из строя, пользователи продолжают прозрачно работать через второй. Предусмотрена аутентификация пользователей через Radius/LDAP/NTLM, имеется поддержка streaming media.

Cisco Systems. В продукции применена ОС Cisco IOS, оптимизированная для организации телекоммуникаций, а также технология Cisco Network Caching, которая минимизирует избыточный трафик, передаваемый по каналам WAN. Она позволяет повысить производительность сети за счет того, что большинство запросов к внешним ресурсам исполняется локально, а не за счет передачи этих запросов к удаленным серверным группам. Такое решение защищает внутреннюю сеть от неконтролируемых перегрузок в интернете или в корпоративной сети, что позволяет повысить качество предоставляемых услуг и доступность информации, хранящейся на внешних серверах.

Архитектурно данная технология оптимизирована для использования одной кэширующей платформы, объединяющей в себе поддержку протокола WCCP на всех критичных

устройствах активного сетевого оборудования. WCCP (Web Cache Control Protocol) - протокол, разработанный компанией Cisco Systems и ставший фактически стандартом для прозрачных кэширующих систем; он поддерживается практически всеми поставщиками устройств кэширования.

Cobalt Networks. Эта компания поставляет недорогие кэширующие продукты для предприятий малого и среднего бизнеса. В сентябре 2000 года она была приобретена компанией Sun Microsystems (www.sun.ru), поэтому новые продукты поставляются под маркой Sun Cobalt. Компания утверждает, что сокращение полосы частот, обеспечиваемое этими продуктами, достигает 50%.

Compaq Computer. В представленных в обзоре продуктах Compaq, образующих линейку для потребителей разных классов, используется ПО кэширования Novell ICS Caching. Однако Compaq Computer Corporation и Inktomi 21 марта 2001 года подписали соглашение, в соответствии с которым Compaq становится партнером Inktomi в разработке интегрированных платформ сетевой доставки контента (компания Inktomi специализируется на разработке ПО для масштабируемых сетевых инфраструктур для интернета). Целью указанного соглашения является разработка программных платформ и программно-аппаратных серверов для доставки контента. В рамках этой программы предполагается объединить кэш-серверы Compaq TaskSmart C-Series с ПО Inktomi Traffic Server и Inktomi MediaBridge. Такое изделие должно появиться на рынке летом 2001 года.

По утверждению компании используемая полоса уменьшается на величину до 30%, сокращение времени доставки контента пользователю (или провайдеру) или времени отклика сети достигает десятикратного, Разгрузка Web-серверов составляет до 80% запросов.

Intel. Для изделий этой корпорации характерна высокая отказоустойчивость. Имеется порт аварийного управления Emergency Management Port (EMP), позволяющий управлять устройством даже при отказе программного обеспечения или сети. Предусмотрено несколько вариантов прозрачного режима, включая коммутацию четвертого уровня, WPAD и маршрутизацию на основе правил. Применяется архитектура DataFlow, которая обеспечивает одновременную буферизацию и передачу потоков данных для повышения пропускной способности и бесперебойного ввода/вывода.

8. Факторы, на которые необходимо обратить внимание при покупке

Технические характеристики. Одна из основных характеристик прокси-кэш-сервера - емкость дисковых массивов, используемых для кэширования, и эта характеристика представляется наиболее объективной и легко проверяемой. Что касается остальных характеристик, а именно экономии полосы пропускания, ускорения обращения к контенту и разгрузки Web-серверов, то они обычно оказываются трудноизмеримыми на практике, во всяком случае, такую проверку невозможно сделать при приобретении изделия, и к этим показателям, декларируемым производителями и продавцами, следует относиться скептически. Однако чем больше емкость и быстродействие дискового массива, тем, как правило, выше и остальные показатели.

Простота установки и использования. Наиболее предпочтительными с этой точки зрения являются приборы типа solution in a box, которые включают все необходимые аппаратные и программные средства и относятся к категории plug-and-play. Это, например, прокси-кэш-серверы CacheFlow и Intel.

Возможность изменения оптимизируемой характеристики. Основное предназначение прокси-кэш-сервера - экономия полосы пропускания и сокращение времени реакции на запросы пользователя. Опыт показывает, что достичь наибольшего выигрыша одновременно по этим двум показателям не удается и надо выбрать что-то одно, причем выбор того или иного варианта определяется конкретной задачей пользователя. По этой причине весьма желательно иметь возможность переключать режимы работы прокси-кэшсервера: максимальная экономия полосы пропускания и максимальное сокращение времени реакции. Например, в прокси-кэш-сервере CacheFlow просто имеется соответствующий тумблер.

Гибкость конфигурирования. С течением времени у пользователя может возникнуть необходимость в переконфигурировании прокси-кэш-сервера. В этом случае желательно, чтобы он был способен работать в соответствии с моделями прямого, обратного и прозрачного кэширования.

Администрирование плохо сконструированного прокси-кэш-сервера отнимает массу времени у системных администраторов, особенно если в корпорации установлен кластер таких серверов. Желательно, чтобы все прокси-кэш-серверы могли администрироваться централизованно и при этом имели бы интерфейсы на основе браузера и командной строки.

Масштабируемость и надежность. Высокая масштабируемость достигается в том случае, если возможна кластеризация прокси-кэш-серверов или построение иерархических кластерных систем. Базовым показателем надежности является коэффициент готовности, на который надо обращать особое внимание при покупке.

Размеры, масса, энергопотребление. Прокси-кэш-серверы могут существенно различаться по этим показателям, поэтому при их выборе следует учитывать возможности,

предоставляемые тем рабочим местом, куда кэш-сервер предполагается установить.

9. Обновление информации в кэше

Современные кэши используют пассивное или активное кэширование. При пассивном кэшировании кэш-сервер проверяет свежесть контента. Обычно кэш-сервер посылает команду get (в HTTP) для запрашивания объекта от контент-сервера. В тех случаях, когда объект уже был сохранен, кэш-сервер использует модифицированную команду get if, в соответствии с которой объект скачивается, если он был изменен после последнего запроса. Затем кэш-сервер сравнивает даты изменения объекта, поступившего от сервера, и объекта, хранящегося в кэше, и направляет пользователю самый последний вариант.

Слабое место пассивного кэширования - производительность: пользователи должны ждать, пока кэш-сервер проверит каждый запрос. Однако в этом случае возможны некоторые улучшения, например, если производится проверка свежести данных по дате окончание заданного срока, заносимой в заголовок объекта. Когда объект достигает определенного «возраста», кэш-сервер запрашивает свежий контент.

При активном кэшировании улучшение характеристик достигается с использованием эвристических методов оценки срока жизни объекта. Сервер при этом проводит вычисления, используя такие данные, как дата занесения объекта в кэш, продолжительность его пребывания в кэше, IP-адрес источника и множество подобных сведений. При таком подходе не нужно проверять каждый запрос. Вместо этого кэшсервер делает определенные предположения о времени жизни объекта, скажем два дня. В течение этого интервала все запросы объекта немедленно обслуживаются из кэша, однако по истечении этого срока кэш обновляет объект.

10. Размещение прокси-кэш-серверов

Различают логическое и физическое размещение прокси-кэш-сервера. В случае корпоративной сети используют три основные логические конфигурации:

- сервер помещается рядом с маршрутизатором (в этом случае он обрабатывает трафик по протоколу управления Web-кэшированием, WCCP);
- сервер объединяется с коммутатором четвертого или более высокого уровня (в этом случае он управляет трафиком);
- сервер встраивается в коммутатор второго или третьего уровня.

Физически прокси-кэш-сервер необходимо размещать как можно ближе к пользователям. Для корпоративной сети это означает, что кэширование должно осуществляться на границе сети.

Что касается сотрудников корпорации, работающих на дому, то кэширование нужной им информации выполняют прокси-кэш-серверы, размещенные в точке присутствия ISP.

Иногда функции кэша совмещают с функциями межсетевого экрана, что позволяет сэкономить некоторые средства. Однако тогда поддержка аппаратного и программного обеспечения оказывается относительно сложной.

11. Прокси-сервер Squid

11.1. Подготовка к установка SQUID.

В первую очередь нам необходимо скачать прокси сервер. В данном примере мы разместим исходники сервера в разделе /root/distr. Так, скорее всего, его в вашей системе нет, то следующим блоком команд мы его создадим и загрузим туда SQUID.

Сперва необходима скачать дистрибутив из интернета и скинуть его на флешку. Затем монтируем флешку:

mount –t msdosfs /dev/da0s1 /mnt

Затем создадим директорию, куда будем копировать:

freebsd# mkdir -p /root/distr

Теперь скопируем архив в нашу папку:

freebsd# cp /mnt/squid-2.6.STABLE7-20070120.tar /root/distr

После скачивания распаковываем архив:

freebsd# tar xvfz squid-2.6.STABLE7-20070120.tar.gz freebsd# cd squid-2.6.STABLE7-20070120.tar

Теперь все готово к компиляции и установке!

11.2. Компиляция и установка прокси сервера SQUID

Скомпилировать и установить довольно просто. Выполняем следующий набор команд:

 # ./configure --prefix=/usr/local/squid

 # make all

 # make install

Если команда configure выдаст ошибку из за отсутствия в системе Perl вы можете легко это исправить добавив его в систему следующей командой:

 # pkg_add -r perl

Или же установив дистрибутив, скачав его из интернета и скопировав его на ваш компьютер с флешки. Чтобы установить дистрибутив, необходимо разархивировать его и выполнить следующие команды:

sh Configure -de

make

make test

make install

После добавления Perl повторите набор команд конфигурации и установки.

11.3. Настройка SQUID

Файл конфигурации лежит в следующей директории /usr/local/squid/squid.conf

Открываем его любым редактором и первое что необходимо задать - "visible_hostname". Например:

visible_hostname freebsd

Замените freebsd на любое имя хоста и сохраните файл (необходимо, чтобы имя хоста совпадало с именем вашего компьютера, на котором вы устанавливаете SQUID).

Следующее что мы должны сделать для запуска squid это создать раздел в который будем сохранять логи и где squid будет хранить cache. И конечно настроить доступ и разрешения к этим разделам. Выполним следующий набор команд:

```
# mkdir -p /usr/local/squid/var/logs/
# chmod 777 /usr/local/squid/var/logs/

# mkdir -p /usr/local/squid/var/cache/
# chmod 777 /usr/local/squid/var/cache/
```

И обязательно необходимо что бы squid создал структуру разделов для хранения cache перед его первым запуском. Выполнить эту команду обазятельно:

```
# /usr/local/squid/sbin/squid -z
```

Теперь squid готов к первому запуску!

11.4. Старт и остановка прокси сервера

Для того что бы запустить squid достаточно выполнить следующую команду:

 # /usr/local/squid/sbin/squid

Остановить squid можно так:

 # kill -9 `cat /usr/local/squid/var/logs/squid.pid`

Можно настроить запуск SQUID при старте системы создав следующий файл

/usr/local/etc/rc.d/squid.sh с минимальным содержанием:

#!/bin/sh
/usr/local/squid/sbin/squid

Обязательно надо разрешить выполнять данный файл:

 # chmod 755 /usr/local/rc.d/squid.sh

11.5. Конфигурационный файл

visible_hostname anyhost
http_port 8080
icp_port 3130
cache_peer 10.5.2.24 parent 8080 3130 proxy-only
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

dns_nameservers 10.5.2.24
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours

refresh_pattern ^ftp: 1440 20% 10080

refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern . 0 20% 4320

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT acl
get_post method GET POST

acl net2 src 10.5.2.0/255.255.255.0
acl net219 src 10.5.219.0/255.255.255.0
acl net220 src 10.5.220.0/255.255.255.0
acl net224 src 10.5.224.0/255.255.255.0
acl net226 src 10.5.226.0/255.255.255.0
acl net231 src 10.5.231.0/255.255.255.0
acl net157 src 10.5.157.0/255.255.255.0
acl net158 src 10.5.158.0/255.255.255.0

http_access allow net2
http_access allow net219
http_access allow net220

http_access allow net224
http_access allow net157

http_access allow net158
http_access allow net226
http_access deny net231
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_reply_access allow all
icp_access
allow all uri_whitespace encode
nonhierarchical_direct off
coredump_dir /usr/local/squid/var/cache
request_entities on

Domain Name System (DNS)

1. **Обзор**

По умолчанию во FreeBSD используется одна из версий программы BIND (Berkeley Internet Name Domain), являющейся самой распространенной реализацией протокола DNS. DNS - это протокол, при помощи которого имена преобразуются в IP-адреса и наоборот. Например, в ответ на запрос о www.FreeBSD.org будет получен IP-адрес веб-сервера Проекта FreeBSD, а запрос о ftp.FreeBSD.org возвратит IP-адрес соответствующей машины с FTP-сервером. Точно также происходит и обратный процесс. Запрос, содержащий IP-адрес машины, возвратит имя хоста. Для выполнения запросов к DNS вовсе не обязательно иметь в системе работающий сервер имён.

FreeBSD в настоящее время поставляется с сервером DNS BIND9, предоставляющим расширенные настройки

безопасности, новую схему расположения файлов конфигурации и автоматические настройки для chroot.

В сети Интернете DNS управляется через достаточно сложную систему авторизированных корневых серверов имён, серверов доменов первого уровня (Top Level Domain, TLD) и других менее крупных серверов имён, которые содержат и кэшируют информацию о конкретных доменах.

На данный момент пакет BIND поддерживается Internet Software Consortium http://www.isc.org/.

2. Используемая терминология

Для изучения данного материала необходимо понимать значения некоторых терминов, связанных с работой DNS.

Таблица 2.1. Основные термины

Термин	Определение
Прямой запрос к DNS (forward DNS)	Преобразование имён хостов в адреса IP
Ориджин (origin)	Обозначает домен, покрываемый конкретным файлом зоны
named, bind, сервер имён	Общеупотребительные названия для обозначения пакета BIND, обеспечивающего работу сервера имён во FreeBSD.
Резолвер	Системный процесс, посредством которого машина обращается к серверу имён для получения информации о зоне
Обратный DNS (reverse DNS)	Операция, обратная прямому запросу к DNS; преобразование адресов IP в имена хостов
Корневая зона	Начало иерархии зон Интернет. Все зоны находятся под корневой зоной, подобно тому, как все файлы располагаются ниже корневого каталога.
Зона	Отдельный домен, поддомен или часть

	DNS, управляемая одним сервером.

Примеры зон:

- . является корневой зоной
- *org.* — домен верхнего уровня (TLD) в корневой зоне.
- *example.org.* является зоной в домене верхнего уровня (TLD) org..
- *1.168.192.in-addr.arpa* является зоной, в которую включены все IP-адреса, формирующие пространство адресов 192.168.1.*.

Как можно видеть, уточняющая часть имени хоста появляется слева. Например, example.org. более точен, чем org., также, как org. более точен, чем корневая зона. Расположение каждой части имени хоста сильно похоже на файловую систему: каталог /dev расположен в корневой файловой системе, и так далее.

3. **Файл настройки DNS-клиента**

Файл настройки ясно документирует настройки DNS-клиента. Администратор может указать до трех серверов имен, два из которых являются резервными - на случай, если не ответит первый сервер. Кроме того, файл содержит имя домена по умолчанию и прочие параметры работы. Файл *resolv.conf* - важнейшая часть настройки службы имен.

/etc/resolv.conf - это простой файл, подходящий для чтения людьми. Существуют некоторые вариации команд файла, зависящие от системы. Ниже перечислены записи, поддерживаемые большинством систем:

nameserver адрес

Записи nameserver указывают IP-адреса серверов, опрашиваемых DNS-клиентом на предмет получения доменной информации. Опрос серверов имен происходит в порядке следования записей *nameserver*. Если от сервера не получен ответ, DNS-клиент посылает запрос следующему серверу по

списку, и так до тех пор, пока не будет достигнут последний из серверов. Если файл *resolv.conf* не существует либо в файле отсутствуют записи nameserver, все запросы направляются локальному узлу. Однако если файл *resolv.conf* существует и содержит записи nameserver, обращение к локальному узлу происходит только в том случае, если присутствует соответствующая запись nameserver. Указывайте официальный IP-адрес локального узла (или адрес 0.0.0.0), но не кольцевой адрес. Официальный адрес позволяет избежать проблем, возникающих в некоторых вариантах Unix при использовании кольцевого адреса. Вариант *resolv.conf* для чистого клиента DNS никогда не содержит записи *namerserver*, указывающей на локальный узел.

domain имя

Запись *domain* определяет доменное имя по умолчанию. DNS-клиент добавляет доменное имя по умолчанию к любому имени узла, не содержащему точки. Дополненное таким образом имя узла используется в запросе к серверу имен. Например, если DNS-клиент получает имя *crab* (не содержащее точки), он добавляет доменное имя по умолчанию в процессе конструирования запроса. Предположим, значение имени в записи domain - *wrotethebook.com*, тогда DNS-клиент создает запрос для *crab.wrotethebook.com*. Переменная среды LOCALDOMAIN, будучи установленной, имеет приоритет более

высокий, чем запись *domain*, и значение переменной используется для дополнения имени узла.

search домен...

Запись *search* определяет ряд доменов, в которых производится поиск, если имя узла не содержит точки. Предположим, файл содержит запись *search essex.wrotethebook.com butler.wrotethebook.com*. Запрос для узла по имени cookbook будет преобразован сначала в запрос для имени cookbook.essex.wrotethebook.com. Если поиск для такого

имени не принес положительных результатов, DNS-клиент создает запрос для *cookbook.butler.wrotethebook.com*. Вновь получив отрицательный результат, клиент DNS прервет процесс поиска для имени узла. Используйте запись *search* либо запись domain. Предпочтение отдавайте команде *search*. Никогда не используйте обе команды одновременно. Переменная среды LOCALDOMAIN имеет более высокий приоритет, чем запись *search*.

sortlist сеть [/маска сети] ...

Адреса, принадлежащие перечисленным в команде *sortlist* сетям, являются для DNS- клиента предпочтительными. Если DNS-клиент получает несколько адресов в ответ на запрос по многосетевому узлу или маршрутизатору, адреса сортируются таким образом, что адреса сетей *sortlist* предшествуют адресам всех прочих сетей. В ином случае адреса возвращаются приложению в порядке их получения от сервера имен.

Команда *sortlist* используется редко, поскольку затрудняет работу таких серверных механизмов, как распределение нагрузки. Основным исключением является ситуация, когда список сортировки настраивается для предпочтения адресов локальной сети всем прочим адресам. В последнем случае, если клиент DNS состоит в сети 172.16.0.0/16 и один из адресов, полученных в многоадресном ответе, также принадлежит этой сети, адрес сети 172.16.0.0 будет предшествовать всем прочим адресам.

options параметр ...

Запись *options* позволяет устанавливать необязательные параметры настройки клиента DNS. Доступны следующие параметры:

- *debug* – включает отладку - печать отладочных сообщений на стандартный вывод/ debug работает только в случае, если библиотека DNS-клиента была собрана с ключом - DDEBUG (в большинстве случаев это не так).

• *ndots:/i* – устанавливает число точек в имени узла, наличие которого служит критерием необходимости использования списка поиска перед отправкой запроса серверу имен. По умолчанию имеет значение 1. Таким образом, к имени узла с одной точкой не добавляется доменное имя перед отправкой серверу имен. Если указать параметр ndots:2, к имени узла с одной точкой добавляется домен из списка поиска перед отправкой запроса, но не к имени с двумя или более точками. Параметр *ndots* может пригодиться, если одну из составляющих имени домена можно спутать с доменом высшего уровня, и пользователи постоянно усекают имена из этого домена. В таком случае запросы будут передаваться для разрешения прежде всего корневым серверам имен для поиска в домене верхнего

уровня, прежде чем, в конечном итоге, вернуться на локальный сервер имен. Беспокоить корневые серверы по пустякам - очень плохой тон. Используйте *ndots*, чтобы обязать DNS-клиент принудительно дополнять проблемные имена локальным доменным именем, чтобы разрешение происходило без обращения к корневым серверам.

• *timeout:n* – устанавливает начальный интервал ожидания ответа DNS-клиентом. По умолчанию интервал ожидания равен 5 секундам для первого запроса к каждому серверу. В пакете BIND для системы Solaris 8 данный параметр имеет синтаксис *retrans: n.*

• *attempts: n* – задает число повторных попыток получить ответ на запрос. По умолчанию имеет значение 2, то есть DNS-клиент дважды повторяет попытку получить ответ для каждого из серверов в списке серверов, прежде чем вернуть приложению сообщение об ошибке. В пакете BIND для системы Solaris 8 данный параметр имеет синтаксис *retry: n* и значение по умолчанию 4.

• *rotate* – включает циклический механизм «round-robin» выбора серверов имен. В обычной ситуации DNS-клиент

посылает запрос первому серверу из списка, а следующему серверу - лишь в том случае, если первый сервер не ответил на запрос. Параметр *rotate* предписывает DNS-клиенту распределить нагрузку поровну между всеми серверами имен.

• *no-check-names* – отключает проверку доменных имен на соответствие документу RFC 952, DOD Internet Host Table Specification (Спецификация таблицы узлов сети Интернет Министерства обороны). По умолчанию доменные имена, содержащие подчеркивание (_), символы не из таблицы ASCII либо управляющие символы ASCII, считаются ошибочными. Воспользуйтесь этим параметром, если существует необходимость работать с именами, содержащими подчеркивание.

• *inet6* – предписывает DNS-клиенту создавать запросы адресов IPv6.

Чаще всего файл настройки *resolv.conf* содержит в списке поиска локальное доменное имя, указывает локальный узел в качестве первого сервера имен, а также один или два резервных сервера имен. Пример такой настройки:

```
#    Файл настройки клиента DNS
#
search wrotethebook.com
#    обратись, прежде всего, к себе nameserver 172.16.12.2
#    затем к узлу crab
nameserver 172.16.12.1
#    и, наконец, к узлу ora nameserver 172.16.1.2
```

Пример основан на воображаемой сети, поэтому по умолчанию для доменного имени указано имя *wrotethebook.com*. Файл взят с узла *rodent*, который и обозначен в качестве первого сервера имен. В качестве резервных серверов выступают *crab* и *ora*. Настройка не содержит параметров и списка сортировки, поскольку они применяются нечасто. Так выглядит файл настройки обычного DNS-клиента.

3.1. **Настройка чистого DNS-клиента**

Настройки чистого DNS-клиента очень просты. Они идентичны настройкам обычного клиента, но не указывают локальную систему в качестве сервера имен. Вот пример файла *resolv.conf* для системы чистого DNS-клиента:

 # Файл настройки DNS-клиента # search wrotethebook.com # обратись к узлу crab nameserver 172.16.12.1 # затем к узлу ora nameserver 172.16.1.2

Данные настройки предписывают DNS-клиенту передавать все запросы узлу *crab*, а если *crab* не ответил – узлу *ora*. Ни при каких обстоятельствах запросы не разрешаются локально. Столь простой файл настройки - все, что требуется для работы чистого DNS- клиента.

4. **Причины, по которым вам может понадобиться сервер имён**

Сервера имён обычно используются в двух видах: авторитетный сервер имён и кэширующий сервер имён.

Авторитетный сервер имён нужен, когда:

- нужно предоставлять информацию о DNS остальному миру, отвечая на запросы авторизированно.
- зарегистрирован домен, такой, как *example.org* и в этом домене требуется поставить имена машин в соответствие с их адресами IP.
- блоку адресов IP требуется обратные записи DNS (IP в имена хостов).
- резервный *(slave)* сервер имён должен отвечать на запросы.

