ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО БЮДЖЕТНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
МОСКОВСКОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА СВЯЗИ И ИНФОРМАТИКИ

Ткачук Е.О.

Администрирование сетевых устройств в инфокоммуникационных системах

Методическое пособие по выполнению лабораторных работ и практических занятий

Ростов-на-Дону
2019

УДК 681.5037.26

Ткачук Е.О.

Администрирование сетевых устройств в инфокоммуникационных системах. Методическое пособие к практическим занятиям и по выполнению лабораторных работ и практичексих занятий. / Моск. техн. ун-т связи и информатики, Сев.-Кавк. филиал. – Ростов н/Д, 2019, 149 с.

В пособии даются организационно-методические указания к лабораторным вопросам и порядок выполнения и оформления лабораторных работ.

Предназначено для студентов обеих форм обучения, изучающих дисциплину «Администрирование сетевых устройств в инфокоммуникационных системах», а также может быть полезно всем остальным студентам, желающим самостоятельно современные сетевые программные технологии.

Рецензент канд. техн. наук, доц. А.Н. Чикалов (СКФ МТУСИ)

Обсуждено и утверждено на заседании кафедры ИВТ (протокол заседания кафедры №1 от 26.08.2019).

Оглавление

Практическое занятие 1. Установка, настройка и конфигурирование виртуальной машины

Цель работы:
- Приобретение навыков установки и создания виртуальных машин в Oracle VM VirtualBox
- Приобретение навыков установки и начальной настройки операционной системы Ubuntu

Задание:

1. Скачать или получить у преподавателя образ установочного диска ОС Ubuntu
2. Создать виртуальную машину в VirtualBox.
3. Установить ОС Ubuntu на созданную виртуальную машину.
4. В установленной операционной системе:

-установить дополнения гостевой ОС
-настроить рабочие столы (эффекты, изображения);
-изменить раскладку клавиатуры по умолчанию;
-определить тип сеанса загружаемую по умолчанию.

5. Ознакомиться и описать панель инструментов Ubuntu.

Краткие теоретические сведения

Занятия по дисциплине «Операционные системы проводятся с использованием технологии виртуализации.

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере: при том каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор.

Благодаря использованию данной технологии у студента появляется возможность безопасно работать с различными гостевыми операционными системами, не подвергая опасности настройку хостовой операционной системы компьютера учебного класса.

В качества средства виртуализации будем использовать программное средство Oracle VM VirtualBox.

VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других. Использование виртуальной машины (ВМ) на домашнем ПК, прежде всего, позволит одновременно запускать несколько операционных систем (гостевые ОС).

К примеру, в данный момент на компьютере или ноутбуке установлен один из выпусков операционной системы Microsoft Windows (хостовая ОС). Установка же виртуальной машины, в данном случае VirtualBox, позволяет использовать в среде хостовой ОС любые другие системы (гостевые), включая macOS, Linux, Android, Windows и так далее, вариантов здесь может быть очень много.

Программа была создана компанией Innotek с использованием исходного кода Qemu. Первая публично доступная версия VirtualBox появилась 15 января 2007 года. В феврале 2008 года Innotek был приобретён компанией Sun Microsystems, модель распространения VirtualBox при этом не изменилась. В январе 2010 года Sun Microsystems была поглощена корпорацией Oracle, модель распространения осталась прежней.

Ключевые возможности

Кроссплатформенность

Модульность

Поддержка USB 2.0, когда устройства хост-машины становятся доступными для гостевых операционных систем (только в проприетарной версии)

Поддержка 64-битных гостевых систем(начиная с версии 2.0), даже на 32-битных хост-системах (начиная с версии 2.1, для этого обязательна поддержка технологии виртуализации процессором)

Поддержка SMP на стороне гостевой системы (начиная с версии 3.0, для этого обязательна поддержка технологии виртуализации процессором)

Встроенный RDP-сервер, а также поддержка клиентских USB-устройств поверх протокола RDP (только в проприетарной версии)

Экспериментальная поддержка аппаратного 3D-ускорения (OpenGL, DirectX 8/9 (с использованием кода wine) (только в 32-битных Windows XP, Vista, 7 и 8, для гостевых DOS / Windows 3.x / 95 / 98 / ME поддержка аппаратного 3D-ускорения не предусмотрена)

Поддержка образов жёстких дисков VMDK (VMware) и VHD (Microsoft Virtual PC), включая snapshots (начиная с версии 2.1)

Поддержка iSCSI (только в проприетарной версии)