Кэширующий сервер имён нужен, когда:

- локальный сервер DNS может кэшировать информацию и отвечать на запросы быстрее, чем это происходит при прямом опросе внешнего сервера имён.

Например, когда кто-нибудь запрашивает информацию о www.FreeBSD.org, то обычно резолвер обращается к серверу имён вашего провайдера, посылает запрос и ожидает ответа. С локальным кэширующим сервером DNS запрос во внешний мир будет делаться всего один раз. Каждый дополнительный запрос не будет посылаться за пределы локальной сети, потому что информация уже имеется в кэше.

5. **Как это работает**

Во FreeBSD даемон BIND, по очевидным причинам, называется named.

Таблица 2.2. Даемон BIND

Файл	Описание
named	Даемон BIND
rndc	Программа управления даемоном сервера имён
/etc/namedb	Каталог, в котором
Файл	Описание
	информация о зонах BIND
/etc/namedb/named.conf	Конфигурационный файл для

Файлы зон обычно располагаются в каталоге */etc/namedb* и содержат информацию о зоне DNS, за которую отвечает сервер имён.

В зависимости от способа конфигурации зоны на сервере файлы зон могут располагаться в подкаталогах *master, slave* или *dynamic* иерархии */etc/namedb*. Эти файлы содержат DNS

информацию, которую и будет сообщать в ответ на запросы сервер имен.

6. **Запуск BIND**

Так как сервер имён BIND устанавливается по умолчанию, его настройка сравнительно проста.

Стандартная конфигурация *named* запускает простой кэширующий сервер в ограниченной среде *chroot*. Для одноразового запуска даемона в этой конфигурации используйте команду

 # /etc/rc.d/named forcestart

Чтобы даемон named запускался во время загрузки, поместите в */etc/rc.conf* следующую строку:

namedenable="YES"

Разумеется, существует множество различных конфигураций */etc/namedb/named.conf*, лежащих за рамками данного документа. Разнообразные опции запуска named во FreeBSD описаны в переменных *named_** файла */etc/defaults/rc.conf* и странице справочника *rc.conf*.

1.7. Конфигурационные файлы

Файлы конфигурации даемона *named* расположены в каталоге */etc/namedb* и, за исключением случая, когда вам требуется просто резолвер, требуют модификации.

7.1. **Использование make-localhost**

Для создания основной зоны для локального хоста перейдите в каталог */etc/namedb* и выполните команду

 # sh make-localhost

В каталоге master должны появиться файлы *localhost.rev* для локальной адресной зоны и localhost-v6.rev для для конфигурации IPv6. Ссылки на эти файлы уже содержатся в файле конфигурации *named.conf*.

/etc/namedb/named.conf

```
// $FreeBSD$
//
// If you are going to set up an authoritative server, make sure you
// understand the hairy details of how DNS works. Even with
// simple mistakes, you can break connectivity for affected parties,
// or cause huge amounts of useless Internet traffic.
options {
// All file and path names are relative to the chroot directory,
// if any, and should be fully qualified.
directory "/etc/namedb";
pid-file "/var/run/named/pid";
dump-file "/var/dump/named_dump.db";
statistics-file "/var/stats/named.stats";
// If named is being used only as a local resolver, this is a safe default.
// For named to be accessible to the network, comment this option, specify
// the proper IP address, or delete this option.
listen-on { 127.0.0.1; };
// If you have IPv6 enabled on this system, uncomment this option for
// use as a local resolver. To give access to the network, specify
// an IPv6 address, or the keyword "any".
// listen-on-v6 { ::1; };
// These zones are already covered by the empty zones listed below.
// If you remove the related empty zones below, comment these lines out.
disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
disable-empty-zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.O
```

IP6.ARPA";

disable-empty-zone

"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.O IP6.ARPA";

// If you've got a DNS server around at your upstream provider, enter

// its IP address here, and enable the line below. This will make you

// benefit from its cache, thus reduce overall DNS traffic in the Internet.

/

forwarders {

127.0.0.1;

};

**/*

// If the 'forwarders' clause is not empty the default is to 'forward first'

// which will fall back to sending a query from your local server if the name

// servers in 'forwarders' do not have the answer. Alternatively you can

// force your name server to never initiate queries of its own by enabling

the

// following line:

// forward only;

// If you wish to have forwarding configured automatically based on

// the entries in /etc/resolv.conf, uncomment the following line and

// set named_auto_forward=yes in /etc/rc.conf. You can also enable

// named_auto_forward_only (the effect of which is described above).

// include "/etc/namedb/auto_forward.conf";

Как и говорится в комментариях, если вы хотите получить эффект от использования кэша провайдера, то можно включить раздел *forwarders*. В обычном случае сервер имён будет рекурсивно опрашивать определённые серверы имён Интернет до тех пор, пока не получит ответ на свой запрос. При включении этого раздела он будет автоматически опрашивать сервер имён вашего провайдера (или тот, который здесь указан), используя преимущества его кэша. наличия нужной информации. Если соответствующий сервер имён провайдера работает быстро и имеет хороший канал связи, то в результате такой настройки вы можете получить хороший результат.

Предупреждение

127.0.0.1 здесь работать не будет. Измените его на IP-адрес сервера имён провайдера.

*/**

Modern versions of BIND use a random UDP port for each outgoing query by default in order to dramatically reduce the possibility of cache poisoning. All users are strongly encouraged to utilize this feature, and to configure their firewalls to accommodate it. AS A LAST RESORT in order to get around a restrictive firewall policy you can try enabling the option below. Use of this option will significantly reduce your ability to withstand cache poisoning attacks, and should be avoided if at all possible.

*Replace NNNNN in the example with a number between 49160 and 65530. */ // query-source address * port NNNNN;*

};

// If you enable a local name server, don't forget to enter 127.0.0.1 // first in your /etc/resolv.conf so this server will be queried.

// Also, make sure to enable it in /etc/rc.conf.

// The traditional root hints mechanism. Use this, OR the slave zones below. zone "." { type hint; file "/etc/namedb/named.root"; };

/ Slaving the following zones from the root name servers has some significant advantages:*

1. *Faster local resolution for your users*
2. *No spurious traffic will be sent from your network to the roots*
3. *Greater resilience to any potential root server failure/DDoS*

On the other hand, this method requires more monitoring than the hints file to be sure that an unexpected failure mode has not incapacitated your server. Name servers that are serving a lot of clients will benefit more from this approach than individual hosts. Use with caution.

To use this mechanism, uncomment the entries below, and comment the hint zone above.

*As documented at http://dns.icann.org/services/axfr/ these zones: "." (the root), ARPA, IN-ADDR.ARPA, IP6.ARPA, and ROOT-SERVERS.NET are available for AXFR from these servers on IPv4 and IPv6: xfr.lax.dns.icann.org, xfr.cjr.dns.icann.org */ /* zone "." {*

```
type slave;
file "/etc/namedb/slave/root.slave";
masters { 192.5.5.241;      // F.ROOT-SERVERS.NET.
};
notify no;
};
zone type file          "arpa" {
slave;
"/etc/namedb/slave/arpa.slave";
masters { 192.5.5.241;      // F.ROOT-SERVERS.NET.
};
notify no;
};
*/
```

/ Serving the following zones locally will prevent any queries for these zones leaving your network and going to the root*

name servers. This has two significant advantages:

1. Faster local resolution for your users

2. No spurious traffic will be sent from your network to the roots

**/*

// RFCs 1912 and 5735 (and BCP 32 for localhost)

zone "localhost" { type master; file "/etc/namedb/master/localhostforward.

db"; };

zone "127.in-addr.arpa" { type master; file "/etc/namedb/master/

localhost-reverse.db"; };

zone "255.in-addr.arpa" { type master; file "/etc/namedb/master/

empty.db"; };

// RFC 1912-style zone for IPv6 localhost address

zone "0.ip6.arpa" { type master; file "/etc/namedb/master/localhostreverse. db"; };

// "This" Network (RFCs 1912 and 5735)

zone "0.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Private Use Networks (RFCs 1918 and 5735)

zone "10.in-addr.arpa" { type master; file "/etc/namedb/master/

empty.db"; };

zone "16.172.in-addr.arpa" empty.db"; };

zone "17.172.in-addr.arpa" empty.db"; };

zone "18.172.in-addr.arpa" empty.db"; };

zone "19.172.in-addr.arpa" empty.db"; };

zone "20.172.in-addr.arpa" empty.db"; };

zone "21.172.in-addr.arpa" empty.db"; };

zone "22.172.in-addr.arpa" empty.db"; };

zone "23.172.in-addr.arpa" empty.db"; };
zone "24.172.in-addr.arpa" empty.db"; };
zone "25.172.in-addr.arpa" empty.db"; };
zone "26.172.in-addr.arpa" empty.db"; };
zone "27.172.in-addr.arpa" empty.db"; };
zone "28.172.in-addr.arpa"
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master; file "/etc/namedb/master/
{ type master;
file "/etc/namedb/master/

empty.db"; };
zone "29.172.in-addr.arpa" { type
empty.db"; };
zone "30.172.in-addr.arpa" { type
empty.db"; };
zone "31.172.in-addr.arpa" { type
empty.db"; };
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/

```
5735)
master; file "/etc/namedb/master/
5735 and 5736)
master; file "/etc/namedb/master/
empty.db"; };
// TEST-NET-[1-3] for Documentation (RFCs 5735 and 5737)
zone "2.0.192.in-addr.arpa" { type master; file
"/etc/namedb/master/
empty.db"; };
zone "100.51.198.in-addr.arpa" { type master; file "/etc/namedb/
master/empty.db"; };
zone "113.0.203.in-addr.arpa" { type master; file "/etc/namedb/
master/empty.db"; };
// IPv6 Range for Documentation (RFC 3849)
zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file "/etc/namedb/
master/empty.db"; };
// Domain Names for Documentation and Testing (BCP 32)
zone "test" { type master; file "/etc/namedb/master/empty.db"; };
zone "example" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file
    "/etc/namedb/master/empty.db"
zone "example.net" { type master; file
    "/etc/namedb/master/empty.db"
zone "example.org"{ type master; file
    "/etc/namedb/master/empty.db"
// Router Benchmark Testing (RFCs 2544 and 5735)
zone "18.198.in-addr.arpa" { type master;
    file "/etc/namedb/master/
empty.db"; };
```

```
zone "19.198.in-addr.arpa" { type master;
    file "/etc/namedb/master/
empty.db"; };
// IANA Reserved - Old Class E zone "240.in-addr.arpa" {
    type
empty.db"; };
zone "241.in-addr.arpa" { type
empty.db"; };
zone "242.in-addr.arpa" { type
empty.db"; };
zone "243.in-addr.arpa" { type
empty.db"; };
zone "244.in-addr.arpa" { type
empty.db"; };
zone "245.in-addr.arpa" { type
empty.db"; };
zone "246.in-addr.arpa" { type
empty.db"; };
zone "247.in-addr.arpa" { type
empty.db"; };
zone "248.in-addr.arpa" { type
empty.db"; };
zone "249.in-addr.arpa" { type
empty.db"; };
zone "250.in-addr.arpa" { type
empty.db"; };
zone "251.in-addr.arpa" { type
Space (RFC 5735)
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
master; file "/etc/namedb/master/
```

```
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/
master;     file     "/etc/namedb/master/

empty.db"; };
zone        "252.in-addr.arpa"   {       type    master;         file
    "/etc/namedb/master/
empty.db"; };
zone        "253.in-addr.arpa"   {       type    master;         file
    "/etc/namedb/master/
empty.db"; };
zone        "254.in-addr.arpa"   {       type    master;         file
    "/etc/namedb/master/
empty.db"; };
// IPv6 Unassigned Addresses (RFC 4291)
zone        "1.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "3.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "4.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "5.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "6.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "7.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
zone        "8.ip6.arpa"   {       type    master;         file
    "/etc/namedb/master/empty.db";      };
```

```
zone "9.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "a.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "b.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "c.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "d.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "e.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "0.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "1.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "2.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "3.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "4.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "5.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "6.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "7.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "8.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "9.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
```

```
zone "a.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "b.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "0.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "1.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "2.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "3.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "4.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "5.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "6.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "7.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "d.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.0
db"; };
// IPv6 Link Local (RFC 4291)
zone "8.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "9.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
zone "a.e.f.ip6.arpa" { type master; file
    "/etc/namedb/master/empty.db"; };
```

zone "b.e.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Deprecated Site-Local Addresses (RFC 3879)

zone "c.e.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

zone "d.e.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

zone "e.e.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

zone "f.e.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IP6.INT is Deprecated (RFC 4159)

zone "ip6.int" { type master; file "/etc/namedb/master/empty.db"; };

// NB: Do not use the IP addresses below, they are faked, and only

// serve demonstration/documentation purposes!

//

// Example slave zone config entries. It can be convenient to become

// a slave at least for the zone your own domain is in. Ask

// your network administrator for the IP address of the responsible

// master name server.

//

// Do not forget to include the reverse lookup zone!

// This is named after the first bytes of the IP address, in reverse

// order, with ".IN-ADDR.ARPA" appended, or ".IP6.ARPA" for IPv6.

//

// Before starting to set up a master zone, make sure you fully

```
// understand how DNS and BIND work. There are sometimes
// non-obvious pitfalls. Setting up a slave zone is usually simpler.
//
// NB: Don't blindly enable the examples below. :-) Use actual names
// and addresses instead.
/* An example dynamic zone
key "exampleorgkey" { algorithm hmac-md5;
secret "sf87HJqjkqh8ac87a02lla==";
};
zone "example.org" {
type master;
allow-update { key "exampleorgkey";
};
file "dynamic/example.org";
};
*/
/* Example of a slave reverse zone
zone "1.168.192.in-addr.arpa" {
type slave;
file "/etc/namedb/slave/1.168.192.in-addr.arpa";
masters {
192.168.1.1;
};
};
*/
```

Это примеры описаний прямой и обратной зон из файла *named.conf* для вторичных серверов.

Для каждой новой зоны, которую будет обслуживать сервер имён, в файл *named.conf* должна быть добавлена запись.

К примеру, самая простая запись для домена *example.org* может выглядеть вот так:

zone "example.org" {
type master;
file "master/example.org";
};

Зона является первичной, что отражается в поле *type*, и информация о зоне хранится в файле */etc/namedb/master/example.org*, что указывается в поле *file*.

zone "example.org" { type slave; file "slave/example.org";
};

В случае вторичной зоны информация о ней передается с основного сервера имён для заданной зоны и сохраняется в указанном файле. Если и когда основной сервер имён выходит и строя или недосягаем, то скачанная информация о зоне будет находиться на вторичных серверах, и они смогут обслуживать эту зону.

7.2. **Файлы зон**

Пример файла зоны example.org для основного сервера (располагающийся в файле */etc/namedb/master/example.org*) имеет такой вид:

```
$TTL 3600;      1 hour
example.org.    IN    SOA    ns1.example.org.
admin.example.org. (    2006051501
                        1           ;Serial
                        10800       ;Refresh
                        3600        ;Retry
                        604800      ;Expire
                        300         ;Negative Response TTL

); DNS Servers
      IN    NS    ns1.example.org.
      IN    NS    ns2.example.org.
; MX Records
```

IN MX 10 mx.example.org.
IN MX 20 mail.example.org
IN A 192.168.1.1
; Machine Names
localhost IN A 127.0.0.1
ns1 IN A 192.168.1.2
ns2 IN A 192.168.1.3
mx IN A 192.168.1.4
mail IN A 192.168.1.5
; Aliases
www IN CNAME example.org.

Заметьте, что все имена хостов, оканчивающиеся на «.», задают полное имя, тогда как все имена без символа «.» на конце считаются заданными относительно *origin*. Например, ns1 преобразуется в *ns1.example.org.* Файл зоны имеет следующий формат:

recordname IN recordtype value

Наиболее часто используемые записи DNS:

SOA

начало зоны ответственности

NS

авторитативный сервер имен

A

адрес хоста

CNAME

каноническое имя для алиаса

MX

обмен почтой

PTR

указатель на доменное имя (используется в обратных зонах DNS)

example.org. IN SOA ns1.example.org. admin.example.org. (
2006051501 ; Serial
10800 ; Refresh after 3 hours
3600 ; Retry after 1 hour
604800 ; Expire after 1 week
300) ; Minimum TTL of 1 day
example.org.

имя домена, а так же ориджин для этого файла зоны.

ns1.example.org.

основной/авторитативный сервер имён для этой зоны.

admin.example.org.

человек, отвечающий за эту зону, адрес электронной почты с символом "@" замененным на точку. *(<admin@example.org>* становится admin.*example.org*)

2006051501

последовательный номер файла. При каждом изменении файла зоны это число должно увеличиваться. В настоящее время для нумерации многие администраторы предпочитают формат ггггммддвв. 2006051501 будет означать, что файл последний раз изменялся 15.05.2006, а последнее число 01 означает, что это была первая модификация файла за

день. Последовательный номер важен, так как он служит для того, чтобы вторичные серверы узнавали об обновлении зоны.

IN NS ns1.example.org.

Это NS-запись. Такие записи должны иметься для всех серверов имён, которые будут отвечать за зону.

localhost IN A 127.0.0.1
ns1 IN A 192.168.1.2
ns2 IN A 192.168.1.3
mx IN A 192.168.1.4
mail IN A 192.168.1.5

Записи типа A служат для обозначения имён машин. Как это видно выше, имя ns1.example.org будет преобразовано в 192.168.1.2.

IN A 192.168.1.1

Эта строка присваивает IP адрес 192.168.1.1 текущему ориджину, в данном случае домену *example.org*.

www IN CNAME @

Записи с каноническими именами обычно используются для присвоения машинам псевдонимов. В этом примере www является псевдонимом для "главной" машины, соответствующей ориджину, то есть example.org (192.168.1.1). Записи CNAME могут использоваться для присвоения псевдонимов именам хостов или для использования одного имени несколькими машинами по очереди.

IN MX 10 mail.example.org

MX-запись указывает, какие почтовые серверы отвечают за обработку входящей электронной почты для зоны. *mail.example.org* является именем почтового сервера, а 10 обозначает приоритет этого почтового сервера.

Можно иметь несколько почтовых серверов с приоритетами, например, 10, 20 и так далее. Почтовый сервер, пытающийся доставить почту для *example.org*, сначала попробует связаться с машиной, имеющий MX-запись с самым большим приоритетом (наименьшим числовым значением в поле MX), затем с приоритетом поменьше и так далее, до тех пор, пока почта не будет отправлена.

Для файлов зон *in-addr.arpa* (обратные записи DNS) используется тот же самый формат, отличающийся только использованием записей PTR вместо A или CNAME.

$TTL 3600

1.168.192.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (

```
                2006051501      ; Serial
                 10800          ; Refresh
                 3600           ; Retry
                604800          ; Expire

                300     )       ; Negative Response TTL

        IN      NS      ns1.example.org.
        IN      NS      ns2.example.org.
1       IN      PTR     example.org.
2       IN      PTR     ns1.example.org.
3       IN      PTR     ns2.example.org.
4       IN      PTR     mx.example.org.
5       IN      PTR     mail.example.org.
```

В этом файле дается полное соответствие имён хостов IP-адресам в нашем описанном ранее вымышленном домене.

8. **Кэширующий сервер имён**

Кэширующий сервер имён - это сервер имён, не отвечающий ни за какую зону. Он просто выполняет запросы от своего имени и сохраняет результаты для последующего использования. Для настройки такого сервера достаточно исключить все описания зон из стандартной конфигурации сервера имён.

9. **Безопасность**

Хотя BIND является самой распространенной реализацией DNS, всегда стоит вопрос об обеспечении безопасности. Время от времени обнаруживаются возможные и реальные бреши в безопасности.

FreeBSD автоматически запускает *named* в ограниченном окружении (*chroot*); помимо этого, есть еще несколько механизмов, помогающих защититься от возможных атак на сервис DNS.

Весьма полезно прочесть сообщения безопасности CERT (*http://www.cert.org/*) и подписаться на Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью (*http://lists.FreeBSD.org/mailman/listinfo/freebsd-security- notifications*) для того, чтобы быть в курсе текущих проблем с обеспечением безопасности Internet и FreeBSD.

Подсказка

Если возникают проблемы, то наличие последних исходных текстов и свежеоткомпилированного *named* не помешает.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Для выполнения лабораторных работ ознакомиться с предложенным материалом по установке и базовой настройке ОС FreeBSD произвести ее установку и настройку.
2. Ознакомиться с предложенным материалом для получения информации о настройке прокси-сервера Squid под ОС FreeBSD
3. Сконфигурировать дистрибутив и установить прокси-сервер Squid
4. Произвести настройку ПО (сервера настраиваются каскадом)
5. Проверить работоспособность.
6. Ознакомиться с предложенным материалом для получения базовой информации о DNS в ОС FreeBSD
7. Настроить DNS-клиент (резолвер), файл resolv.conf
8. Настроить кэширующий DNS-сервер (BIND)
9. Настроить зоны прямого и обратного отображения для учебной сети FreeBSD
10. Проверить работоспособность DNS-клиента и DNS-сервера
11. Подготовить ответы на контрольные вопросы

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

.

Отчет на защиту предоставляется в печатном или электронном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), этапы выполнения работы (со скриншотами), результаты выполнения работы (скриншоты и содержимое файлов), выводы.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Что такое сетевой интерфейс?
2. Что такое виртуальный сервер?
3. Какая утилита позволяет читать и изменять настройки сетевых интерфейсов?
4. Что такое «синоним» (alias) сетевого интерфейса?
5. Что такое DHCP?
6. Как настраивать DHCP сервер?
7. Как настраивать DHCP клиент?
8. Что такое прокси-сервер?
9. В каких целях применяются прокси-серверы?
10. Какие преимущества в работе в сети Интернет с прокси?
11. Какие существуют виду прокси?
12. Что такое HTTP-прокси? Какими возможностями обладает?
13. Что такое FTP-прокси?
14. Что такое HTPS-прокси? В чем отличие от HTTP-прокси?
15. Что такое Maping-прокси?
16. Что такое Socks-прокси?
17. Что такое кэширование?
18. Что такое сетевые кэши? Для чего они нужны?
19. Что такое ICP? HTCP?
20. В чем отличие кэш-сервера от прокси-сервера?
21. Зачем нужен кэш-сервер?
22. Что такое прокси-кэш-сервер? Как это работает?
23. Что значит прозрачное кэширование?

24. Какие существуют архитектуры (модели) прокси-кэш-сервера? В чем их суть?
25. Как можно классифицировать прокси-кэш-серверы, представленные на рынке?
26. Какие фирмы занимаются разработкой и производством прокси-кэш-серверов? Чем характеризуется их продукция?
27. На что стоит обратить внимание при покупке прокси-кэш-сервера?
28. Что такое активное и пассивное кэширование? В чем отличие?
29. Каким образом размещают прокси-кэш-серверы?
30. Что такое каскадная настройка прокси-серверов? Объясните каскадный режим работы прокси-сервера.
31. Что такое DNS?
32. По каким причинам может понадобиться сервер имен?
33. Какая программа в ОС FreeBSD отвечает за работу системы DNS?
34. Как запустить BIND? Что это такое?
35. Какой файл используется для настройки DNS клиента?
36. Что такое зона в понятии DNS? Типы зон.
37. Когда используется кэширующий сервер имен?
38. Какая программа используется для управления сервером имен?

ЛАБОРАТОРНАЯ РАБОТА №3
НАСТРОЙКА МАРШРУТИЗАЦИИ В ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

ЦЕЛЬ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

Целью выполнения лабораторной работы является формирование практических навыков по настройке маршрутизации.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с реализацией функций маршрутизатора в системах на базе ОС Windows.
2. Изучить функционирование протоколов маршрутизации и средств диагностики.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Большинство протоколов маршрутизации, применяемых в современных сетях с коммутацией пакетов, ведут свое происхождение от сети Internet и ее предшественницы — сети ARPANET. Для того чтобы понять их назначение и особенности, полезно сначала познакомиться со структурой сети Internet, которая наложила отпечаток на терминологию и типы протоколов.

Internet изначально строилась как сеть, объединяющая большое количество существующих систем. С самого начала в ее структуре выделяли магистральную сеть (core backbone network): а сети, присоединенные к магистрали, рассматривались как автономные системы (autonomous systems, AS). Магистральная сеть и каждая из автономных систем имели свое собственное административное управление и собственные протоколы маршрутизации. Необходимо подчеркнуть, что автономная система и домен имен Internet — это разные понятия, кото-

рые служат разным целям. Автономная система объединяет сети, в которых под общим административным руководством одной организации осуществляется маршрутизация, а домен объединяет компьютеры (возможно, принадлежащие разным сетям), в которых под общим административным руководством одной организации осуществляется назначение уникальных символьных имен. Естественно, области действия автономной системы и домена имен могут в частном случае совпадать, если одна организация выполняет обе указанные функции.

Общая схема архитектуры сети Internet показана на рис. 3.1. Далее маршрутизаторы мы будем называть шлюзами, чтобы оставаться в русле традиционной терминологии Internet.

Шлюзы, которые используются для образования сетей и подсетей внутри автономной системы, называются внутренними шлюзами (interior gateways), а шлюзы, с помощью которых автономные системы присоединяются к магистрали сети, называются внешними шлюзами (exterior gateways). Магистраль сети также является автономной системой. Все автономные системы имеют уникальный 16-разрядный номер, который выделяется организацией, учредившей новую автономную систему, InterNIC.

Соответственно протоколы маршрутизации внутри автономных систем называются протоколами внутренних шлюзов (interior gateway protocol, IGP), а протоколы, определяющие обмен маршрутной информацией между внешними шлюзами и шлюзами магистральной сети — протоколами внешних шлюзов (exterior gateway protocol, EGP). Внутри магистральной сети также допустим любой собственный внутренний протокол IGP.

Смысл разделения всей сети Internet на автономные системы — в ее многоуровневом модульном представлении, что необходимо для любой крупной системы, способной к расширению в больших масштабах. Изменение протоколов маршрутизации внутри какой-либо автономной системы никак не должно влиять на работу остальных автономных систем. Кроме того, деление Internet на автономные системы должно способствовать агрегированию информации в магистральных и внешних шлюзах. Внутренние шлюзы могут использовать для внутренней

маршрутизации достаточно подробные графы связей между собой, чтобы выбрать наиболее рациональный маршрут. Однако если информация такой степени детализации будет храниться во всех маршрутизаторах сети, то топологические базы данных так разрастутся, что потребуют наличия памяти гигантских размеров, а время принятия решений о маршрутизации станет неприемлемо большим.

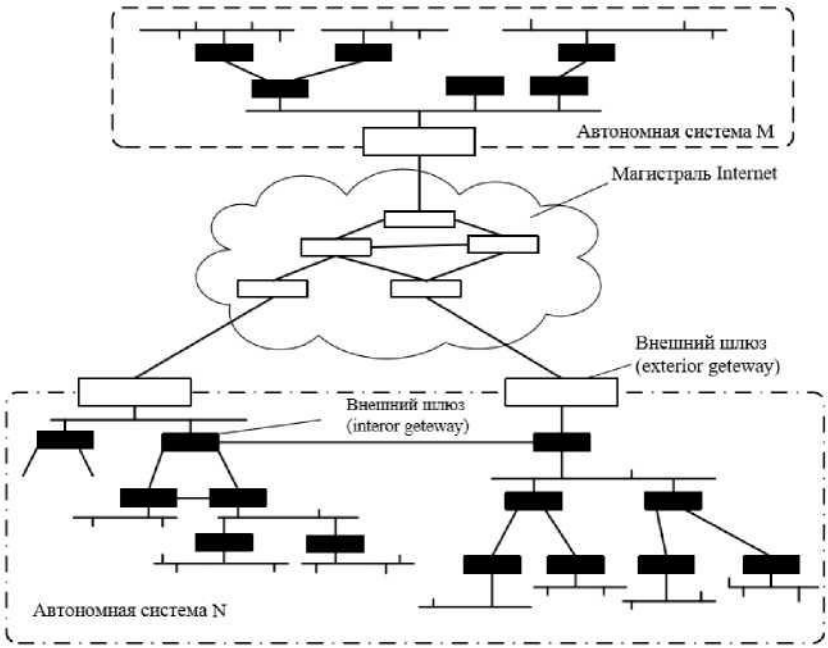


Рис. 3.1. Магистраль и автономные системы Internet

Поэтому детальная топологическая информация остается внутри автономной системы, а автономную систему как единое целое для остальной части Internet представляют внешние шлюзы, которые сообщают о внутреннем составе автономной системы минимально необходимые сведения — количество IP-сетей, их адреса и внутреннее расстояние до этих сетей от данного внешнего шлюза.

Приведенная на рис.3.1. структура Internet с единственной магистралью достаточно долго соответствовала действительности, поэтому

специально для нее был разработан протокол обмена маршрутной информации между автономными системами, названный EGP. Однако по мере развития сетей поставщиков услуг структура Internet стала гораздо более сложной, с произвольным характером связей между автономными системами. Поэтому протокол EGP уступил место протоколу BGP, который позволяет распознать наличие петель между автономными системами и исключить их из межсистемных маршрутов. Протоколы EGP и BGP используются только во внешних шлюзах автономных систем, которые чаще всего организуются поставщиками услуг Internet. В маршрутизаторах корпоративных сетей работают внутренние протоколы маршрутизации, такие как RIP и OSPF.

Построение таблицы маршрутизации

Протокол RIP (Routing Information Protocol)

Протокол RIP является внутренним протоколом маршрутизации дистанционно-векторного типа, он представляет собой один из наиболее ранних протоколов обмена маршрутной информацией и до сих пор чрезвычайно распространен в вычислительных сетях ввиду простоты реализации.

Для IP имеются две версии протокола RIP: первая и вторая. Протокол RIPvl не поддерживает масок, то есть он распространяет между маршрутизаторами только информацию о номерах сетей и расстояниях до них, а информацию о масках этих сетей не распространяет, считая, что все адреса принадлежат к стандартным классам A, B или C. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня. Так как при построении таблиц маршрутизации работа версии 2 принципиально не отличается от версии 1, то в дальнейшем для упрощения записей будет описываться работа первой версии.

В качестве расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, метрики, учитывающие пропускную способность, вносимые задержки и надежность сетей, а также любые

комбинации этих метрик. Метрика должна обладать свойством аддитивности — метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализации RIP используется простейшая метрика — количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 3.2.

Этап 1 — создание минимальных таблиц В этой сети имеется восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: Ml, М2, М3 и М4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако для работы протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются.

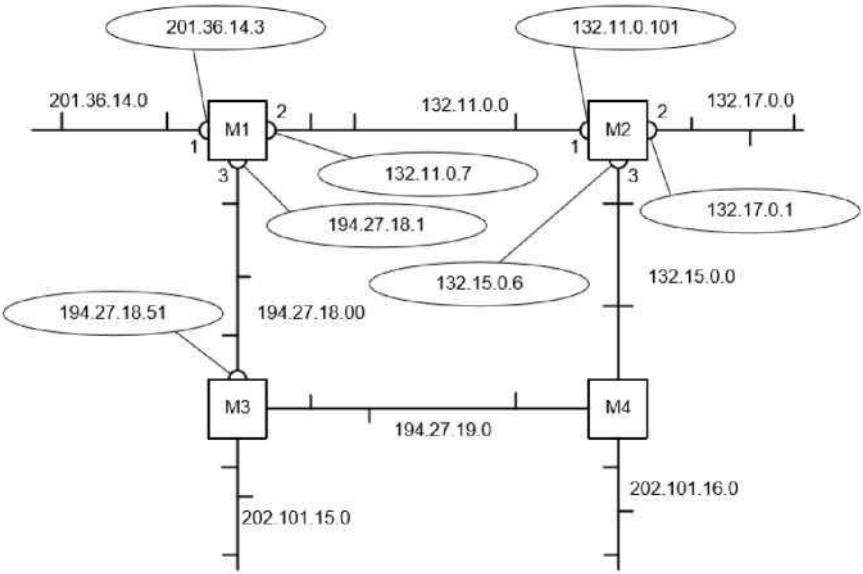


Рис. 3. 2. Сеть, объединенная RIP-маршрутизаторами

В исходном состоянии в каждом маршрутизаторе программным

обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети. На рисунке адреса портов маршрутизаторов в отличие от адресов сетей помещены в овалы.

Этап 2 — рассылка минимальных таблиц соседям. После инициализации каждого маршрутизатора он начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица.

RIP-сообщения передаются в пакетах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

Соседями являются те маршрутизаторы, которым данный маршрутизатор непосредственно может передать IP-пакет по какой- либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора Ml соседями являются маршрутизаторы М2 и М3, а для маршрутизатора М4 — маршрутизаторы М2 и М3.

Таким образом, маршрутизатор Ml передает маршрутизатору М2 и М3 следующее сообщение:

сеть 201.36.14.0, расстояние 1;
сеть 132.11.0.0, расстояние 1;
сеть 194.27.18.0, расстояние 1.

Этап 3 — получение RIP-сообщений от соседей и обработка полученной информации. После получения аналогичных сообщений от маршрутизаторов М2 и М3 маршрутизатор Ml наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора будет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации.

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (расстояние в хо-

пах меньше), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись; если же имеется несколько равнозначных в отношении расстояния путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение — если худшая информация о какой- либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 — рассылка новой, уже не минимальной, таблицы соседям Каждый маршрутизатор отсылает новое RIP-сообщёние всем своим соседям. В этом сообщении он помещает данные о всех известных ему сетях — как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 - получение RIP-сообщений от соседей и обработка полученной информации. Этап 5 повторяет этап 3 — маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

На этом этапе маршрутизатор Ml получил от маршрутизатора М3 информацию о сети 132.15.0.0, которую тот в свою очередь на предыдущем цикле работы получил от маршрутизатора М4. Маршрутизатор уже знает о сети 132.15.0.0, причем старая информация имеет лучшую метрику, чем новая, поэтому новая информация об этой сети отбрасывается.

О сети 202.101.16.0 маршрутизатор Ml узнает на этом этапе впервые, причем данные о ней приходят от двух соседей — от М3 и М4. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, которые пришли первыми. В нашем примере считается, что маршрутизатор М2 опередил маршрутизатор М3 и первым переслал свое RIP-сообщение маршрутизатору Ml.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится

корректный режим маршрутизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети будут достижимы из любой сети с помощью некоторого рационального маршрута. Пакеты будут доходить до адресатов и не зацикливаться в деталях, подобных той, которая образуется на рис. 3.2, маршрутизаторами М1-М2-М3-М4.

Очевидно, если в сети все маршрутизаторы, их интерфейсы и соединяющие их каналы связи постоянно работоспособны, то объявления по протоколу RIP можно делать достаточно редко, например, один раз в день. Однако в сетях постоянно происходят изменения — изменяется как работоспособность маршрутизаторов и каналов, так и сами маршрутизаторы, и каналы могут добавляться в существующую сеть или же выводиться из ее состава.

Для адаптации к изменениям в сети протокол RIP использует ряд механизмов.

Адаптация RIP-маршрутизаторов к изменениям состояния сети

К новым маршрутам RIP-маршрутизаторы приспосабливаются просто — они передают новую информацию в очередном сообщении своим соседям и постепенно эта информация становится известна всем маршрутизаторам сети. А вот к отрицательным изменениям, связанным с потерей какого-либо маршрута, RIP-маршрутизаторы приспосабливаются сложнее. Это связано с тем, что в формате сообщений протокола RIP нет поля, которое бы указывало на то, что путь к данной сети больше не существует.

Вместо этого используются два механизма уведомления о том, что некоторый маршрут более недействителен:

- истечение времени жизни маршрута;
- указание специального расстояния (бесконечности) до сети, ставшей недоступной.

Для отработки первого механизма каждая запись таблицы маршрутизации (как и записи таблицы продвижения моста/коммутатора), полученная по протоколу RIP, имеет время жизни (TTL). При поступле-

нии очередного RIP-сообщения, которое подтверждает справедливость данной записи, таймер TTL устанавливается в исходное состояние, а затем из него каждую секунду вычитается единица. Если за время тайм-аута не придет новое маршрутное сообщение об этом маршруте, то он помечается как недействительный.

Время тайм-аута связано с периодом рассылки векторов по сети. В RIP IP период рассылки выбран равным 30 секундам, а в качестве таймаута выбрано шестикратное значение периода рассылки, то есть 180 секунд. Выбор достаточно малого времени периода рассылки объясняется несколькими причинами, которые станут понятны из дальнейшего изложения. Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступна, а не просто произошли потери RIP сообщений (а это возможно, так как RIP использует транспортный протокол UDP, который не обеспечивает надежной доставки сообщений).

Если какой-либо маршрутизатор отказывает и перестает слать своим соседям сообщения о сетях, которые можно достичь через него, то через 180 секунд все записи, которые породил этот маршрутизатор, станут недействительными у его ближайших соседей. После этого процесс повторится уже для соседей ближайших соседей — они вычеркнут подобные записи уже через 360 секунд, так как первые 180 секунд ближайшие соседи еще передавали сообщения об этих записях.

Как видно из объяснения, сведения о недоступных через отказавший маршрутизатор сетях распространяются по сети не очень быстро, время распространения кратно времени жизни записи, а коэффициент кратности равен количеству хопов между самыми дальними маршрутизаторами сети. В этом заключается одна из причин выбора в качестве периода рассылки небольшой величины в 30 секунд.

Если отказывает не маршрутизатор, а интерфейс или сеть, связывающие его с каким-либо соседом, то ситуация сводится к только что описанной — снова начинает работать механизм тайм-аута и ставшие недействительными маршруты постепенно будут вычеркнуты из таблиц всех маршрутизаторов сети.

Тайм-аут работает в тех случаях, когда маршрутизатор не может

послать соседям сообщение об отказавшем маршруте, так как-либо он сам неработоспособен, либо неработоспособна линия связи, по которой можно было бы передать сообщение.

Когда же сообщение послать можно, RIP-маршрутизаторы не используют специальный признак в сообщении, а указывают бесконечное расстояние до сети, причем в протоколе RIP оно выбрано равным 16 хопам (при другой метрике необходимо указать маршрутизатору ее значение, считающееся бесконечностью). Получив сообщение, в котором некоторая сеть сопровождается расстоянием 16 (или 15, что приводит к тому же результату, так как маршрутизатор наращивает полученное значение на 1), маршрутизатор должен проверить, исходит ли эта «плохая» информация о сети от того же маршрутизатора, сообщение которого послужило в свое время основанием для записи о данной сети в таблице маршрутизации. Если это тот же маршрутизатор, то информация считается достоверной и маршрут помечается как недоступный.

Такое небольшое значение «бесконечного» расстояния вызвано тем, что в некоторых случаях отказы связей в сети вызывают длительные периоды некорректной работы RIP-маршрутизаторов, выражающейся в зацикливании пакетов в петлях сети. И чем меньше расстояние, используемое в качестве «бесконечного», тем такие периоды становятся короче.

Рассмотрим случай зацикливания пакетов на примере сети, изображенной на рис. 3.2.

Пусть маршрутизатор Ml обнаружил, что его связь с непосредственно подключенной сетью 201.36.14.0 потеряна (например, по причине отказа интерфейса 201.36.14.3). Ml отметил в своей таблице маршрутизации, что сеть 201.36.14.0 недоступна. В худшем случае он обнаружил это сразу же после отправки очередных RIP-сообщений, так что до начала нового цикла его объявлений, в котором он должен сообщить соседям, что расстояние до сети 201.36.14.0 стало равным 16, остается почти 30 секунд.

Каждый маршрутизатор работает на основании своего внутреннего таймера, не синхронизируя работу по рассылке объявлений с другими

маршрутизаторами. Поэтому весьма вероятно, маршрутизатор М2 опередил маршрутизатор Ml и передал ему свое сообщение раньше, чем Ml успел передать новость о недостижимости сети 201.36.14.0. А в этом сообщении имеются данные, порожденные следующей записью в таблице маршрутизации М2.

Эта запись была получена от маршрутизатора Ml и корректна до отказа интерфейса 201.36.14.3, а теперь она устарела, но маршрутизатор М2 об этом не узнал.

Теперь маршрутизатор Ml получил новую информацию о сети 201.36.14.0 — эта сеть достижима через маршрутизатор М2 с метрикой 2. Раньше Ml также получал эту информацию от М2. Но игнорировал ее, так как его собственная метрика для 201.36.14.0 была лучше. Теперь Ml должен принять данные о сети 201.36.14.0, полученные от М2, и заменить запись в таблице маршрутизации о недостижимости этой сети.

В результате в сети образовалась маршрутная петля: пакеты, направляемые узлам сети 201.36.14.0, будут передаваться маршрутизатором М2 маршрутизатору Ml, а маршрутизатор Ml будет возвращать их маршрутизатору М2. IP-пакеты будут циркулировать по этой петле до тех пор, пока не истечет время жизни каждого пакета.

Маршрутная петля будет существовать в сети достаточно долго. Рассмотрим периоды времени, кратные времени жизни записей в таблицах маршрутизаторов.

- Время 0-180 с. После отказа интерфейса в маршрутизаторах Ml и М2 будут сохраняться некорректные записи, приведенные выше. Маршрутизатор М2 по-прежнему снабжает маршрутизатор Ml своей записью о сети 201.36.14.0 с метрикой 2, так как ее время жизни не истекло. Пакеты зацикливаются.
- Время 180-360 с. В начале этого периода у маршрутизатора М2 истекает время жизни записи о сети 201.36.14.0 с метрикой 2, так как маршрутизатор Ml в предыдущий период посылал ему сообщения о сети 201.36.14.0 с худшей метрикой, чем у М2, и они не могли подтверждать эту запись. Теперь маршрутизатор

М2 принимает от маршрутизатора Ml запись о сети 201.36.14.0

с метрикой 3 и трансформирует ее в запись с метрикой 4. Маршрутизатор Ml не получает новых сообщений от маршрутизатора М2 о сети 201.36.14.0 с метрикой 2, поэтому время жизни его записи начинает уменьшаться. Пакеты продолжают зацикливаться.

• Время 360-540 с. Теперь у маршрутизатора Ml истекает время жизни записи о сети 201.36.14.0 с метрикой 3. Маршрутизаторы Ml и М2 опять меняются ролями — М2 снабжает Ml устаревшей информацией о пути к сети 201.36.14.0, уже с метрикой 4, которую Ml преобразует в метрику 5. Пакеты продолжают зацикливаться.

Если бы в протоколе RIP не было выбрано расстояние 16 в качестве недостижимого, то описанный процесс длился бы до бесконечности (вернее, пока не была бы исчерпана разрядная сетка поля расстояния и не было бы зафиксировано переполнения при очередном наращивании расстояния).

В результате маршрутизатор М2 на очередном этапе описанного процесса получает от маршрутизатора Ml метрику 15, которая после наращивания, превращаясь в метрику 16, фиксирует недостижимость сети. Период нестабильной работы сети длился 36 минут!

Ограничение в 15 хопов сужает область применения протокола RIP до сетей, в которых число промежуточных маршрутизаторов не может быть больше 15. Для более масштабных сетей нужно применять другие протоколы маршрутизации, например, OSPF, или разбивать сеть на автономные области.

Приведенный пример хорошо иллюстрирует главную причину нестабильной работы маршрутизаторов, работающих по протоколу RIP. Эта причина коренится в самом принципе работы дистанционно-векторных протоколов — пользовании информацией, полученной из вторых рук. Действительно, маршрутизатор М2 передал маршрутизатору Ml информацию о достижимости сети 201.36.14.0, за достоверность которой он сам не отвечает. Искоренить эту причину полностью нельзя, ведь сам способ построения таблиц маршрутизации связан с передачей чужой информации без указания источника ее происхождения.

Не следует думать, что при любых отказах интерфейсов и маршрутизаторов в сетях возникают маршрутные петли. Если бы маршрутизатор Ml успел передать сообщение о недостижимости сети 201.36.14.0 раньше ложной информации маршрутизатора М2, то маршрутная петля не образовалась бы. Так что маршрутные петли даже без дополнительных методов борьбы с ними, описанными в следующем разделе, возникают в среднем не более чем в половине потенциально возможных случаев.

Методы борьбы с ложными маршрутами в протоколе RIP.

Несмотря на то, что протокол RIP не в состоянии полностью исключить переходные состояния в сети, когда некоторые маршрутизаторы пользуются устаревшей информацией об уже несуществующих маршрутах, имеется несколько методов, которые во многих случаях решают подобные проблемы.

Ситуация с петлей, образующейся между соседними маршрутизаторами, описанная в предыдущем разделе, надежно решается с помощью метода, получившем название расщепления горизонта (split horizon). Метод заключается в том, что маршрутная информация о некоторой сети, хранящаяся в таблице маршрутизации, никогда не передается тому маршрутизатору, от которого она получена (это следующий маршрутизатор в данном маршруте). Если маршрутизатор М2 в рассмотренном выше примере поддерживает технику расщепления горизонта, то он не передаст маршрутизатору Ml устаревшую информацию о сети 201.36.14.0, так как получил ее именно от маршрутизатора Ml.

Практически все сегодняшние маршрутизаторы, работающие по протоколу RIP, используют технику расщепления горизонта.

Однако расщепление горизонта не помогает в тех случаях, когда петли образуются не двумя, а несколькими маршрутизаторами.

Рассмотрим более детально ситуацию, которая возникнет в сети, приведенной на рис. 3.2, в случае потери связи маршрутизатора 2 с сетью А. Пусть все маршрутизаторы этой сети поддерживают технику расщепления горизонта. Маршрутизаторы М2 и М3 не будут возвращать маршрутизатору в этой ситуации данные о сети 201.36.14.0 с метрикой

2, так как они получили эту информацию от маршрутизатора Ml. Однако они будут передавать маршрутизатору информацию о достижимости сети 201.36.14.0 с метрикой 4 через себя, так как получили эту информацию по сложному маршруту, а не от маршрутизатора Ml непосредственно. Например, маршрутизатор М2 получил эту информацию по цепочке М4-М3-М1. Поэтому маршрутизатор Ml снова может быть обманут, пока каждый из маршрутизаторов в цепочке М3-М4М2 не вычеркнет запись о достижимости сети 1 (а это произойдет через период 3 х 180 секунд).