Поддержка виртуализации аудиоустройств (эмуляция AC97 или SoundBlaster 16 или Intel HD Audio на выбор)

Поддержка различных видов сетевого взаимодействия (NAT, Host Networking via Bridged, Internal)

Поддержка цепочки сохраненных состояний виртуальной машины (snapshots), к которым может быть произведён откат из любого состояния гостевой системы

Поддержка Shared Folders для простого обмена файлами между хостовой и гостевой системами (для гостевых систем Windows 2000 и новее, Linux и Solaris)[6]

Поддержка интеграции рабочих столов (seamless mode) хостовой и гостевой операционной системой

Поддержка формата OVF/OVA

Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс)

Базовая версия полностью открыта по лицензии GNU GPL, соответственно нет ограничений в использовании

Пакет дополнений

VirtualBox Guest Additions — комплект программного обеспечения, устанавливаемый в гостевую операционную систему и расширяющий её возможности по взаимодействию с системой виртуализации и хост-системой. Например, после установки специального драйвера «виртуальной видеокарты» становится возможным изменять разрешение рабочего стола гостевой ОС произвольным образом вслед за размером окна VirtualBox, в котором запущена виртуальная машина.

До версии 4.0.0 существовало две версии, различавшиеся по лицензии и функциональности. Начиная с 4.0.0 закрытые компоненты вынесены в отдельный пакет дополнений (Extension Pack):

Пакет дополнений содержит закрытые компоненты и распространяется под проприетарной лицензией PUEL (бесплатно только в персональных целях или для ознакомления):

RDP сервер — позволяет подключаться к виртуальной системе удалённо с помощью любого RDP-совместимого клиента;

Поддержка USB — позволяет передавать виртуальной машине USB устройства;

Intel PXE boot ROM — загрузка операционной системы по сети. Используется для создания тонких клиентов/бездисковых рабочих станций.

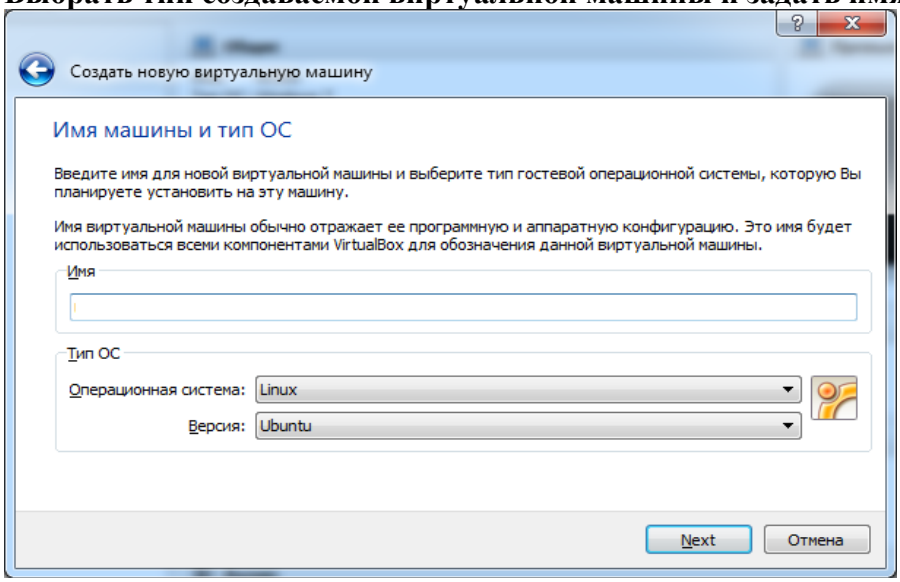
Ход работы:

Установка операционной системы Ubuntu производиться на виртуальную машину Oracle VM VirtualBox. Сначала необходимо установить Oracle VM VirtualBox и создать виртуальную машину.