Для предотвращения зацикливания пакетов по составным петлям при отказах связей применяются два других приема, называемые триггерными *обновлениями (triggered updates)* и *замораживанием изменений (hold down).*

Способ триггерных обновлений состоит в том, что маршрутизатор, получив данные об изменении метрики до какой-либо сети, не ждет истечения периода передачи таблицы маршрутизации, а передает данные об изменившемся маршруте немедленно. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но он перегружает сеть служебными сообщениями, поэтому триггерные объявления также делаются с некоторой задержкой. Поэтому возможна ситуация, когда регулярное обновление в каком-либо маршрутизаторе чуть опередит по времени приход триггерного обновления от предыдущего в цепочке маршрутизатора и данный маршрутизатор успеет передать по сети устаревшую информацию о несуществующем маршруте.

Второй прием позволяет исключить подобные ситуации. Он связан с введением тайм-аута на принятие новых данных о сети, которая только что стала недоступной. Этот тайм-аут предотвращает принятие устаревших сведений о некотором маршруте от тех маршрутизаторов, которые находятся на некотором расстоянии от отказавшей связи и передают устаревшие сведения о ее работоспособности. Предполагается, что в течение тайм-аута «замораживания изменений» эти маршрутизаторы вычеркнут данный маршрут из своих таблиц, так как не получат о нем новых записей и не будут распространять устаревшие све-

дения по сети.

Протокол «состояния связей» OSPF

Протокол OSPF (Open Shortest Path First, открытый протокол «кратчайший путь первым») является реализацией алгоритма состояния связей (он принят в 1991 году) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

В OSPF процесс построения таблицы маршрутизации разбивается на два крупных этапа. На первом этапе каждый маршрутизатор строит граф связей сети, в котором вершинами графа являются маршрутизаторы и IP-сети, а ребрами — интерфейсы маршрутизаторов. Все маршрутизаторы для этого обмениваются со своими соседями той информацией о графе сети, которой они располагают к данному моменту времени. Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно другая — это информация о топологии сети. Эти сообщения называются router links advertisement — объявление о связях маршрутизатора. Кроме того, при передаче топологической информации маршрутизаторы ее не модифицируют, как это делают RIP маршрутизаторы, а передают в неизменном виде. В результате распространения топологической информации все маршрутизаторы сети располагают идентичными сведениями о графе сети, которые хранятся в топологической базе данных маршрутизатора.

Второй этап состоит в нахождении оптимальных маршрутов с помощью полученного графа. Каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут до каждой известной ему сети. В каждом найденном таким образом маршруте запоминается только один шаг — до следующего маршрутизатора, в соответствии с принципом одношаговой маршрутизации. Данные об этом шаге и попадают в таблицу маршрутизации.

После первоначального построения таблицы маршрутизации необходимо отслеживать изменения состояния сети и вносить коррективы в таблицу маршрутизации. Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутизаторы передают специальные ко-

роткие сообщения HELLO. Если состояние сети не меняется, то OSPF маршрутизаторы корректировкой своих таблиц маршрутизации не занимаются и не посылают соседям объявления о связях. Если же состояние связи изменилось, то ближайшим соседям посылается новое объявление, касающееся только данной связи, что, конечно, экономит пропускную способность сети. Получив новое объявление об изменении состояния связи, маршрутизатор перестраивает граф сети, заново ищет оптимальные маршруты (не обязательно все, а только те, на которых отразилось данное изменение) и корректирует свою таблицу маршрутизации. Одновременно маршрутизатор ретранслирует объявление каждому из своих ближайших соседей (кроме того, от которого он получил это объявление).

При появлении новой связи или нового соседа маршрутизатор узнает об этом из новых сообщений HELLO. В сообщениях HELLO указывается достаточно детальная информация о том маршрутизаторе, который послал это сообщение, а также о его ближайших соседях, чтобы данный маршрутизатор можно было однозначно идентифицировать. Сообщения HELLO отправляются через каждые 10 секунд, чтобы повысить скорость адаптации маршрутизаторов к изменениям, происходящим в сети. Небольшой объем этих сообщений делает возможной такое частое тестирование состояния соседей и связей с ними.

Так как маршрутизаторы являются одними из вершин графа, то они обязательно должны иметь идентификаторы.

Протокол OSPF обычно использует метрику, учитывающую пропускную способность сетей. Кроме того, возможно использование двух других метрик, учитывающих требования к качеству обслуживания в IP пакете, — задержки передачи пакетов и надежности передачи пакетов сетью. Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Выбор нужной таблицы происходит в зависимости от требований к качеству обслуживания пришедшего пакета (см. рис. 3.3).

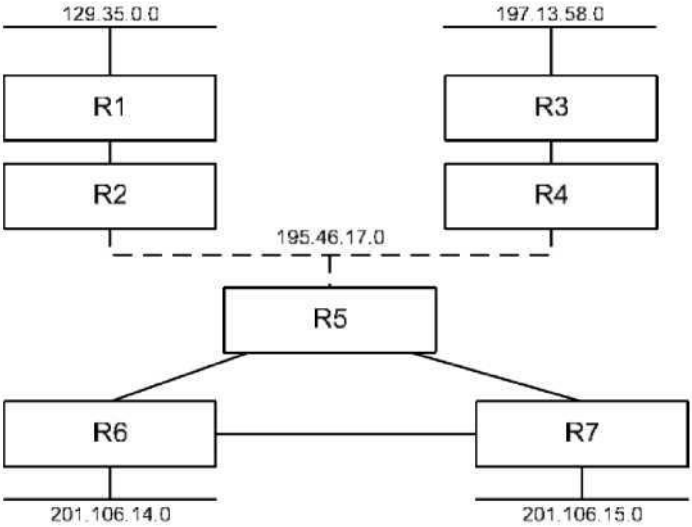


Рис. 3.3. Построение таблицы маршрутизации по протоколу OSPF

Маршрутизаторы соединены как с локальными сетями, так и непосредственно между собой глобальными каналами типа «точка- точка». Данной сети соответствует граф, приведенный на рис. 3.4.

Протокол OSPF в своих объявлениях распространяет информацию о связях двух типов: маршрутизатор - маршрутизатор и маршрутизатор - сеть. Примером связи первого типа служит связь «R3 - R4», а второго — связь «R4 - 195.46.17.0». Если каналам «точка-точка» дать IP-адреса, то они станут дополнительными вершинами графа, как и локальные сети. Вместе с 1Радресом сети передается также информация о маске сети.

После инициализации OSPF-маршрутизаторы знают только о связях с непосредственно подключенными сетями, как и RIP- маршрутизаторы. Они начинают распространять эту информацию своим соседям. Одновременно они посылают сообщения HELLO по всем своим интерфейсам, так что почти сразу же маршрутизатор узнает идентификаторы своих ближайших соседей, что пополняет его топологическую базу новой информацией, которую он узнал непосредственно. Далее топологическая информация начинает распространяться по сети от со-

седа к соседу и через некоторое время достигает самых удаленных маршрутизаторов.

Каждая связь характеризуется метрикой. Протокол OSPF поддерживает стандартные для многих протоколов (например, для протокола Spanning Tree) значения расстояний для метрики, отражающей производительность сетей: Ethernet — 10 единиц, Fast Ethernet — 1 единица, канал Tl — 65 единиц, канал 56 Кбит/с — 1785 единиц и т. д.

При выборе оптимального пути на графе с каждым ребром графа связана метрика, которая добавляется к пути, если данное ребро в него входит. Пусть на приведенном примере маршрутизатор R5 связан с R6 и R7 каналами Tl, а R6 и R7 связаны между собой каналом 56 Кбит/с. Тогда R7 определит оптимальный маршрут до сети 201.106.14.0 как составной, проходящий сначала через маршрутизатор R5, а затем через R6, поскольку у этого маршрута метрика будет равна 65+65 = 130 единиц. Непосредственный маршрут через R6 не будет оптимальным, так как его метрика равна 1785. При использовании хопов был бы выбран маршрут через R6, что не было бы оптимальным.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками. Если такие записи образуются в таблице маршрутизации, то маршрутизатор реализует режим баланса загрузки маршрутов (load balancing), отправляя пакеты попеременно по каждому из маршрутов.

При инициализации маршрутизаторов, а также для более надежной синхронизации топологических баз маршрутизаторы периодически обмениваются всеми записями базы, но этот период существенно больше, чем у RIP-маршрутизаторов.

Так как информация о некоторой связи изначально генерируется только тем маршрутизатором, который выяснил фактическое состояние этой связи путем тестирования с помощью сообщений HELLO, а остальные маршрутизаторы только ретранслируют эту информацию без преобразования, то недостоверная информация о достижимости сетей, которая может появляться в RIP-маршрутизаторах, в OSPF-маршрутизаторах появиться не может, а устаревшая информация быстро заменяется новой, так как при измене-

нии состояния связи новое сообщение генерируется сразу же.

Периоды нестабильной работы в OSPF-сетях могут возникать. Например, при отказе связи, когда информация об этом не дошла до какого либо маршрутизатора и он отправляет пакеты сети назначения, считая эту связь работоспособной. Однако эти периоды продолжаются недолго, причем пакеты не зацикливаются в маршрутных петлях, а просто отбрасываются при невозможности их передать через неработоспособную связь.

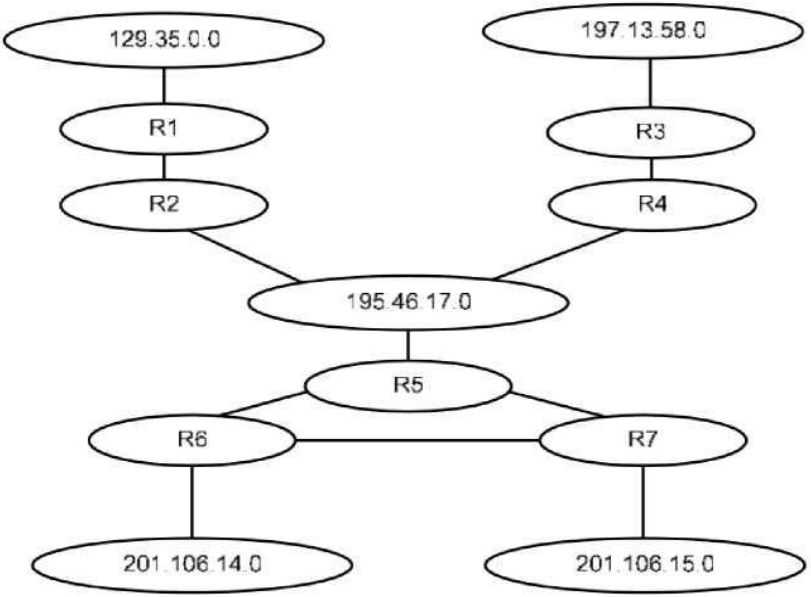


Рис. 3.4. Граф сети, построенный протоколом OSPF

К недостаткам протокола OSPF следует отнести его вычислительную сложность, которая быстро растет с увеличением размерности сети, то есть количества сетей, маршрутизаторов и связей между ними. Для преодоления этого недостатка в протоколе OSPF вводится понятие области сети (area) (не нужно путать с автономной системой Internet). Маршрутизаторы, принадлежащие некоторой области, строят граф связей только для этой области, что сокращает размерность сети. Между областями информация о связях не передается, а пограничные для

областей маршрутизаторы обмениваются только информацией об адресах сетей, имеющихся в каждой из областей, и расстоянием от пограничного маршрутизатора до каждой сети. При передаче пакетов между областями выбирается один из пограничных маршрутизаторов области, а именно тот, у которого расстояние до нужной сети меньше. Этот стиль напоминает стиль работы протокола RIP, но нестабильность здесь устраняется тем, что петлевидные связи между областями запрещены. При передаче адресов в другую область OSPF маршрутизаторы агрегируют несколько адресов в один, если обнаруживают у них общий префикс.

OSPF-маршрутизаторы могут принимать адресную информацию от других протоколов маршрутизации, например, от протокола RIP, что полезно для работы в гетерогенных сетях. Такая адресная информация обрабатывается так же, как и внешняя информация между разными областями.

Команда route и таблица маршрутизации в Windows

Создание статических маршрутов выполняется с помощью специальных программ из комплекта TCP/IP. В Windows утилита для создания (или удаления) элементов таблицы маршрутизации называется Route.exe и запускается из командной строки со следующими параметрами и переключателями:

ROUTE [-f] [-p] [command [destination] [MASK netmask] [gateway] [METRIC metric] [IF interface]]

-f — удаляет все элементы в таблице маршрутизации. При использовании этого переключателя с командой ADD сначала удаляются старые элементы, а затем добавляется новый.

-p — при использовании с командой ADD создает в таблице постоянный элемент маршрута, который сохраняется даже после перезапуска системы. При использовании с командой PRINT отображает на экране только постоянные маршруты.

command — ключевое слово, которое конкретизирует выполняемое

действие.

destination — адрес сети или хоста в строке таблицы, на которую направлено действие команды.

MASK netmask — маска подсети, которую следует применять к адресу, заданному в переменной destination.

gateway — адрес маршрутизатора, на который должны отправляться пакеты, адресованные хосту или сети, заданным в переменной destination.

METRIC metric — значение метрики, характеризующее относительную эффективность данного маршрута.

IF interface — адрес платы сетевого адаптера, которой система должна пользоваться для передачи данных маршрутизатору, адрес которого задан в переменной gateway.

Переменная command принимает одно из четырех значений:

- PRINT — отобразить содержимое таблицы маршрутизации (при использовании с параметром p отображаются только неудаляемые маршруты);
- ADD — создать новый маршрут;
- DELETE — удалить существующий маршрут;
- CHANGE — изменить параметры существующего маршрута.

Команда **ROUTE PRINT** отображает текущее содержимое таблицы маршрутизации. Для удаления маршрута воспользуйтесь командой **ROUTE DELETE**, указав с помощью переменной destination, какой маршрут нужно удалить. Чтобы создать новый маршрут, введите команду **ROUTE ADD** с параметрами маршрута, заданными в соответствующих переменных. Подобным образом работает и команда **ROUTE CHANGE**, за исключением того, что указанные в ней параметры присваиваются существующему маршруту, заданному с помощью переменной destination. Переменная destination содержит адрес сети или хоста, информацию о маршруте, к которым вы вводите. Другими переменными задаются маска подсети, адрес шлюза, адрес интерфейса и эффективность маршрута.

Таблица 3.1. Структура таблицы маршрутизации в Windows

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.2.100	192.168.2.5	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.2.0	255.255.255.0	192.168.2.5	192.168.2.5	1
192.168.2.5	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.2.255	255.255.255.255	192.168.2.5	192.168.2.5	1
224.0.0.0	224.0.0.0	192.168.2.5	192.168.2.5	1
255.255.255.255	255.255.255.255	192.168.2.5	0.0.0.0	1

Записи в таблице расположены горизонтально. Назначение информации в каждом из столбцов приведено ниже.

- Сетевой адрес (Network Address). Содержит адрес сети, для которой приведена информация маршрутизации. В общем случае для большинства записей в этом поле размещается адрес сети, но оно также может содержать информацию маршрутизации для определенного узла. Последняя называется маршрутом узла (host route).
- Маска подсети (Netmask). Задает так называемую маску подсети, используемую для определения, какие из битов в сетевом адресе являются идентификатором сети.

- Адрес шлюза (Gateway Address). Указывает IP-адрес шлюза (маршрутизатора), который система должна использовать для посылки

пакетов по заданному сетевому адресу. Если это запись для сети, к которой система подключена непосредственно, тогда поле содержит адрес сетевого интерфейса системы.

- Интерфейс (Interface). В этом столбце сохраняется IP-адрес сетевого интерфейса системы, служащий для отправки трафика по адресу шлюза.
- Метрика маршрута (Metric). Указывает расстояние между системой и сетью назначения, обычно выражается в количестве транзитов, необходимых для того, чтобы трафик достиг целевого адреса.

ПОСТРОЕНИЕ СЕТИ С МАРШРТИЗАТОРОМ С ИСПОЛЬЗОВАНИЕМ CISCO PACKET TRACER

Пусть необходимо построить маршрутизируемую IP-сеть, объединяющая несколько VLAN с помощью маршрутизатора, работающего в режиме Router:

Вариант 1 (рис. 3.5):

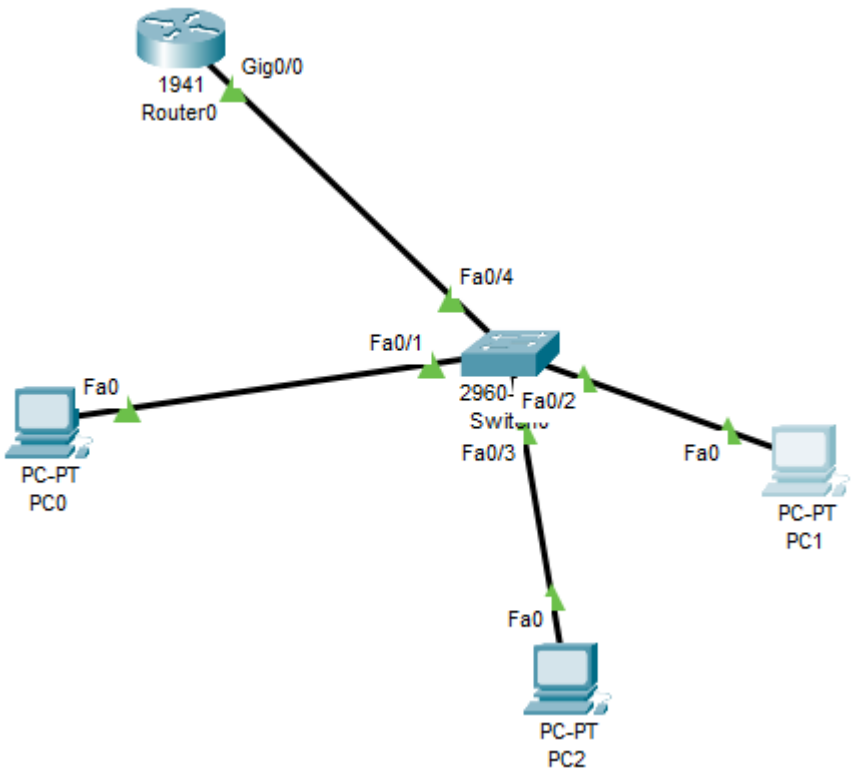


Рис. 3.5 . Маршрутизация по Варианту 1

1. Запускаем Cisco Packet Tracer;
2. Создадим три компьютера, один коммутатор 2960 и маршрутизатор 1941 (рис. 1). Пусть будет три сегмента VLAN2, VLAN3 и VLAN4.
3. Настроим коммутатор в режиме глобального конфигурирования.

- Создадим VLAN2, VLAN3 и VLAN4 и назначим им имена;
- Определяем компьютеры в соответствующий интерфейс. Компьютер PC0 подключен к интерфейсу fastEthernet 0/1, PC1 к

fastEthernet 0/2, PC1 к fastEthernet 0/3. Настройте интерфейсы коммутатора (FastEthernet 0/1, FastEthernet 0/2, FastEthernet 0/3) с помощью команд - interface fastEthernet 0/1, switchport mode access, switchport access vlan 2. Аналогично для оставшихся VLAN3 (vlan 3) и VLAN4(vlan 4);

- Настроим trunk порт коммутатора для обеспечения трафика всех vlan, который идет до маршрутизатора (в нашем случае порт коммутатора fastEthernet 0/4): configurate terminal, interface FastEthernet0/4, switchport mode trunk, switchport trunk allowed vlan 2,3,4.

4. Настроим маршрутизатор:

- Заходим в CLI (интерфейс командной строки);
- Режим глобального конфигурирования (enable-config terminal);
- Необходимо поднять физический порт (в нашем случае gigabitEthernet 0/0) с помощью команд interface gigabitEthernet 0/0, no shutdown.

5. Т.к. на маршрутизатор приходит три VLAN, необходимо создать подинтерфейсы, которым будут соответствовать свой VLAN с помощью команд interface gigabitEthernet 0/0.2, encapsulation dotlQ 2, и зададим IP адрес (ip address 192.168.2.1 255.255.255.0), no shutdown. Аналогично сделайте для VLAN 3 (interface gigabitEthernet 0/0.3, encapsulation dotlQ 3, ip address 192.168.3.1 255.255.255.0) и VLAN 4 (interface gigabitEthernet 0/0.4, encapsulation dotlQ 4, ip address 192.168.4.1 255.255.255.0)
6. Сохраните. (wr mem)
7. Настройте компьютеры. Например, в нашем случае для PC0 IP адрес 192.168.2.2, маска 255.255.255.0 и шлюз 192.168.2.1. Аналогично для остальных компьютеров (PC1, PC2).
8. Проверьте соединение.

Вариант 2

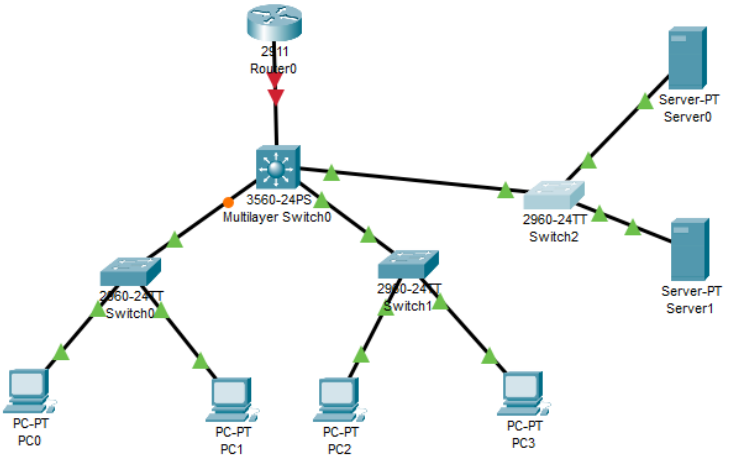


Рис.3.6. Маршрутизация по Варианту 2

1. Запускаем Cisco Packet Tracer;
2. Добавим в схему четыре компьютера, два сервера, три коммутатора 2960 и маршрутизаторы 3560 и 2911 (рис. 3.6). Пусть будет три сегмента VLAN2, VLAN3 и VLAN4.
3. Наши компьютеры PC0 и PC2 находятся в VLAN 2, PC1 и PC3 в VLAN 3, два сервера находятся в своем выделенном VLAN 4;
4. Настройте коммутаторы Switch0 (vlan2, vlan3), Switch1(vlan2, vlan3) и Switch3 (vlan4).
5. Для компьютеров присвойте IP адреса: PC0 – 192.168.2.2, PC2 – 192.168.2.3, PC1 – 192.168.3.2, PC3 – 192.168.3.3, для серверов 192.168.4.2 и 192.168.4.3;
6. Настройте коммутатор L3 (3560);

- Создайте VLAN 5;

- Настройте коммутатор L3 для данного сегмента. Поднимите виртуальный интерфейс с помощью команд ip address 192.168.55.2
- 255.255.255.0, no shutdown;
- Порт gigabitEthernet 0/1 определите как access порт;

7. Настроим маршрутизатор (2911):

- Режим глобального конфигурирования.
- Поднимите физический интерфейс (в нашем случае gigabitEthernet 0/0). И задаем IP адрес ip address 192.168.55.1 255.255.255.0;

8. Проверим сеть.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. С помощью Cisco Packet Tracer создать сети по вариантам 1 и 2.
2. Проверить работоспособность сетей при помощи утилит ping и tracert.
3. Ответить на контрольные вопросы и оформить отчет.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

Отчет на защиту предоставляется в печатном или электронном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, краткое описание тестируемой программы и ее модулей, результаты выполнения работы (скриншоты и содержимое файлов), выводы.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1 Дайте определение понятиям «магистральная сеть» и «автономные системы»

2 Раскройте различие внутренних и внешних шлюзов.

3 Раскройте различие протоколов внутренних и вешних шлюзов.

4 Раскройте различие протоколов EGP и BGP.

5 Приведите примеры внутренних протоколов IGP.

6 Опишите назначение протокола RIP.

7 Назовите метрики, предусмотренные стандартом протокола RIP для определения расстояния до сети.

8 Приведите этапы построения таблиц маршрутизации с помощью протокола RIP.

9 Назовите механизмы уведомления о недействительных маршрутах в протоколе RIP.

10 Перечислите методы борьбы с ложными маршрутами в протоколе RIP.

11 Раскройте сущность метода расщепления горизонта.

12 Раскройте сущность метода триггерных обновлений.

13 Раскройте сущность метода замораживания изменений.

14 Раскройте назначение протокола OSPF.

15 Приведите этапы построения таблиц маршрутизации с помощью протокола OSPF.

16 Перечислите недостатки протокола OSPF.

ЛАБОРАТОРНАЯ РАБОТА №4
НАСТРОЙКА ВИРТУАЛЬНОЙ ЛОКАЛЬНОЙ СЕТИ

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: формирование практических навыков по настройке и использованию коммутаторов в компьютерных сетях..

Задачи: пнять, что такое управляющее ПО, научиться управлять свитчем с использованием различных интерфейсов подключения, понять назначение адресных таблиц.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Свитч содержит ПО, которое позволяет менять и наблюдать режимы его работы. Это ПО не требует функционирования на свитче, но если вы это сделаете, то вы можете улучшить эффективность свитча, и кроме того улучшить в целом производительность вашей сети.

Таблица 4.1. Характеристики ПО, которые поддерживаются в свичах серии 1100/3300 и 610/630

Характеристика	**1100/610**	**3300/630**
Количество поддерживаемых MAC адресов	6000	12000
Групповое управление	Поддерживает до 4х свитчей (не поддерживает серии 610/630.)	Поддерживает до 4х свитчей (не поддерживает серии 610/630.)

Продолжение табл 4.1.

Характеристика	**1100/610**	**3300/630**

Модели адресации.	Хранение и отправление, быстрое отправление, свободная фрагмента-	Хранение и отправление.
Режимы duplex	Half и full на всех портах.	Half и fUll на всех портах.
Потоковый контроль	Поддерживается всеми портами.	Поддерживается всеми портами.
Приоритет трафика	Поддерживается	Поддерживается
PACE	Поддерживается всеми портами.	Поддерживается всеми портами.
Безопасность	Поддерживается всеми портами.	Поддерживается всеми портами.
Резервные связи	Поддерживается.	Поддерживается.
Транкование портов	Поддержка двух транковых портов	Поддержка двух транковых портов
Broadcast Storm control	Поддерживается	Поддерживается
Виртуальные локальные сети.	Поддерживается 16 VLANs использующих стандарт IEEE 802.1Q .	Поддерживается 16 VLANs использующих стандарт IEEE 802.1Q .
Fast IP	Поддерживается	Поддерживается
Групповая фильтрация.	Поддерживается IEEE 802.1p и IGMP фильтрация	Поддерживается IEEE 802.1p и IGMP фильтрация
Spanning Tree protocol	Поддерживается	Поддерживается

Продолжение табл 4.1.

Характеристика	1100/610	3300/630
RMON	Поддерживается 7 групп Статистика, История, Оповещения, хосты, Лучшие N и др.	Поддерживается 7 групп Статистика, История, Оповещения, хосты, и др.
Roving анализ	Поддерживается	Поддерживается
Управление	Поддерживается web интерфейс, консоль, и SNMP.	Поддерживается web интерфейс, консоль, и SNMP.

Характеристики ПО

Управление группой

Модули в свитчах моделей 1100/3300 могут быть взаимосвязаны, поэтому они являются стеком - группы устройств, которые управляются как одно устройство.

Групповые возможности не поддерживаются модулями свитчей моделей 610/630.

Вы можете соединить вместе эти модули свитчей двумя способами:

- Матричный порт сзади каждого свитча позволяет соединить два свитча. Для этого нужен Матричный кабель (номер 3C16965)
- Слот расширенного модуля сзади каждого свитча позволяет вам установить матричный модуль (раздельный номер 3C16965). Матричный модуль предоставляет 4 порта и допускает чтобы вы соединили 4 свитча используя Матричный кабель.

Продвижение пакетов

Модули в свитчах серии 3300/630 поддерживают хранение и пересылку пакетов. В этом режиме, полученные пакеты буферизуются полностью, перед тем как их направляют. Это гарантирует то, что только хорошие пакеты будут переданы по назначению.

В свитче 1100/610 поддерживает 3 модели адресации в дополнении к хранению и пересылке:

Быстрое продвижение - пакеты адресуются сразу же при считывании адреса получателя. При быстрой адресации, адресация пакетов занимает меньше времени, но ошибочные пакеты передаются в сеть, потому что нет времени их проверить.

Свободная фрагментация – пакеты адресуются, когда меньшие 512 бит пакета приняты, это дает уверенность, что пакеты, вызывающие коллизии, не пройдут в сеть. Со свободной фрагментацией, пакетам требуется меньше времени чтоб быть направленными, но все ошибочные пакеты, за исключением фрагментов, будут переданы.

Интеллектуальная модель - свитч наблюдает за количеством ошибочных пакетов в сети и соответственно меняет модель адресации. Если свитч обнаруживает 20 или больше ошибок в секунду, то модель адресации меняется на хранение и продвижение, пока количество ошибок в секунду не станет равно 0.

Дуплексный режим

Все порты на свитче могут быть установлены в один из двух дуплексных режимов.

Полу дуплекс - позволяет отправлять и получать пакеты, но не одновременно. Это модель дуплекса Eternet-а по умолчанию.

Полный дуплекс - позволяет отправлять и получать пакеты одновременно, в следствии удваивается пропускная способность связи. В дополнении полный дуплекс поддерживает 100BASE- FX кабель, протяжённостью до 2 км (656 фт).

Для эффективности связи оба конца должны использовать одинаковые режимы дуплекса. Если соединение использует авто определение связи, это будет сделано автоматически. Если соединение не использует авто определение связи, то оба порта должны быть переведены в режим полного или полу дуплекса вручную.

Контроль потоков

Все порты свитча поддерживают потоковый контроль, т.е. механизм контроля перегрузок. Причина перегрузки в том, что одно или несколько устройств посылает трафик на уже перегруженный порт свитча. Потоковый контроль предотвращает потерю пакетов и препятствуют генерации пакетов от устройства, пока перегрузка не закон-

читься.

Потоковый контроль осуществляется двумя путями:

- IEEE 802.3x стандарт, для портов, работающих в режиме полного дуплекса.
- Разумное Потоковое Управление (РПУ), патентованный метод потокового контроля компании 3Com, для портов с полу дуплексом. РПУ должно быть включено, если порт подключён к другому свитчу или конечной станции. Если порт подключен к повторителю с локальным трафиком, РПУ должен быть отключен.

Приоритезация трафика

Свитч поддерживает IEEE 802.1p приоритет трафика, при котором данные с высшем приоритетом проходят через свитч без задержек со стороны других данных. Система работает посредством использования многочисленных очередей трафика, которые присутствуют в аппаратуре свитча - трафик с большим приоритетом проходит по отдельным очередям от остального трафика, и он всегда обладает преимуществом по отношению к другому трафику.

PACE

Свитч поддерживает PACE ((Приоритетный доступ к управлению), который является собственностью 3com. Это позволяет мультимедийному трафику двигаться по сети эффективно.

PACE обеспечивает две главные характеристики:

- Скрытый класс сервиса - эта характеристика увеличивает приоритет трафика от приложений мультимедиа и обеспечивает такую же функциональность как IEEE 802.1p приоритет трафика.
- Интерактивный доступ - когда двусторонний мультимедийный трафик передается по Ethernet или Fast Ethernet, может возникнуть интерференция, потому что доступ к полосе пропускания неравномерно распределяется в одном направлении. Возможность интерактивного доступа позволяет распределять доступную полосу пропускания равномерно в двух направлениях, при этом улучшая качество мультимедийного трафика.

Безопасность

Каждый порт вашего свитча может использовать возможности

безопасности, которые защищают устройства в вашей сети от подключения посторонних пользователей. Когда возможности безопасности активированы на порту, то он работает в режиме Одиночного распознавания адреса.

В этом режиме свитч:

- Помещает все MAC адреса в базу данных свитча.
- Распознает адрес первого пакета, пришедшего в порт.
- Определяет адрес как постоянный.

Когда первый адрес распознан:

- Порт будет заблокирован, если на порт прибудет запрос от другого адреса.
- Никакой другой адрес не может быть распознан пока безопасность включена или адрес не удален из базы данных вручную.
- Адрес не может быть распознан другим портом пока безопасность включена или адрес не удален из базы данных вручную.

Резервные связи

Возможность использования резервных связей в свитче предоставляет возможность вам защитить критические связи и предотвратить время простоя сети, если эти связи нарушаться. Активируя резервные связи, вы уверены в том, что если главная коммуникационная связь откажет, то резервная дублирующая автоматически немедленно продолжит выполнять задачи главной связи. Каждая пара главной и резервной связи образуют пару резервных связей.

Резервные связи — это простой способ создать избыточность, которая обеспечивает вас мгновенной реакцией на ошибку связи. Резервные связи быстро настраиваются, вы имеете полный контроль над их конфигурацией, и порт на другом конце гибкой связи гибкой связи может и не поддерживать гибкую связь.

Транкование портов

Ваш свитч поддерживает транкование портов - соединение, позволяющее устройствам передавать данные используя до 4х связей параллельно. Транкование портов обладает двумя достоинствами:

- Они могут увеличить от 2х до 4х раз полосу пропускания.
- Они обеспечивают избыточность - если одно соединение обор-

вётся, то другие связи передадут трафик разорванного соединения.

Контроль широковещательного шторма

Свитч поддерживает контроль широковещательного шторма. Это система, которая автоматически генерирует сообщение для каждого порта при наблюдении уровня широковещательного трафика на порт. Если широковещательный трафик превышает 2976 пакетов в секунду, то порт блокирует широковещательный трафик до тех пор, пока его уровень не упадёт до 1488 пакетов в секунду. Эта система предотвращает обработку избыточного количества широковещательного трафика, который может быть результатом повреждения или неправильной настройки сетевого оборудования.

Виртуальные сети

Свитч поддерживает до 16 виртуальных сетей. Виртуальная сеть - это гибкая группа устройств, которые могут быть размещены где угодно в сети, но они обмениваются данными как один и тот же физический сегмент. При помощи виртуальной сети вы можете сегментировать вашу сеть без ограничения физических соединений - препятствия традиционного сетевого проектирования. В качестве примера при помощи виртуальной сети вы можете сегментировать вашу сеть таким образом:

- Группы департаментов - для примера вы можете иметь одну виртуальную сеть для департамента маркетинга, другую для финансового департамента и ещё одну для департамента исследований.
- Иерархические группы - одна сеть для директоров, другая для менеджеров, а третья для всего остального персонала.
- Группы использования - одна сеть для Эл почты, другая для приложений мультимедиа.

Fast IP

Ваш свитч поддерживает Fast IP, система, которая позволяет уменьшить нагрузку на маршрутизаторы, когда виртуальная сеть установлена в вашей сети.

Устройства из разных виртуальных сетей могут обмениваться данными используя устройства маршрутизации, если присутствует большое количество внутреннего трафика виртуальных сетей, то маршру-

тизатор может быть перегружен и производительность сети может быть уменьшена. Fast IP позволяет рабочим станциям и свитчу находить кратчайшие и безопасные участки для внутрисетевого трафика, который обходит маршрутизатор.

Многоадресная фильтрация

Свитч поддерживает 2 системы многоадресной фильтрации.

- IEEE 802.1p которая использует GARP Многоадресный протокол регистрации. (GMPR)
- IGMP (Протокол управления группами в Интернете)

Система позволяет свитчу адресовать многоадресный трафик к точке назначения, что лучше, чем осуществлять широковещание.

Протокол покрывающего дерева

Свитч поддерживает протокол покрывающего дерева (ППД). Это система размещения мостов, которая делает вашу сеть более гибкой к ошибкам связи и также предоставляет защиту от петель - одной из основных причин широковещательного шторма.

ППД позволяет осуществлять параллельные каналы сетевого трафика и использовать процесс обнаружения петель для того чтобы:

- Показывать эффективность каждого канала.
- Активировать самые эффективные каналы с самой широкой полосой пропускания.
- Отключать неэффективные каналы.
- Активировать менее эффективный канал, если более эффективный канал откажет.

Удаленный мониторинг

Свитч поддерживает удалённый мониторинг. Это система, которая позволяет вам контролировать сеть удаленно. Свитч имеет зонд удаленного мониторинга. Это ПО которое ежеминутно собирает информацию о сегментах сети, подключенной к свитчу. Если у вас имеется управляющая рабочая станция, с приложением удаленного мониторинга, свитч может передавать эту статистику на вашу рабочую станцию по запросу или когда пройден определенный порог.

Roving анализ

Свитч поддерживает roving анализ. Это система, которая позволяет

вам присоединить анализатор сети к одному из портов и использовать его для наблюдения за трафиком других портов свитча. Система работает путем подключения порта анализатора (порта к которому подключён анализатор), и наблюдаемого порта (за которым наблюдают). После того как это пара была определена, и вы включили систему, то свитч копирует весь входящий/исходящий трафик из наблюдаемого порта в порт анализа.

Управление

Вы можете управлять свитчем используя три метода:

- Web-интерфейс управления - свитч имеет внутренний набор web-страниц, которые позволяют управлять им используя браузер с поддержкой java. Вы можете получить доступ к web- интерфейсу, используя:
- Станцию управления, подключенную к сети.
- Станцию управления, подключенную к порту управления свитча, используя межсетевой протокол для последовательного канала (Serial Line Internet Protocol).

Интерфейс управления командной строкой - свитч имеет интерфейс управления командной строкой, который позволяет вам ограничено управлять свитчем. Вы можете получить доступ к интерфейсу управления командной строкой используя:

- Терминал или эмуляцию терминала, подключенную к сети используя Telnet.
- Терминал или эмуляцию терминала, подключенный к порту управления свитча.
- Простой протокол сетевого управления (Simple Network Management Protocol) - вы можете управлять вашим свитчём используя любое приложение для сетевого управления, использующее SNMP, такое как 3Com Transcend Enterprise Manager software.

Методы управления свитчем

Вы можете управлять свитчем используя три метода:

- Web интерфейс управления - свитч имеет внутренний набор web страниц, которые позволяют управлять им, используя браузер с

поддержкой java. Вы можете получить доступ к web интерфейсу используя web браузер.

- Интерфейс управления командной строкой - ваш свитч имеет интерфейс управления командной строкой, который позволяет вам ограничено управлять свитчем.
- Простой протокол сетевого управления (Simple Network Management Protocol) - вы можете управлять вашим свитчем используя любое приложение для сетевого управления, использующее SNMP.

Настройка интерфейса web-управления

Вы можете использовать web-интерфейс при помощи:

- Станции управления, подключенную к сети.
- Станции управления, подключенную к порту управления свитча, используя межсетевой протокол для последовательного канала (Serial Line Internet Protocol).

Если несколько пользователей используют web-интерфейс одновременно, для многих пользователей время ответа может увеличено время ответа web-страницы и тогда появится сообщение «В документе нет данных». Поэтому рекомендуется позволять доступ к интерфейсу только 3-ем пользователям одновременно.

Настройка интерфейса управления командной строкой

Вы можете получить доступ к интерфейсу командной строки используя:

- Терминал или эмуляцию терминала, подключенную к порту управления свитча напрямую или через модем.
- Терминал или эмуляцию терминала, подключенную к свитчу через сети при использовании Telnet.

Настройка SNMP управления

Любое приложение для управления сетью, поддерживающее SNMP может управлять свитчем если:

- Корректные MIBs (Management Information Base) установлены на управляющей станции.
- Управляющая станция подключена к свитчу используя порт VLAN 1

Управление свитчем через сеть

Когда свитч управляется через сеть, то IP информация свитча должен быть настроена следующим образом.

- IP адрес
- Маска подсети

IP адреса

Для правильной работы каждое устройство в вашей сети должно иметь уникальный IP адрес. IP адрес имеет формат X.X.X.X где X - целое число в диапазоне от 0 до 255. Например, 192.168.100.8

IP адрес можно разделить на две части:

- Первая 192.168 определяет сеть, которой принадлежит это устройство.
- Вторая часть 100.8 определяет устройство в сети.

Если сеть вашей организации - внутренняя, то вы можете использовать любой произвольный IP адрес.

Система гарантирует, что каждый IP адрес уникален. Если у вас есть незарегистрированный IP адрес, и вы можете использовать идентичный адрес для другого устройства, то ваша сеть будет работать неправильно.

Подсети и использование маски подсети

Вы можете разделить вашу сеть на подсети. Поддержка подсетей важна потому, что количество битов, определяющих устройство в сети ограничивает количество устройств, используемых в сети. Для примера класс адресов С зарегистрирован на 254 устройства.

Если у вас сеть меньше 254 устройств, то вы можете не использовать подсети.

Маска подсети используется, чтобы разделить часть определяющее устройство на две следующие части:

- Первая часть определяет номер подсети.
- Вторая часть определяет устройство в подсети.

Регистрация пользователя по умолчанию

Если вы управляете свитчем через web-интерфейс или консоль, то вам нужно получить доступ к нему используя имя пользователя и пароль. Свитч содержит 4 имени пользователя по умолчанию, и для каж-

дого имени свой пароль и уровень привилегий.

Таблица 4.3. Типы пользователей

Имя пользователя	Пароль по умолчанию	Уровень привилегий
monitor	monitor	Пользователь может наблюдать, но не менять управление.
manager	manager	Может изменять оперативные параметры, но не специальные параметры.
security	security	Полный доступ
admin	нет	Полный доступ

Для предотвращения несанкционированного доступа к свитчу измените пароли по умолчанию.

Работа с интерфейсом командной строки

Меню интерфейса

Меню интерфейса командной строки показано на рис.4.1.

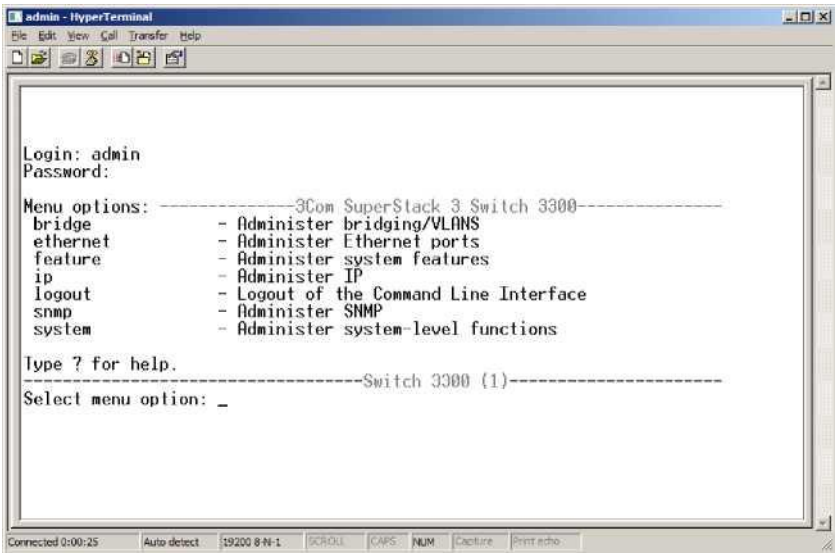


Рис.4.1. Меню командной строки

Интерфейс командной строки разделён на две области:

- Область меню - меню команд (для настройки свитча или перехода в другое меню), каждая команда содержит описание.
- Область команд - после Select menu option вы можете ввести свою команду.

Из верхнего меню вы можете получить доступ к 6 подменю.

Bridge menu - меню содержащее команды для администрирования функций свитча, таких как STP, широковещательная фильтрация и VLANs.

Ethernet menu - меню, позволяющее оперировать портами свитча и отображать их статистику.

Feature menu - это меню содержит команды настройки Roving Analysis Port включения или отключения контроля широковещательного шторма, установки или удаления гибких связей и настройки транкования на свитче.

IP menu - меню содержит средства работы с IP настройками, позволяет пинговать другие устройства и возвращать настройки по умолчанию.

Logout - выход текущего пользователя из командной строки

SNMP menu - содержит команды для изменения SNMP информации для группы.

System menu - команды для изменения информации о свитче и группе

Работа с MAC адресами

Отобразить MAC адреса порта Команда list в Address menu.

1. В меню верхнего уровня введите bridge port address list. Затем появится строка Select bridge ports (1-12, all):
2. Введите номер порта или all для просмотра MAC адресов порта.

Удаление MAC адресов из порта

Команда remove в Address menu.

1. В меню верхнего уровня введите bridge port address remove За-

тем появится строка Enter the address to be removed:

2. Введите адрес, который вы хотите удалить (08-00-02-06-03-bd).

Затем появится строка Enter the VLAN ID for this address (1-4094) [1]:

3. Введите VLAN ID для удаляемого адреса.

Отображение информации о порте

Команда detail в Port menu.

1. В меню верхнего уровня введите bridge port detail Затем появится строка Select bridge port (1-12):

2. Введите номер порта. Если выбранный порт работает в режиме VLT, то будет показано какой VLAN принадлежит порты с тэгированием vlt.

Unit 2, Port State: StpCoat: VLAN Membera VLAN ID	1 Detailed Disabled 19 hip Local ID	Informarion fwdTranaiticns: BroadcastStormControl:	0 Enabled
		Vian Name	Tagging
1	1	Default VLAN	None
Select menu	opt ion:		

Рис.4.2. Информация о порте

Работа с портами свитча

Отображение и изменение настроек порта

Вы можете просматривать и изменять настройки портов при помощи команд Ethernet меню. Эти команды позволяют вам:

- Изменять статус порта (по умолчанию - включено) • Задавать скорость режим дуплекса.
- Изменять режим авто определения.
- Включение или отключение контроля потоков IEEE 802.3x
- Отображать статистику портов свитча.
- Отображать краткую информацию о портах свитча. При работе со свитчем используйте команду unit.

Включение и отключение порта

По умолчанию все порты свитча включены.

1. Введите:
   ethernet portState Затем появится
   Select Ethernet port(s) (1-24):
2. Введите номер порта
   Затем появится
   Enter new value (enable, disable) [enable]:
3. Введите
   enable or disable.

Определение скорости и режима дуплекса

1. Введите: ethernet portMode Затем появится
   Select Ethernet port(s) (1-24):
2. Введите номер порта. Затем появится Если порт:

- 10BASE-T/100BASE-TX, то введите значение (10half,10full,100half,100full):
- 100BASE-FX, то введите значение (100half,100full):
- 10BASE-T, то введите значение (10half,10full):
- Если вы выбрали All, то все возможные режимы.

3. Введите новую скорость и режим дуплекса порта

Порты, которые поддерживают изменения, поменяют настройки.

Для связи без ошибок оба конца связи должны иметь одинаковые режимы дуплекса.

Пока авто определение не включено изменения не вступят в силу.

Включение и отключение автоопределения

Позволяет автоматически устанавливать скорость и режим дуплекса для витой пары.

- Для портов 10BASE-T/100BASE-TX скорость и дуплекс.
- Для портов 10BASE-T дуплекс.