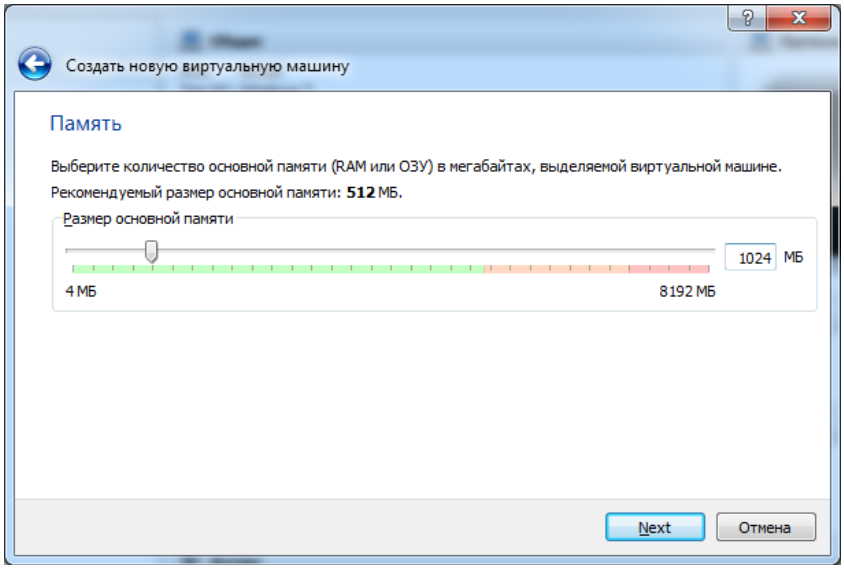
Установка Oracle VM VirtualBox производиться с настройками по умолчанию. После установки виртуальной машины установить пакет расширения (Oracle_VM_VirtualBox_Extension_Pack).

1)Создание виртуальной машины:

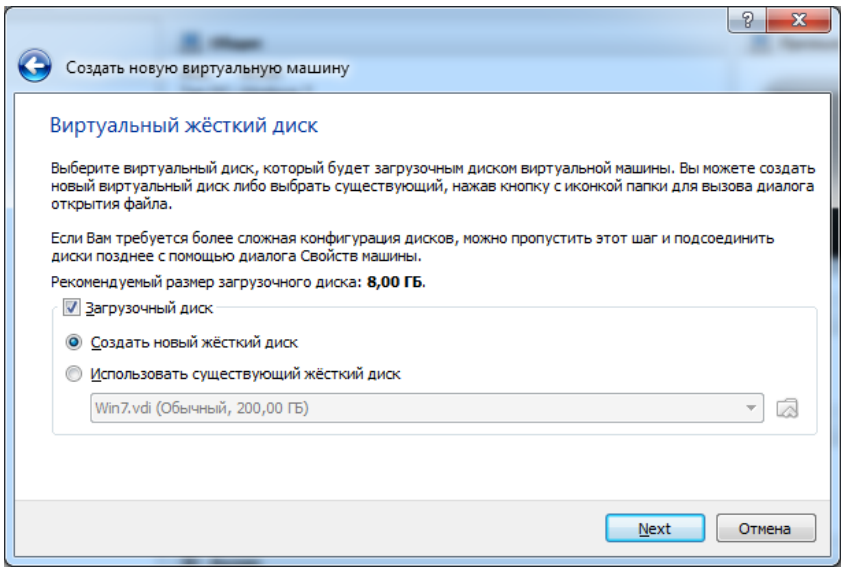
Выбрать тип создаваемой виртуальной машины и задать имя.