Режим дуплекса не определяется если на другом конце связи нет функции авто определения. Поэтому по умолчанию все порты - полу-дуплекс.

1. Введите:
   ethernet autoNegotiation Затем появится
   Select Ethernet port (1-24, all):

2. Введите номер порта или all

Затем появится

Enter new value (enable,disable) [enable]:

3. Введите

enable or disable.

Оптоволоконные порты и порты модуля тринсмитера не авто опре-деляемые.

Включение и выключение контроля потоков

Контроль потоков IEEE 802.3x предотвращает перегрузку порта, работающего в полном дуплексе.

1. Введите:

ethernet flowControl Затем появится

Select Ethernet port (1-24, all):

2. Введите номер порта или all

Затем появится

Введите значение (on,off) [off]:

3. Введите on или off.

Отображение статистики порта

1. Введите:

ethernet statistics Затем появится

Select Ethernet port (1-24):

2. Введите номер порта.

Port :	1	Port Speed:	10Mbps HD Auto
Received Stats		Transmit Stats	
Unicast Packets:	0	Unicast Packets:	50
Non Unicast Packets:	0	Non Unicast Packets:	18734
Octets:	0	Octets:	1397087
Fragments:	0	Collisions:	0
Errors			
Undersize:	0	Oversize	0
CRC Errors:	0	Jabbers	0
Packet Size Analysis			
64 Octets:	13752	256 to 511 Octets:	5

```
65 to 127 Octets:      4404    512 to 1023 Octets:      0
128 to 255 Octets:     623     1024 to 1518 Octets:     00
```

Рис.4.3. Статистика порта

Отображение общей информации о порте

1. Введите: ethernet summary. Затем появится
   Select Ethernet port (1-24, all):
2. Введите номер порта или all
   Затем появится

Port	State	P.x Packets	P.x Octets	Errors
1	Enabled	163542	65439864	4
2	Disabled	0	0	0
3	Enabled	639263	83636219	4
24	Enabled	645232	23142514	0

Рис.4.4. Общая информация о порте

Контроль широковещательного шторма

Включение и выключения контроля широковещательного шторма

1. Введите: feature broadcastStormControl Затем появится
   Enter new value (disable, enable) [disable]:
2. Введите
   enable or disable.
   Затем появится (если ввели enable)
   Enter rising threshold in pps (0-200000) [2976]:
3. Введите пороговое значение увеличения числа пакетов
   Затем появится
   Enter falling threshold in pps (0-200000) [1488]:
4. Введите пороговое значение уменьшения числа пакетов
   Затем появится
   Enter time period in seconds (10-60) [30]:
5. Введите временной интервал между 10 и 60 сек.

Временной интервал определяет промежуток времени после начала широковещательного шторма, когда активируется функция контроля широковещательного шторма. Минимальное время половина этого значения.

Настройки IP-протокола

Отображение и изменение IP информации

Команды меню IP menu. Позволяют вам:

- Определить IP и SLIP информацию для свитча.
- Отображение IP информации для свитча.
- Задать, когда свитч использует BOOTP.
- Пинговать другие устройства в вашей сети.

Пред работой с конкретным свитчем выберите его командой unit.

Определение IP и SLIP (Serial Line Internet Protocol) информации.

Если вы выполняете эту команду впервые, то терминал или его эмуляция должна быть соединена с портом управления свитча посредством нуль модемного кабеля.

1. Введите:
   ip interface define Затем появится
   Enter IP address [0.0.0.0]:
2. Введите нужный IP адрес.
   Затем появится
   Enter subnet mask [0.0.0.0]:
3. Введите маску подсети если нужно.
   Затем появится
   Enter default gateway [0.0.0.0]:
4. Введите IP адрес маршрутизатора (если есть).
   Затем появится
   Enter SLIP address [192.168.101.1]:
   SLIP нужен для работы с web интерфейсом.
5. Введите адрес SLIP, если нужно. Затем появится
   Enter SLIP subnet mask [255.255.255.0]:
6. Введите маску подсети SLIP если нужно

Отображение IP и SLIP информации

1. Введите:

ip interface display

Затем появится

```
IP address                    191.100.40.120
Subnet mask:                  255.255.0.0
Default gateway:              191.100.40.121
SLIP address:                 191.100.40.120
SLIP subnet mask              255.255.0.0
```

Рис.4.5. Информация о IP и SLIP

Пинг других устройств

1. Введите:

ip ping

Затем появится

Enter destination IP address:

2. Введите IP адрес устройства, которое пингуется.

Затем появится

Starting ping, resolution of displayed time is 10 millisec и response from 191.128.40.121: 3 router hops. time = 10ms В случае если устройство недоступно выводится следующее:

No answer from 191.128.40.121

Отображение административной информации свитча

1. Введите:

system display

Затем появится

```
3Com SuperStack 3                 Deve1opme nt
System Name:                      Wiring Closet, Floor 1
Location:                         System Administrator
Contact:

                                      2 days, 3 hours, 10 minutes
Time since reset:                     2.20
Operational Version:                  1
Hardware Version                      1.00
Boot Version:                         08:00:00:00:11:11
MAC Address:                          3C33000
Product No.                           7ZNR001111
Serial Number
```

	161.71.120.152
TFTP Server Address Filename	s2s02_50.bin
Last software upgrade	TFTP Access Violation

Рис.4.6. Административная информация

Управление свитчем с использованием Web-интерфейса

Получение доступа к web-интерфейсу

Получить доступ к web-интерфейсу можно через порт управления или сеть. Для доступа к web-интерфейсу через порт управления вы должны установить, настроить и запустить Serial Web Utility. Serial Web Utility нужна только когда вам нужно получить доступ к web- интерфейсу через порт управления и не требуется при доступе через сеть.

Для доступа к web интерфейсу через сеть, выполните следующее:

1. Убедитесь, что ваша сеть правильно настроена для управления через web интерфейс.
2. Откройте Web браузер.
3. В поле ввода адреса введите URL вашего свитча. (формат http:// xxx.xxx.xxx.xxx // xxx.xxx.xxx.xxx - IP адрес группы).

а. Когда ваш браузер обнаружит группу, появиться диалог ввода имени пользователя и пароля.

4. Введите ваше имя и пароль.

- Если вы уже имеете логин и пароль, то введите их.
- Если вы осуществляете доступ к web интерфейсу в первый раз, введите логин и пароль по умолчанию для подтверждения ваших полномочий. Если настраиваете свитч, то зайдите под логином admin.

Для предотвращения несанкционированного доступа немедленно поменяйте пароль.

После ввода правильного имени пользователя и пароля появится одно из двух:

- Если доступ в первый раз, то стартовая страница.
- Если вы уже получали доступ к web-интерфейсу раньше, то главный webинтерфейс.

Рекомендуется только единовременный доступ до 3х лиц к интерфейсу в одно и то же время.

Иконки управления

С левой стороны главного окна есть несколько иконок управления, которые позволяют вам открыть страницы управления свитчем.

С левой стороны главного окна есть несколько иконок управления, которые позволяют вам открыть страницы управления свитчем.

- **Management Settings -** Нажмите чтобы отобразить страницу настроек управления стеком.
- **Configuration -** Нажмите чтобы отобразить страницу конфигурации стека
- **Health -** Нажмите чтобы отобразить страницу состояния стека.
- **Unit-** Нажмите чтобы отобразить страницу настройки конкретного свитча из стека.

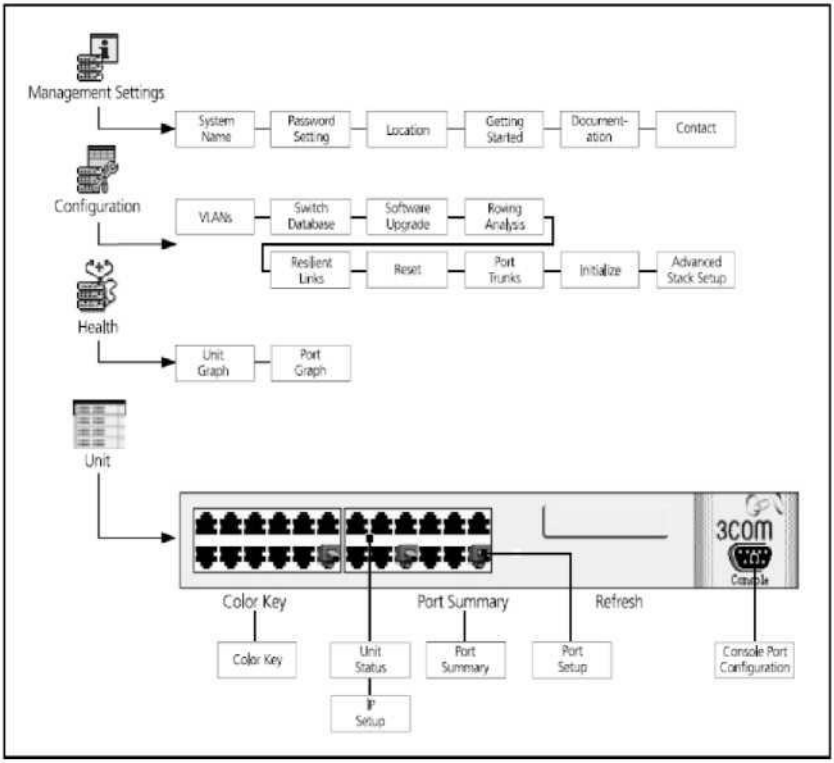


Рис.4.7. Диаграмма навигации

PAGE AREA

PAGE AREA - главный web-интерфейс, включающий страницы которые позволяют вам управлять группой. Web-страницы сгруппированы по 4-ем категориям:

- UNIT PAGES - страницы, позволяющие производить настройку конкретного свитча из группы или его портов.
- Switch Graphic - Эта страница содержит графическую информацию о свитче, которая показывает статус портов. Она всегда отображается поверх всех других страниц.
- Color Key - Эта страница дает возможность вам изучить закодированную цветом информацию, используемую на предыдущей странице.
- Port Summary - Эта страница дает возможность вам изучить скорость и режим дуплекса на портах, показанных на странице Switch Graphic.
- Unit Status - Эта страница дает возможность вам изучить главные детали управления свитчем.
- IP Setup - Эта страница дает возможность вам настроить параметры IP свитча.
- Port Setup - Эта страница дает возможность вам настроить порты свитча.
- Console Port Configuration - Эта страница дает возможность вам настроить порт управления свитча.
- Management Settings Pages - Эти страницы позволяют вам изменить настройки управления группой
- System Name - назначить имя группы.
- Password Setting - поменять пароль
- Location - описать физическое место нахождения свитча
- Getting Started - на начальную страницу
- Documentation - назначить местонахождение онлайн справки и

документации

- Contact - определить человека, к которому можно обращаться с вопросами о стеке.
- Configuration Pages
- VLAN Setup - настройка виртуальных сетей для стека
- Switch Database - Настройка базы данных свитча
- Software Upgrade - Обновление и управление ПО в стеке свитчей
- Roving Analysis Setup - Установка roving анализа на порты свитча
- Resilient Links - Установка гибких ссылок для группы
- Reset - позволяет сбросить настройки свитча.
- Port Trunks Setup - Установка транкования портов
- Initialize - Сброс на начальные настройки
- Advanced Stack Setup - настройка дополнительных особенностей свитча.
- Health Pages - статистика свитча.
- Unit Graph - разная статистика для портов свитча.
- Port Graph -статистика для данного порта свитча.

Настройка отдельного свитча

Его можно настроить используя **UNIT PAGES** .

Для отображение статуса портов - используйте **Switch Graphic.**

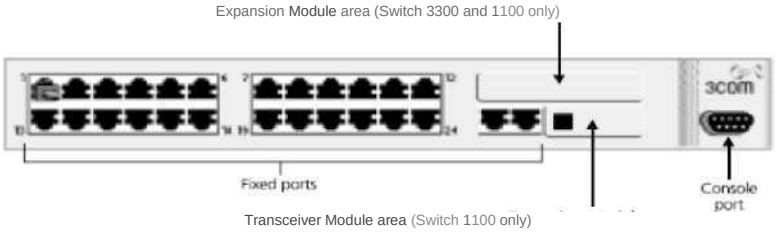


Рис.4.8. Статус портов

Цветовые коды при отображении статуса портов

- Зелёный доступен, подсоединен.

- Чёрный доступен, отсоединен.
- Серый (коннект) , недоступен, подсоединен
- Серый (нет коннекта) недоступен, отсоединен.

Просмотр скорости и режима дуплекса портов

Port Summary

Port	Speed	Duplex	Port	Speed	Duplex
1	100	Full	13	100	Full
2	100	Full	14	100	Full
3	100	Full	15	10	Half
4	100	Full	16	100	Full
5	10	Half	17	10	Half
6	100	Full	18	100	Full
7	100	Full	19	100	Full
8	10	Half	20	100	Full
9	10	Full	21	10	Half
10	10	Half	22	10	Full
11	10	Full	23	10	Half
12	10	Half	24	10	Full

Рис.4.9. Окно Port Summary

Отображение административных подробностей

Unit Status

System Name:	Switch 3300	Location:	
Contact:		Unit Description:	Switch 3300
Hardware Rev.	1	MAC Address:	08:00:4e:35:8c:4d
Software Version:	2.2.0	Boot PROM Version:	1.00
Product Number:	3C33000	TFTP Server:	161.71.120.152
Filename:	upgrade/s2s02_40.bin	Software Upgrade Status:	TFTP Access Violation
Unit UpTime:	16 Hrs 30 Mins 1 Secs		IP Setup

Рис.4.10. Окно Unit Status

Ввод IP информации

Вы можете ввести IP информацию используя **IP Setup**.

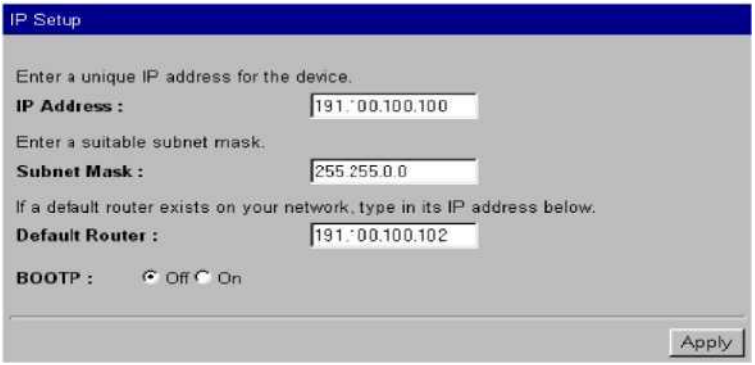


Рис.4.11. Окно IP Setup

Настройка порта при помощи Port Setup

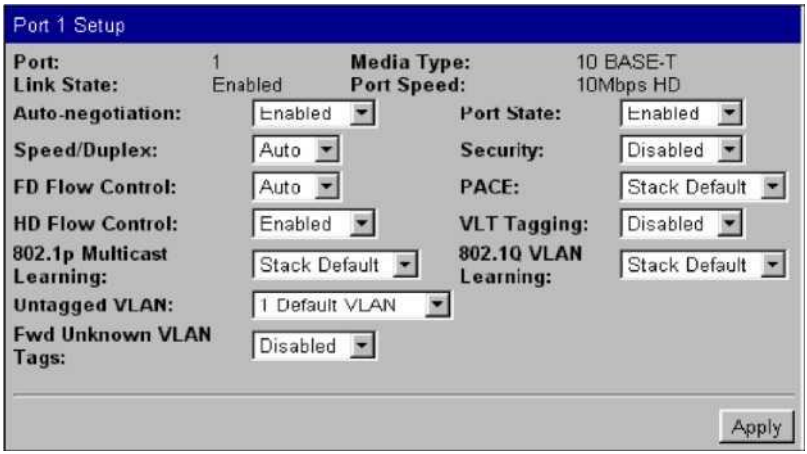


Рис.4.12. Окно Port Setup

Элементы окна Port Setup

Port - Номер выбранного порта.

Link State *Enabled / Disabled* - Состояние связи, подключённой к порту.

Media Type - Кабель, подключённый к порту.

Port Speed - Скорость порта и режим дуплекса. *FC* показывает, что включён контроль утечки.

Auto-negotiation *Enabled / Disabled (*только для витой пары) - Если

авто определение включено на портах 10BASE-T/100BASE-TX скорость и режим дуплекса связи автоматически определяется.

Если авто определение включено на портах 10BASE-T режим дуплекса связи автоматически определяется.

Если отключено, то устанавливаются вручную.

Если порт на другом конце связи не в режиме авто определения, то режим дуплекса не может быть определён. Поэтому порты свитча установлены в полу дуплекс.

Speed/Duplex *100Mbps FD / 100Mbps HD / 10Mbps FD / 10Mbps HD /Авто.* - Скорость и режим дуплекса. Для связи без ошибок оба конца связи должны иметь одинаковый режим дуплекса.

FD Flow Control *Enabled / Disabled /* - Если авто определение отключено, позволяет вам включать или отключать контроль потоков (для полного дуплекса). Позволяет предотвратить перегрузку порта.

HD Flow Control *Enabled / Disabled* - Контроль потоков при работе в режиме полу дуплекса.

802.1p Multicast Learning *Stack Default / Disabled* - Позволяет вам точно определять 802.1p широковещательную фильтрацию (GMPR) и фильтровать широковещательный трафик автоматически.

Untagged VLAN - Позволяет вам определять виртуальную сеть, которой принадлежат порты.

FWD Unknown VLAN Tags *Enabled / Disabled / Auto* - Позволяет определить когда порт направляет трафик который использует неизвестные IEEE 802.1Q тэги. Если IEEE 802.1Q learning выключено, то вы можете выбрать:

Enabled - Если порт другого устройства поддерживает IEEE 802.1Q VLANs.

Disabled - Если нет поддержки IEEE 802.1Q VLANs.

Port State *Enabled / Disabled* - Включить или выключить порт.

Security *Enabled / Disabled* - Включение безопасности, которая препятствует незаконному подсоединению пользователей к устройствам в сети. Когда защита включена, то порт переходит в режим одного адреса. В этом режиме он:

- Заносит все MAC адреса, хранящиеся для порта в БД свитча.

- Запоминает адрес первого пакета, полученного портом.
- Определяет адрес как постоянный вход.

После запоминания первого адреса:

- Порт недоступен для других адресов.
- Не один другой адрес не может быть запомнен, пока защита не отключена или первый адрес не удалён из базы данных вручную.
- Адрес не может быть запомнен для другого порта пока защита не отключена, или адрес не удалён из базы данных вручную.
- Вы можете активировать защиту для резервной связи или для транкования портов.

PACE *Stack Default / Enabled / Disabled* -Позволяет вам выбрать когда порт использует PACE (Приоритетный контроль доступа) для поддержки трафика мультимедиа.

Stack Default — Настройка берётся из страницы настройки группы.

Enabled — Если порт подключен:

К хабу, мосту или маршрутизатору, которые не имеют PACE или PACE отключён.

На конечной станции, на которой PACE включён.

Disabled — Если порт подключен:

К хабу

К хабу, мосту или маршрутизатору, на которых включён PACE.

К конечной станции, которая не имеет PACE или PACE отключён.

VLT Tagging *Enabled / Disabled* - Позволяет вам использовать VLT (Virtual LAN Trunk) (Магистраль Виртуальная сеть) тэгирование. Если оба конца связи поддерживают VLT, то вы можете создать связь VLT t, которая несет трафик для всех Виртуальная сеть, определенных для вашего свитча.

802.1Q VLAN Learning *Stack Default / Disabled* - Позволяет вам настроить, когда порт использует IEEE 802.1Q запоминание (GVRP) для определения портов в виртуальной сети автоматически:

Stack Default —Порт экспортирует настройки 802.1Q VLAN из Advanced Stack Setup

Disable — Порт не использует IEEE 802.Q learning. Если другой конец связи не поддерживает IEEE 802.Q.

Настройка БД свитча

Вы можете настроить БД группы используя страницу Switch Database

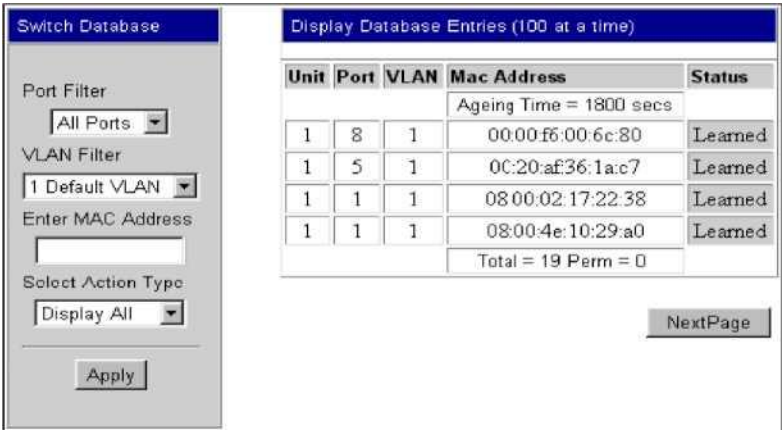


Рис.4.13. Окно Switch Database

Отображение статистики для каждого свитча

Для этого вам потребуется страница Health. Эта страница позволяет вам:

- Просматривать статистику для всех портов свитча.
- Просматривать статистику порта свитча.

Отображение статистики свитча

Для этого воспользуйтесь страницей Unit Graph.

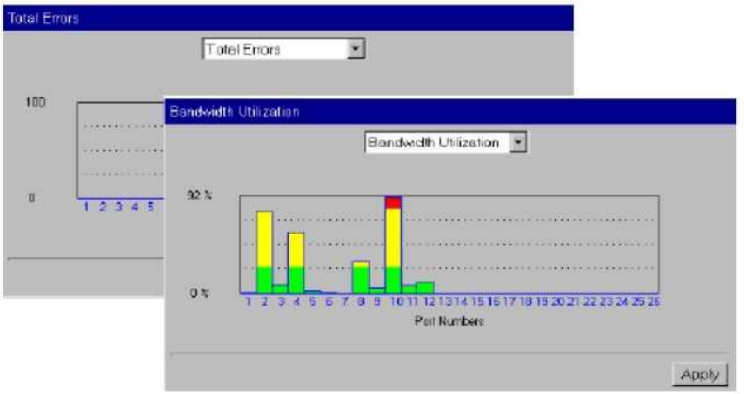


Рис.4.14. Окно Unit Graph.

Для просмотра количества ошибок или загрузки полосы пропускания выберите

1. *Bandwidth Utilization*. /Total Errors
2. Нажмите *Apply*.

Интерпретация статистики.

- Диаграмма загрузки полосы пропускания отображает загрузку за время 30 сек.
- Зелёный - от 0 до 25% Небольшая загрузка сети.
- Жёлтый 26-85% Нормальная загрузка сети.
- Красный 86-100% ! ! ! Ошибка связи или неправильные параметры сети.
- Диаграмма ошибок показывает количество ошибок за 30 сек.

Отображения статистики портов

Для этого откройте страницу Port Graph.

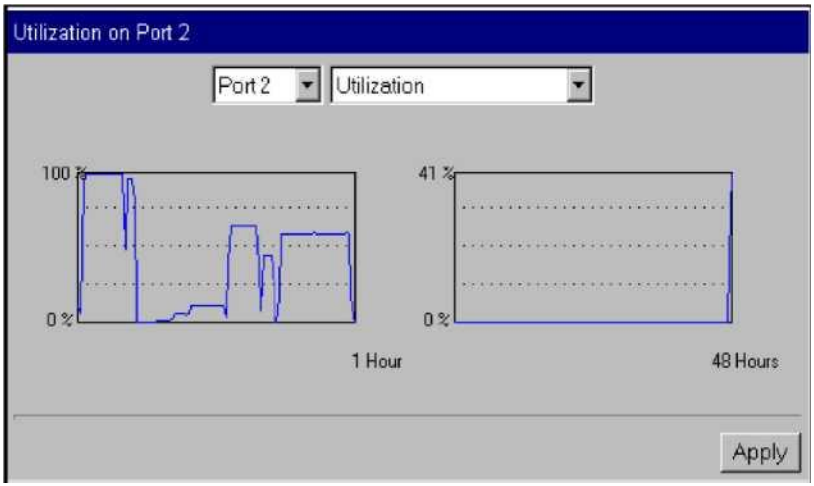


Рис.4.15. Статистика загрузки

Для просмотра статистики загрузки, ошибок или размера пакетов:

1. Выберите порт
2. Выберите *Utilization, Total Errors* or *Packet Size*
3. Нажмите Apply

• Диаграмма загрузки отображает загрузку за время 30 сек.

- Зелёный - от 0 до 25% Небольшая загрузка сети.
- Жёлтый 26-85% Нормальная загрузка сети.
- Красный 86-100% ! ! ! Ошибка связи или неправильные параметры сети.

• Диаграмма ошибок показывает количество ошибок за 1 час или за последние 48 часов.

• Размер пакетов - зависимость пакетов различных размеров получения портом.

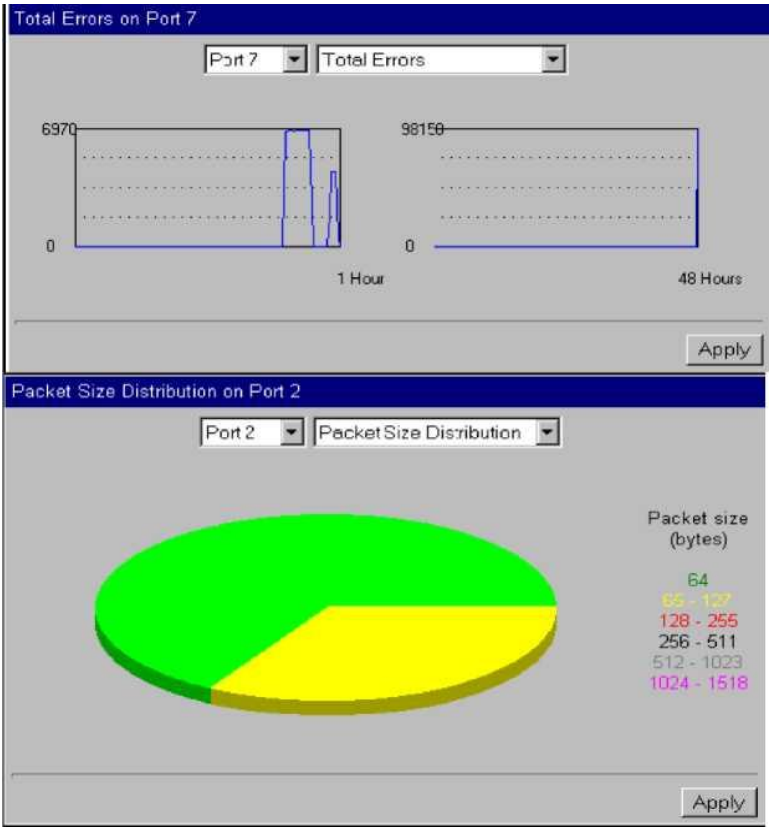


Рис.4.16. Статистики ошибок и размера пакетов

Свитч содержит ПО, которое позволяет менять и наблюдать режимы его работы. Это ПО не требует функционирования на свитче, но если вы это сделаете, то вы можете улучшить эффективность свитча, и кроме того, улучшить в целом производительность вашей сети.

Виртуальные локальные сети

Кроме своего основного назначения — повышения пропускной способности связей в сети — коммутатор позволяет локализовать потоки информации в сети, а также контролировать эти потоки и управлять ими, опираясь на механизм пользовательских фильтров. Однако пользовательский фильтр может запретить передачи кадров только по конкретным адресам, а широковещательный трафик он передает всем сегментам сети. Так требует алгоритм работы моста, который реализован в коммутаторе, поэтому сети, созданные на основе мостов и коммутаторов, иногда называют плоскими — из-за отсутствия барьеров на пути широковещательного трафика.

Технология виртуальных локальных сетей (Virtual LAN, VLAN), которая появилась несколько лет тому назад в коммутаторах, позволяет преодолеть указанное ограничение. Виртуальной сетью называется группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов. Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна, независимо от типа адреса — уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра. Виртуальные сети могут пересекаться, если один или несколько компьютеров входят в состав более чем одной виртуальной сети.

Говорят, что виртуальная сеть образует домен широковещательного трафика (broadcast domain), по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

При использовании технологии виртуальных сетей в коммутаторах одновременно решаются две задачи:

• повышение производительности в каждой из виртуальных сетей, так как коммутатор передает кадры в такой сети только узлу назначения;

• изоляция сетей друг от друга для управления правами доступа пользователей и создания защитных барьеров на пути широковещательных штормов.

Для связи виртуальных сетей в общую сеть требуется привлечение сетевого уровня. Он может быть реализован в отдельном маршрутизаторе, а может работать и в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством — так называемым коммутатором 3-го уровня.

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм группирования в сети портов коммутатора (рис. 4.17). При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко — пропадает эффект полной изоляции сетей.

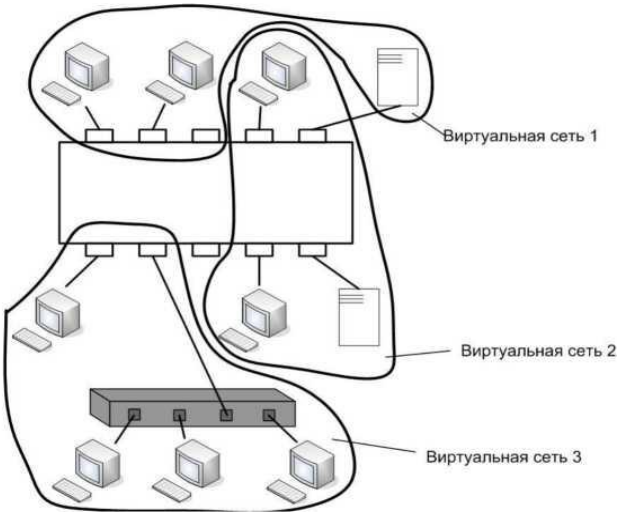


Рис. 4.17. Виртуальные сети, построенные на одном коммутаторе.

Второй способ образования виртуальных сетей основан на группировании MAC-адресов. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует выполнения большого количества ручных операций от администратора. Однако он оказывается более гибким при построении виртуальных сетей на основе нескольких коммутаторов, чем способ группирования портов.

Резервные связи

Резервные связи позволяют защитить важные связи и предотвратить простой сети если эта связь оборвется. Резервные связь состоит из основной и резервной связи. Резервные связь настраивается путем определения основного и резервного порта на обоих концах связи.

При нормальной работе основной порт работает, а резервный отключён. Если основной отключается, то резервный активируется.
Если основной порт восстановился, то вы можете отключить резервный.

Так как использование резервных связей в концентраторах определено только в стандарте FDDI, то для остальных стандартов разработчики концентраторов поддерживают такую функцию с помощью своих частных решений.

Если по какой-либо причине порт отключается (срабатывает механизм автосегментации), концентратор делает активным его резервный порт.

Для автоматического поддержания резервных связей в сложных сетях в коммутаторах реализуется алгоритм покрывающего дерева — Spanning Tree Algorithm. Этот алгоритм основан на периодической генерации служебных кадров, с помощью которых выявляются и блокируются петлевидные связи в сети.

НАСТРОЙКА РЕЗЕРВНЫХ СВЯЗЕЙ И ВИРТУАЛЬНЫХ СЕТЕЙ

Настройка с использованием Web-интерфейса

Получить доступ к web-интерфейсу можно через порт управления или сеть. Основное меню web-интерфейса выглядит следующим образом.

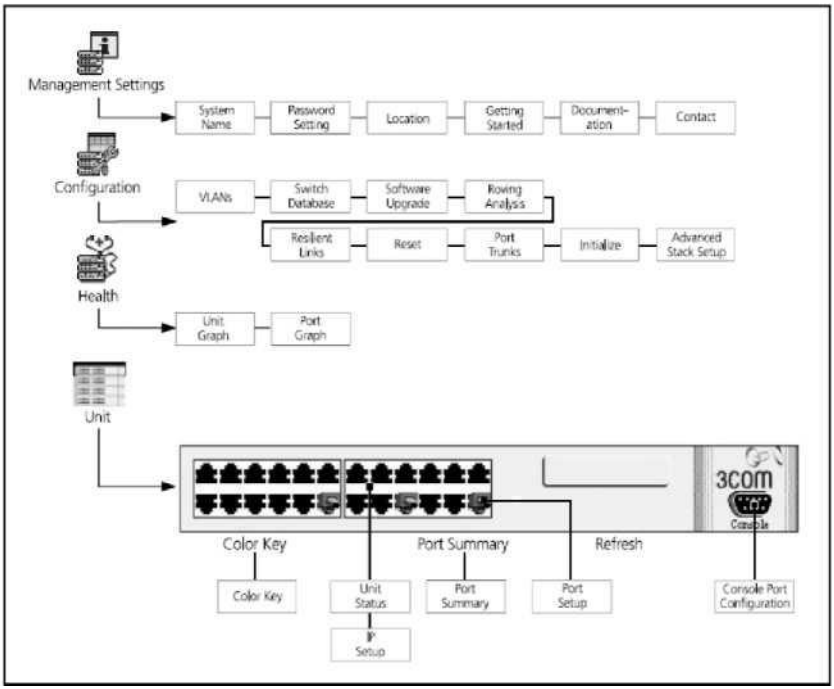


Рис.4.18. Диаграмма навигации

Настройка стека

Вы можете настроить стек, используя страницы настроек. Эти страницы позволяют:

- НастроитьБД свитчей в стеке.
- Настроитьдополнительные параметры стека.
- Настроитьрезервные связи стека.
- Настроить транкование стека.
- Настроить виртуальную сеть стека.

- Настроить roving анализ портоввстеке.
- Перезагрузить свитчи в стеке.
- Обновить управляющее ПО для свитчей в стеке.

Если к существующей группе с включённым *STAP* подсоединен свитч на котором *STAP* отключён, то настройки нового свитча конфликтуют с настройками группы. Подобный исход возможен и при настройках контроля широковещательного шторма. Для преодоления этой проблемы вам надо заранее настроить свитч или настроить группу заранее.

Установка резервной связи

Воспользуйтесь страницей Resilient Links. Создавать резервные связи можно также и через консоль.

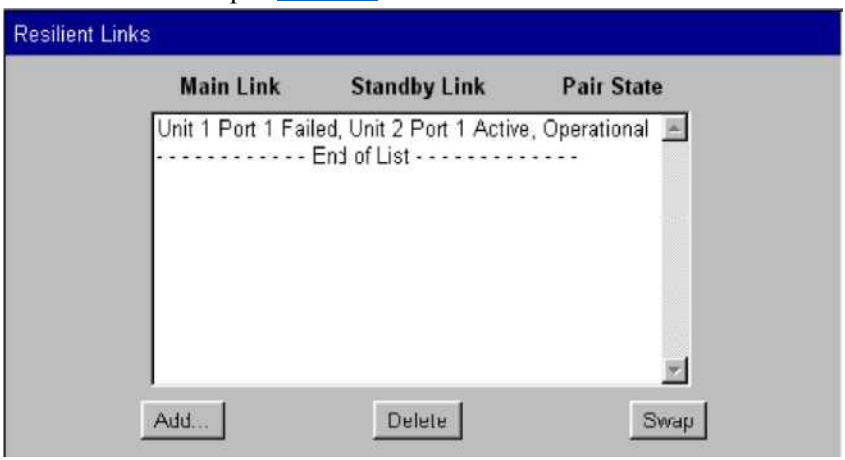


Рис.4.19. Окно Resilient Links

При установке резервных связей помните:

- Резервные связи нельзя устанавливать, если свитч поддерживает STP.
- Резервные связи не могут быть установлены на оптоволоконный или витой пары порт. Основной и резервный должны быть одного типа.
- Резервные связь должна быть определена с двух концов связи.
- Резервная связь устанавливается только тогда, когда:

- Порты принадлежат к одной виртуальной сети.
- Порты используют одну и туже систему виртуальную сеть тэгирования. (802.1Q tagging or VLT tagging)
- Порты имеют одинаковые настройки IEEE 802.1Q VLAN learning.
- Порты имеют одинаковые настройки IEEE 802.1p multicast learning.
- Ни на одном из портов не включена защита
- Ни один из портов не является транкованным.
- Ни один из портов не принадлежит другой резервной связи.

• Состояние портов резервной связи нельзя изменить пока не произойдёт ошибка связи.

Отображение пар резервной связи

Страница Resilient Links показывает пары резервных связей, для группы:

Main Link Unit 1 Port 1 / Unit 1 Port 2 / ... показывает какой порт свитча основной, и его статус.

Standby Link Unit 1 Port 1 / Unit 1 Port 2 / ... показывает какой порт свитча резервный, и его статус.

Pair State *Operational / Not Operational* отображает состояние связи.

Создание резервной связи

Нажмите кнопку ADD, появится страница добавления связи. Выберите свитч, на котором будут основной и резервный порт. Нажмите NEXT.

Из Main link field, выберите основной порт.

Нажмите NEXT.

Из Standby link field, выберите резервный порт.

Нажмите NEXT. Появиться страница с новой связью.

Версия 2.6x ПО для управления свитчём (после 2000г), будет содержать дополнительные опции для гибких связей.

После выбора резервного порта нажмите NEXT. Появится окно режима Свитча. Из выпадающего списка выберите:

Symmetric, если вы хотите передавать через резервную даже после включения основной.

Switchback, автоматически восстановить передачу через основную в случае восстановления ее.

Удаление резервных связей

1. Выберите гибкую связь.
2. Нажмите Delete.

Замена основного порта резервной связи

1. Выберите гибкую связь.
2. Нажмите Swap.

Настройка виртуальной сети

Виртуальные сети можно создавать и настраивать как через web-интерфейс, так и через консоль.

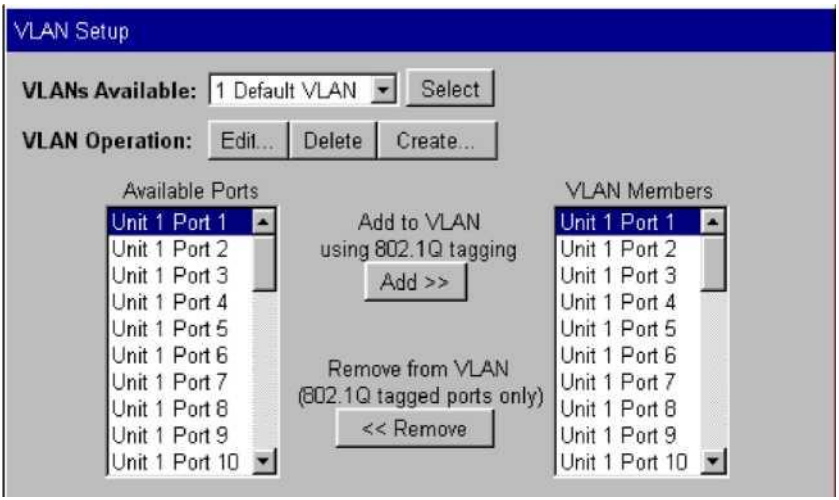


Рис.4.20. Окно VLAN Setup

Настройка параметров виртуальной сети

Страница VLAN Setup позволяет вам определить настройки ВС. Для этого:

1. Нажмите *Creat.* Появится страница Create VLAN.
2. В *VLAN Name,* введите имя сети. Имя может быть до 32 символов.
3. В *802.1Q VLAN ID,* введите уникальный 802.1Q идентификатор

VLAN. 802.1Q ID используется для определения VLAN если вы используете 802.1Q тэгирование в вашей сети, и это может быть любой номер между 2 и 4094. Его надо ввести если вы намереваетесь использовать 802.1Q тэгирование.

4. В *Local ID* listbox, введите локальный ID VLAN. Локальный ID используется для определения VLAN в группе, и может быть номером между 2 и 16 (VLAN1 - по умолчанию).

5. Нажмите *Apply*. Настройки VLAN определены, и страница VLAN Setup отобразит порты, принадлежащие новой VLAN.

Редактирование настроек VLAN

Страница VLAN Setup позволяет вам редактировать настройки любой VLAN:

Из *VLANs Available* listbox, выберите VLAN.

1. Нажмите *Select* button.
2. Нажмите *Edit ....* Страница редактирования настроек VLAN отобразиться.
3. Исправьте требуемую информацию.
4. Нажмите *Apply*. Настройки VLAN определены, и страница VLAN Setup отобразит порты, принадлежащие VLAN.

Удаление VLAN

1. Из *VLANs Available* listbox, выберите VLAN.
2. Нажмите *Select* button.
3. Нажмите *Delete*.

Вы не можете удалить VLAN если к ней присоединены порты.

Отображение портов присоединенных к VLAN

1. Из *VLANs Available* listbox, выберите VLAN.
2. Нажмите *Select* button.

Помещение портов в VLAN используя 802.1Q тэгирование

1. Из *VLANs Available* listbox, выберите VLAN.
2. Нажмите *Select* button.
3. Выберите нужный порт в Available Porsts listbox.
4. Нажмите Add.

Настройка VLAN с использованием интерфейса командной строки

Из основного меню необходимо сделать переход bridge -> vlan

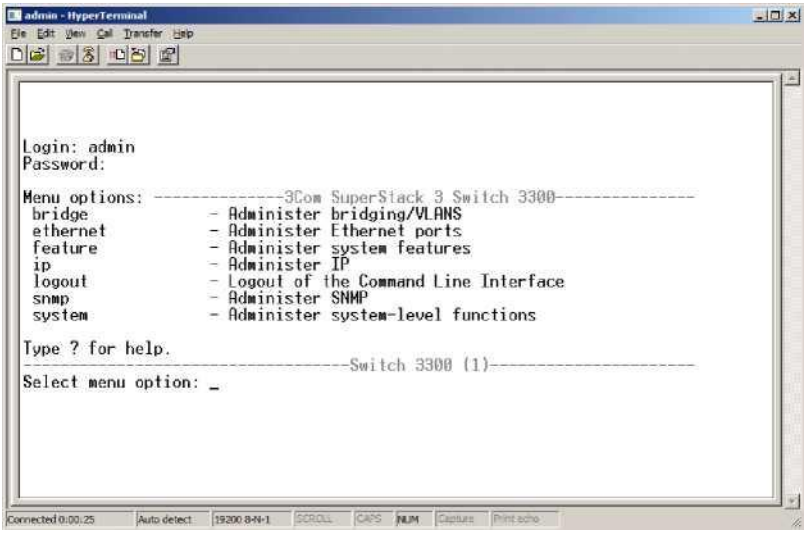


Рис.4.21. Меню командной строки

Создание VLAN

1. Команда create в VLAN menu.
2. В меню верхнего уровня введите bridge vlan create Затем появится строка Enter VLAN ID (1-4094) [1]:
3. Введите номер виртуальной сети, к которую вы создаете. По умолчанию это последний номер VLAN в стеке.
4. Затем появится строка
5. Enter Local ID (1-16) [3]:
6. Введите локальный ID связанный с VLAN. По умолчанию это последний ID VLAN в стеке +1
7. Затем появится строка
8. Enter VLAN Name [VLAN 3]:
9. Введите имя VLAN (макс 32 символа). По умолчанию это VLAN x где x - ID VLAN.

Добавление порта в VLAN

Команда addPort в VLAN menu.

1. В меню верхнего уровня введите **bridge vlan addPort** Затем

появится строка Select VLAN ID (1-4094) [1]:

2. Введите номер виртуальной сети, к которой добавляете порт.
3. Затем появится строка Enter port (1-12, all):
4. Введите номер порта.
5. Затем появится строка Enter tag type (none, 802.1Q):
6. Введите метод тэгирования порта

Удаление VLAN

1. В меню верхнего уровня введите bridge vlan delete Затем появится строка Select VLAN ID (2-4094) :
2. Введите номер виртуальной сети, которую вы удаляете.

Отображение детальной информации о VLAN

1. В меню верхнего уровня введите bridge vlan detail. Затем появится строка Select VLAN ID (2-4094) [1]:
2. Введите номер виртуальной сети, которую вы просматриваете.

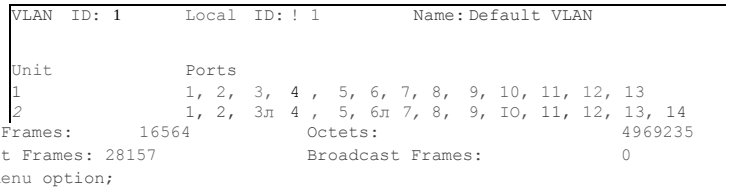


Рис.4.22. Информация о VLAN

Изменение VLAN

1. В меню верхнего уровня введите bridge vlan modify Затем появится строка Select VLAN ID (2-4094) [1]:
2. Введите номер виртуальной сети, которую вы модифицируете. Затем появится строка Enter VLAN Name [VLAN 3]:
3. Введите новое имя сети.

Удаление порта из VLAN

1. В меню верхнего уровня введите bridge vlan removePort Затем появится строка Select VLAN ID (2-4094) [1]:
2. Введите номер Виртуальная сеть, из которой вы удаляете порт.
3. Затем появится строка
4. Select port (1,5,7, all):
5. Введите номер удаляемого порта или all.

Отображение общей информации о VLAN

1. В меню верхнего уровня введите bridge vlan summary
2. Затем появится строка Select VLAN ID (2-4094) [1]:
3. Введите номер Виртуальная сеть, которую хотите просмотреть или all.

```
VLAN ID         Local ID Name
1               1               Default VLAN
Select menu option:
```

Рис.4.23. Общая иформация о VLAN

Резервные связи

Установка резервных связей

1. Введите: feature resilience define. Затем появится
2. Select unit for main link (1-4):
3. Введите номер свитча для основной связи. Затем появится
4. Select port for the main link (1,2,7):
5. Введите номер порта для основной связи
6. Затем появится
7. Select unit for standby link (1-4):
8. Введите номер свитча для резервной связи. Затем появится
9. Select port for the standby link (3,6):
10. Введите номер порта для резервной связи.

Отображение информации о резервных связях

1. Введите:
   feature resilience detail
2. Затем появится

```
    Index   Main  LinkState       Standby Link State Active Link Pair State
1           Unit   1 Port 1   Failed Unit 2 Port 1 Failed Standby Operational
2           Unit   1 Port 6   Failed Unit 2 Port 6 Active Standby Operational
    Select menu option:
```

Рис.4.24. Информация о резервных связях

Работа с портами свитча

Отображение и изменение настроек порта

- Вы можете просматривать и изменять настройки портов при

помощи команд Ethernet меню. Эти команды позволяют вам:

- Изменять статус порта (по умолчанию - включено)
- Задавать скорость режим дуплекса.
- Изменять режим авто определения.
- Включение или отключение контроля потоков IEEE 802.3x
- Отображать статистику портов свитча.
- Отображать краткую информацию о портах свитча. При работе со свитчем используйте команду unit.

Включение и отключение порта

По умолчанию все порты свитча включены.

Введите:

1. ethernet portState Затем появится
2. Select Ethernet port(s) (1-24):
3. Введите номер порта
4. Затем появится
5. Enter new value (enable, disable) [enable]:
6. Введите
7. enable or disable.

СОЗДАНИЕ СЕТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ VIRTUAL LOCAL AREA NETWORK В CISCO PACKET TRACER

Рассмотрим две схемы создания VLAN в Cisco Packet Tracer.

1. **Схема с одним коммутатором.**

Для этого выполнинm следующие действия:

- Создать VLAN;
- Определить Access порты.

Схема с одним коммутатором:

1. Запустить Cisco Packet Tracer;
2. Добавить коммутатор 2960;
3. Добавить 4 компьютера;
4. Соединить прямым кабелем каждый компьютер с коммутатором;

5. Пусть компьютеры PC0, PC1 принадлежат одному сегменту (например, технологи). А PC2 и PC3 принадлежат второму сегменту (например, менеджеры). Выделить каждый сегмент своим цветом. Для этого выбрать функцию Draw и выделить каждый сегмент (например, эллипсом) своим цветом (рис. 4.25);

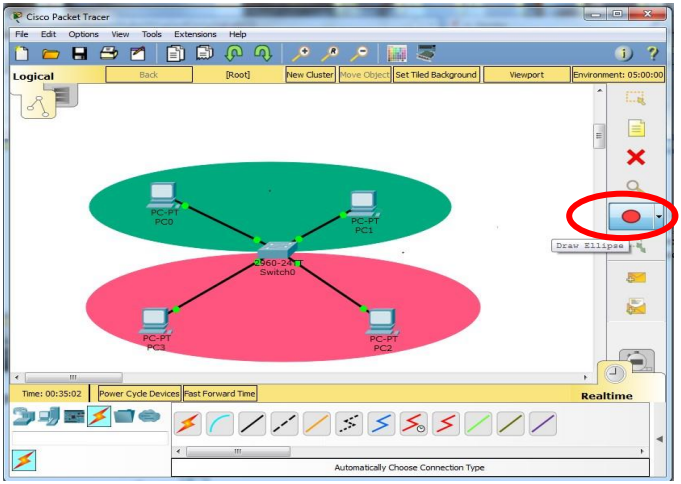


Рис 4.25. Схема с одним коммутатором

6. Зайти в настройки коммутатора (вкладка CLI). Войти в привилегированный режим, режим глобального конфигурирования:

(В Terminal зайти в привилегированный режим с помощью команды enable:

Switch>enable

Перед настройкой необходимо войти в режим «глобального конфигурирования» с помощью команды configure terminal.

Switch#configure terminal

6.1. На данном этапе необходимо определить VLAN, в котором будут находится данные пользователи. По умолчанию все порты коммутатора находятся в VLAN1, возникает необходимость пере-

определения в другой. Для этого необходимо создатьVLAN2 (команда VLAN 2) и дать имя vlan2 (команда name technologi). Выйти из режима VLAN (CTRL Z);

(ИЛИ воспользоваться закладкой Switch-Config-VLAN Database)

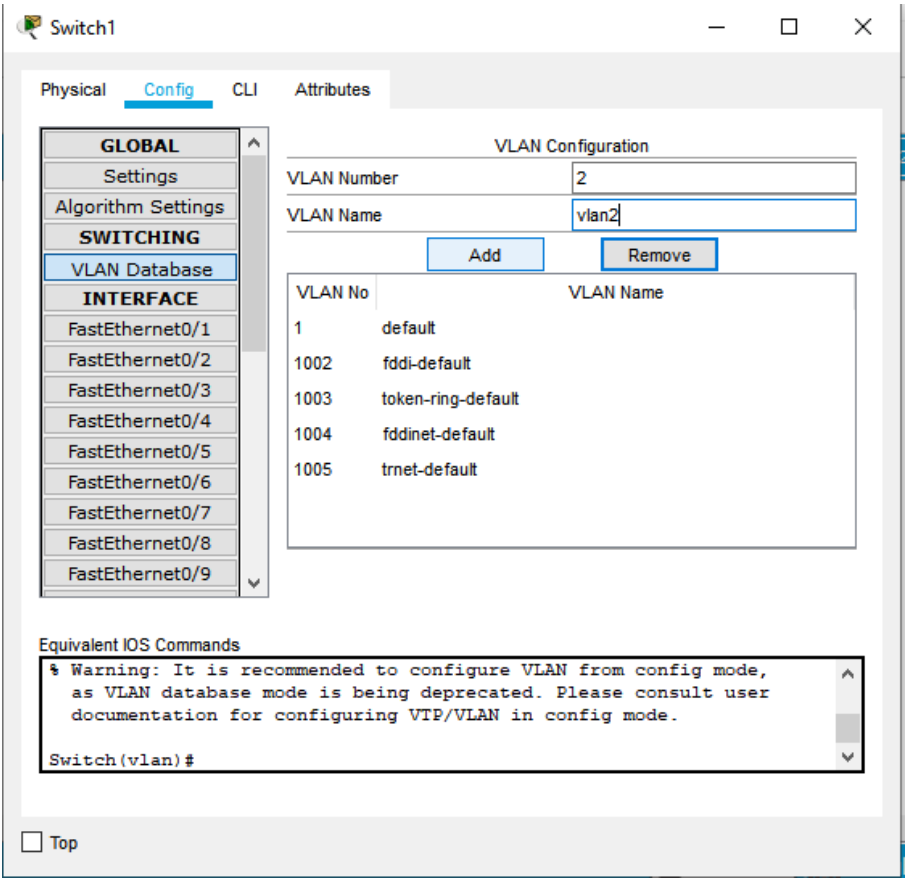


Рис. 4.26. Закладка Switch-Config-VLAN Database

6.2. Теперь надо настроить интерфейс. Поскольку PC0 подключен к порту Fast Ethernet0/1, а PC2 к порту Fast Ethernet0/2, данные порты необходимо определить в только что созданный VLAN2. Для

этого зайти в настройки интерфейса Fast Ethernet0/1 с помощью команды interface Fast Ethernet 0/1.

Определить, что данный порт функционирует в режиме Access (команда switchport mode access), и определить VLAN2 (команда switchport access VLAN 2). Аналогично настроить порт Fast Ethernet0/2.

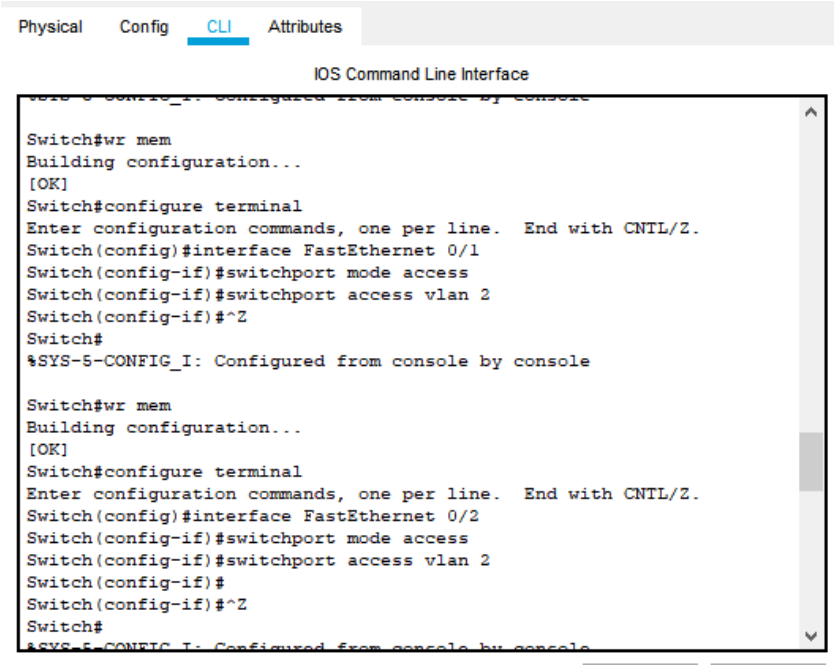


Рис. 4.27. Определение VLAN2

Выхйти из режима конфигурирования. Проделанную работу можно проверить с помощью команды show VLAN или show VLAN brief. Из рис. 4.28 можно увидеть, что порты Fast Ethernet 0/1 и Fast Ethernet 0/2 определены в VLAN2;

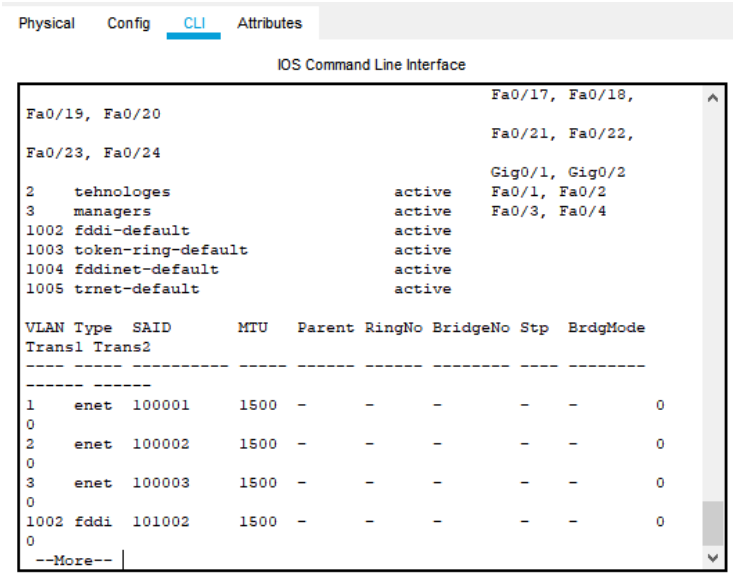


Рис. 4.28. Настройка портов fastEthernet0/1 и fastEthernet0/2

6.3. Проделать аналогичные действия для сегмента в VLAN 3 с названием vlan3.

6.4. Теперь необходимо задать IP адреса (например, для PC0 задать 192.168.2.1, для PC1 задать 192.168.2.2, для PC2 задать 192.168.3.2, для PC3 задать 192.168.3.1).

6.5. Провести проверку. Зайти в Command Prompt для сегмента (VLAN2).

Набрать ping 192.168.2.2. Аналогичные действия провести со вторым сегментом.

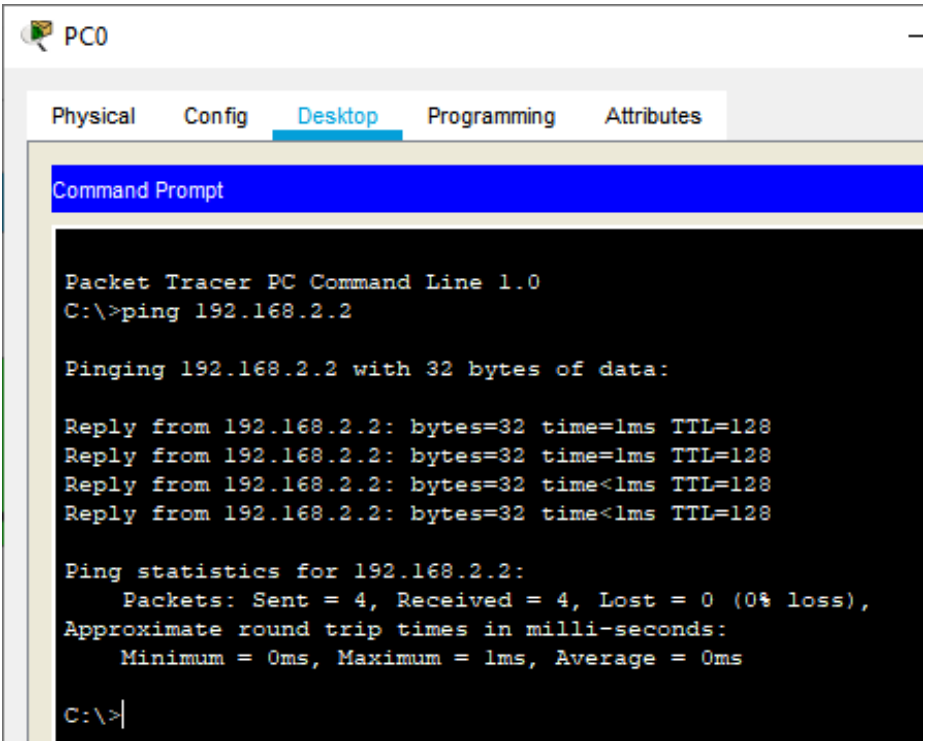


Рис.4.29 Проверка работоспособности сети

2. **Схема с двумя коммутаторами. Для этого необходмо выполнить следующие действия:**

- Создать VLAN;
- Определить Access порты;
- Определить Trunk порты.

Схема с двумя коммутаторами:

1. В схеме с одним коммутатором создать еще одну сеть, состоящую еще из одного коммутатора и 4 компьютеров, соединить два коммутатора перекрестным кабелем к портам GigabitEthernet (рис. 4.30). Для удобства можно скопировать первую сеть.

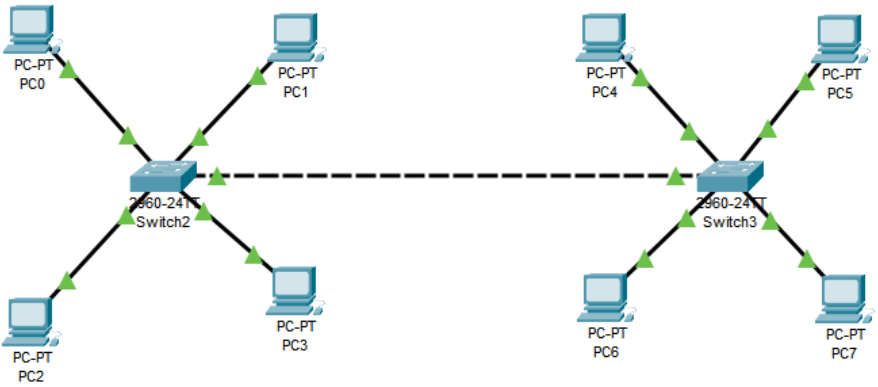


Рис. 4.30. Схема с двумя коммутаторами

2. Добавить IP адреса, для компьютеров PC4 – 192.168.2.3, PC5 – 192.168.2.4, PC6 – 192.168.3.3, PC7 – 192.168.3.4. И объединить их в два сегмента vlan2 и vlan3;

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface FastEthernet0/1
Switch(config-if)#interface mode access
                   ^
% Invalid input detected at '^' marker.

Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#interface FastEthernet0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#interface FastEthernet0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#interface FastEthernet0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#^Z
Switch#
%SYS-5-CONFIG_I: Configured from console by console
wr mem
```

Рис. 4.31. Объединение в два сегмента vlan2 и vlan3

3. Настройки для коммутатора сохранены.

4. Настроить Trunk порт для первого коммутатора (Switch1).

4.1. Режим конфигурирования. Набирать команду interface gigabitEthernet 0/1, switchport mode trunk. Указать VLAN2, которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 2

4.2. Trunk порт для второго коммутатора (Switch2). Режим конфигурирования. Набрать команду interface gigabitEthernet 0/1, switchport mode trunk. Указать VLAN2, которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 2

5. Соединить два коммутатора вторым перекрестным кабелем interface gigabitEthernet 0/2

5.1. Режим конфигурирования. Набрасть команду interface gigabitEthernet 0/2, switchport mode trunk. Указать VLAN3, которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 3

5.2. Trunk порт для второго коммутатора (Switch2). Режим конфигурирования. Набрать команду interface gigabitEthernet 0/2, switchport mode trunk. Указать VLAN3, которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 3

В результате получается следующую схему:

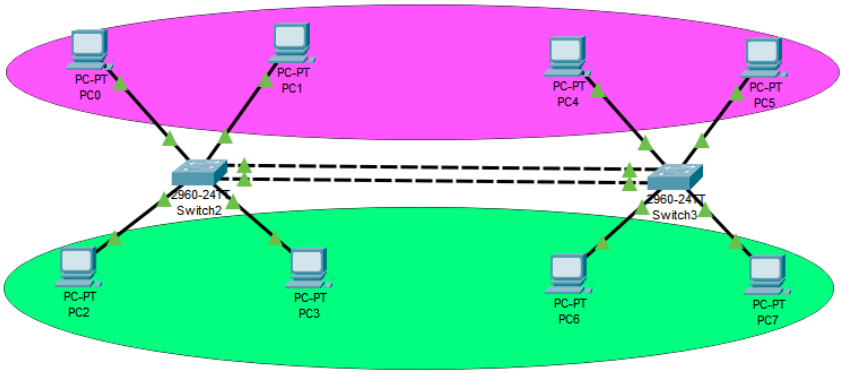


Рис. 4.32. Итоговая сеть

Примечание: В качестве режима работы порта вместо Trunk можно использовать Acces. Основное отличие заключается в том, что

- **Trunk порт** может пропускать тегированный трафик нескольких Vlan (пункт 6). Поэтому может быть использован один кабель.
- **Access порт** будет пропускать нетегированный трафик, принадлежащий только одному Vlan. Команды настройки портов обоих коммутаторов будут: Interface GigabitEthernet0/1, switchport mode access, switchport access vlan 2 (используется два кабеля).

6. **Схема с одним перекрестным кабелем**

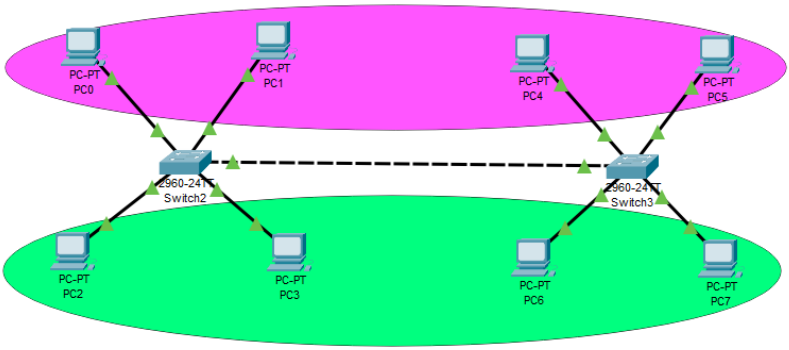


Рис. 4.33. Схема с одним перекрестным кабелем

Необходимо настроить trunk порты коммутаторов:

- Trunk порт для первого коммутатора (Switch1). Режим конфигурирования. Набрать команду interface gigabitEthernet 0/1, switchport mode trunk. Указать VLAN, через которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 2,3
- Trunk порт для второго коммутатора (Switch2). Режим конфигурирования. Набрать команду interface gigabitEthernet 0/1, switchport mode trunk. Указать VLAN, через которые будут передаваться через данное физическое соединение с помощью команды switchport trunk allowed vlan 2,3

7. Проверить взаимодействие данных компьютеров различных сегментов.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. В Cisco Packet Tracer собрать схему сети, используя коммутаторы 2960

VLAN 2

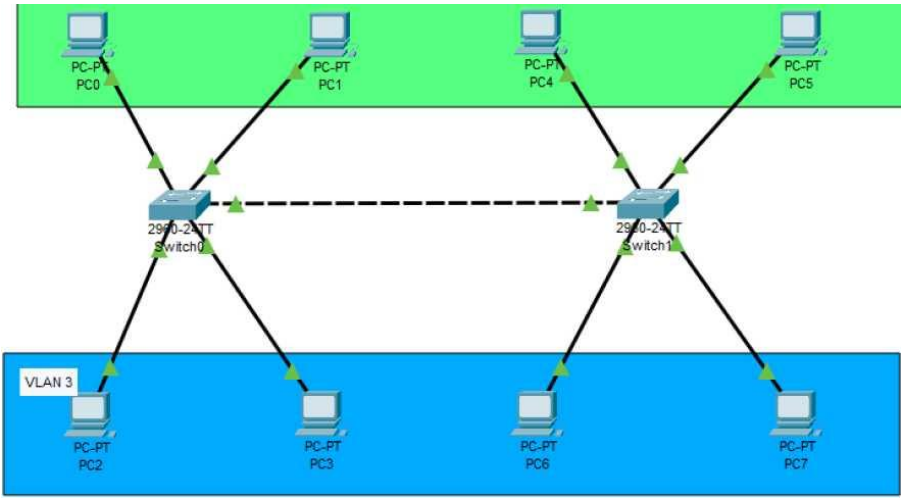


Рис. 4.34. Схема с сети

2. Выполнить следующие действия:
- Создать две VLAN;
- Настроить access порты;
- Настроить trunk порты.

3. Для адресации использовать следующую схему:
- VLAN 2 - 10.X.2.0, где X - номер варианта (по журналу);
- VLAN 3 - 10.X.3.0, где X - номер варианта (по журналу).

4. В режиме симуляции проверить доступ между ПК.
5. Ответить на контрольные вопросы и оформить отчет.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ, ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ

1. Сохраненные в файлах результаты создания сети в формате Cisco Packet Tracer (с расширением pkt)
2. Подготовленный отчет.

Отчет на защиту предоставляется в электронном или печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы со скриншотами, выводы.

В отчете МАКСИМАЛЬНО ПОДРОБНО отобразить этапы настройки и проверки работы VLANов с помощью скриншотов и комментариев.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Опишите назначение резервных связей.
2. Раскройте значение термина управляющее ПО.
3. Дайте определение виртуальным сетям.
4. Перечислите основные задачи виртуальных сетей
5. Приведите алгоритм удаления VLAN.
6. Перечислите способы образования виртуальных сетей.
7. Опишите роль Spanning Tree Algorithm.
8. Перечислите возможные настройки стека.
9. Перечислите и опишите возможные настройки портов.
10. Сформулируете постулаты, при которых устанавливаются резервные связи.

11. Приведите механизм работы виртуальных сетей.
12. Приведите алгоритм установки резервной связи.
13. Перечислите параметры настройки виртуальной сети.
14. Перечислите недостатки пользовательского фильтра.
15. Опишите роль MAC - адресов в образовании виртуальных сетей.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Сергеев, А.Н. Основы локальных компьютерных сетей [Электронный ресурс]: учебное пособие / А.Н. Сергеев. — Санкт-Петербург : Лань, 2016. — 184 с. – Режим доступа: URL: https://e.lanbook.com/book/87591
2. Топорков, С.С. Компьютерные сети для продвинутых пользователей [Электронный ресурс]: учебное пособие / С.С. Топорков. — Москва : ДМК Пресс, 2009. — 192 с. – Режим доступа: URL: https://e.lanbook.com/book/1170

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Ачилов Р.Н. Построение защищенных корпоративных сетей [Электронный ресурс]: учебное пособие / Р.Н.Ачилов. – Москва : ДМК Пресс, 2013. – 250 с. Режим доступа: URL: https://e.lanbook.com/book/66472
2. Ибе О. Компьютерные сети и службы удаленного доступа [Электронный ресурс]: справочник / О.Ибе. - – Москва : ДМК Пресс, 2007. – 336 с. Режим доступа: URL: https://e.lanbook.com/book/1169

ЭЛЕКТРОННЫЕ РЕСУРСЫ:

1. Научная электронная библиотека http://eLIBRARY.RU
2. Электронно-библиотечная система http://e.lanbook.com
3. Компьютерные сети и технологии http://www.xnets.ru