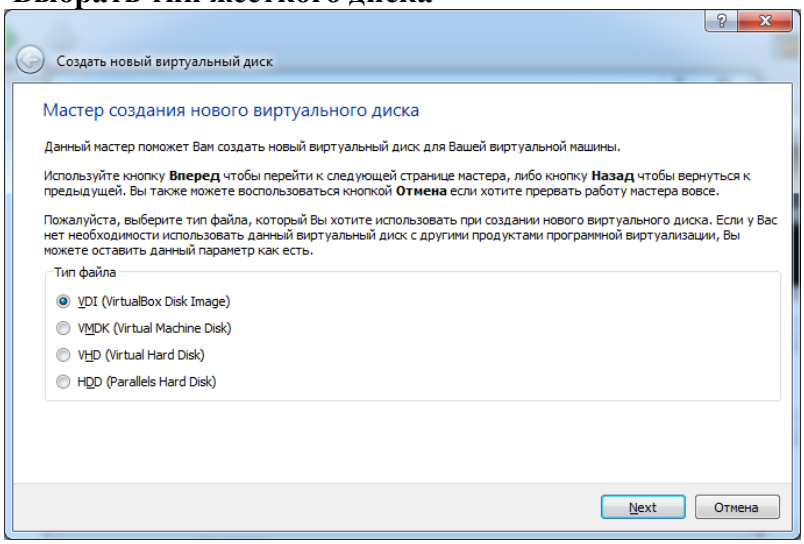
Установить объем памяти 1024 Мб



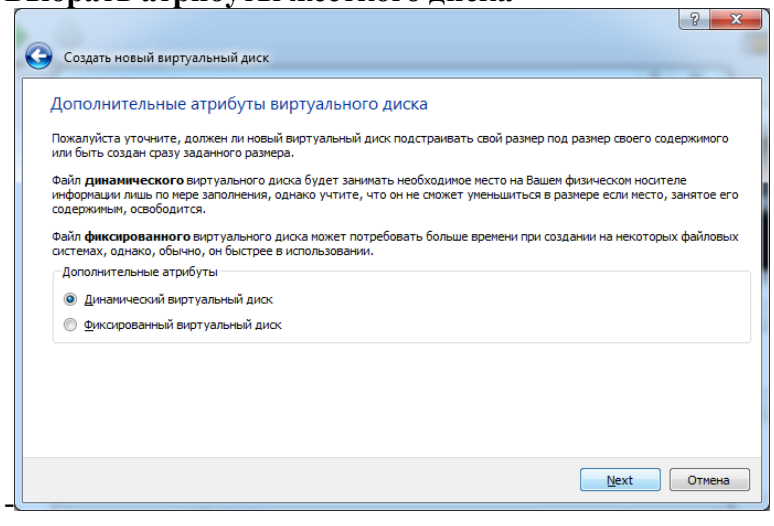
Создать новый жесткий диск



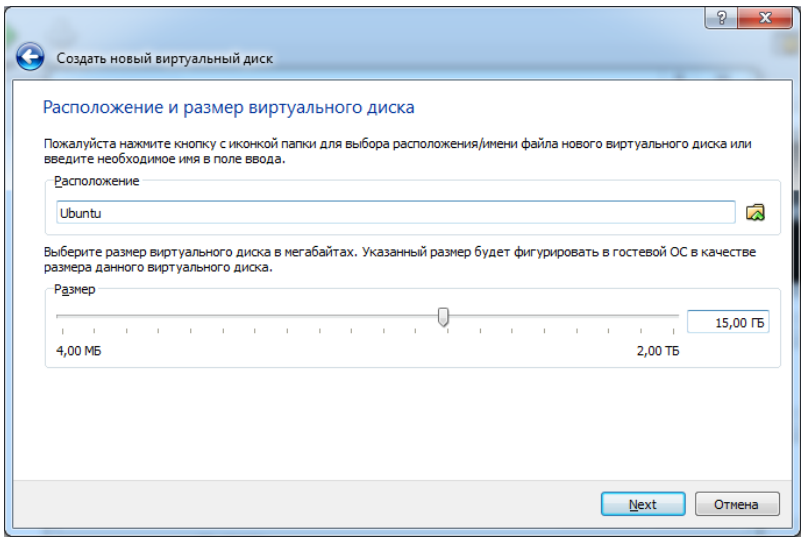
Выбрать тип жесткого диска



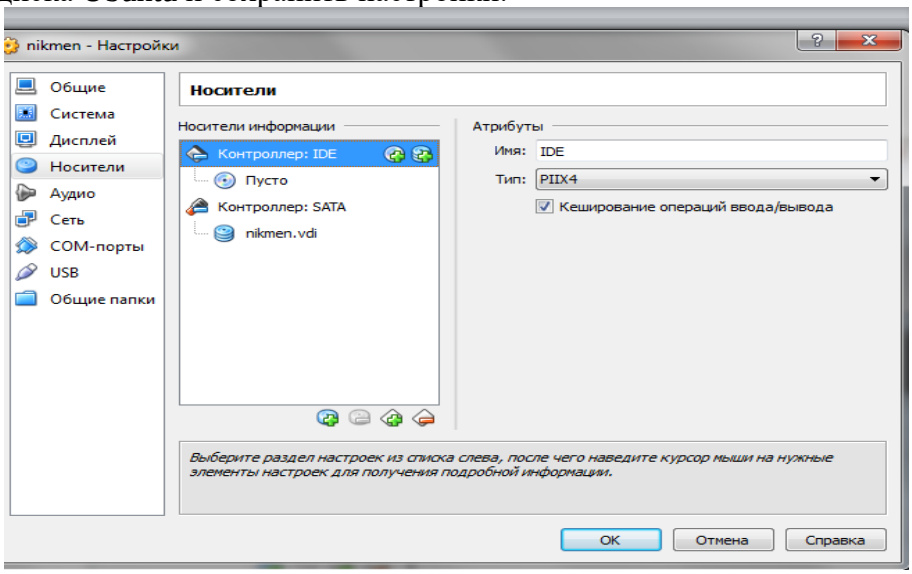
Выбрать атрибуты жесткого диска



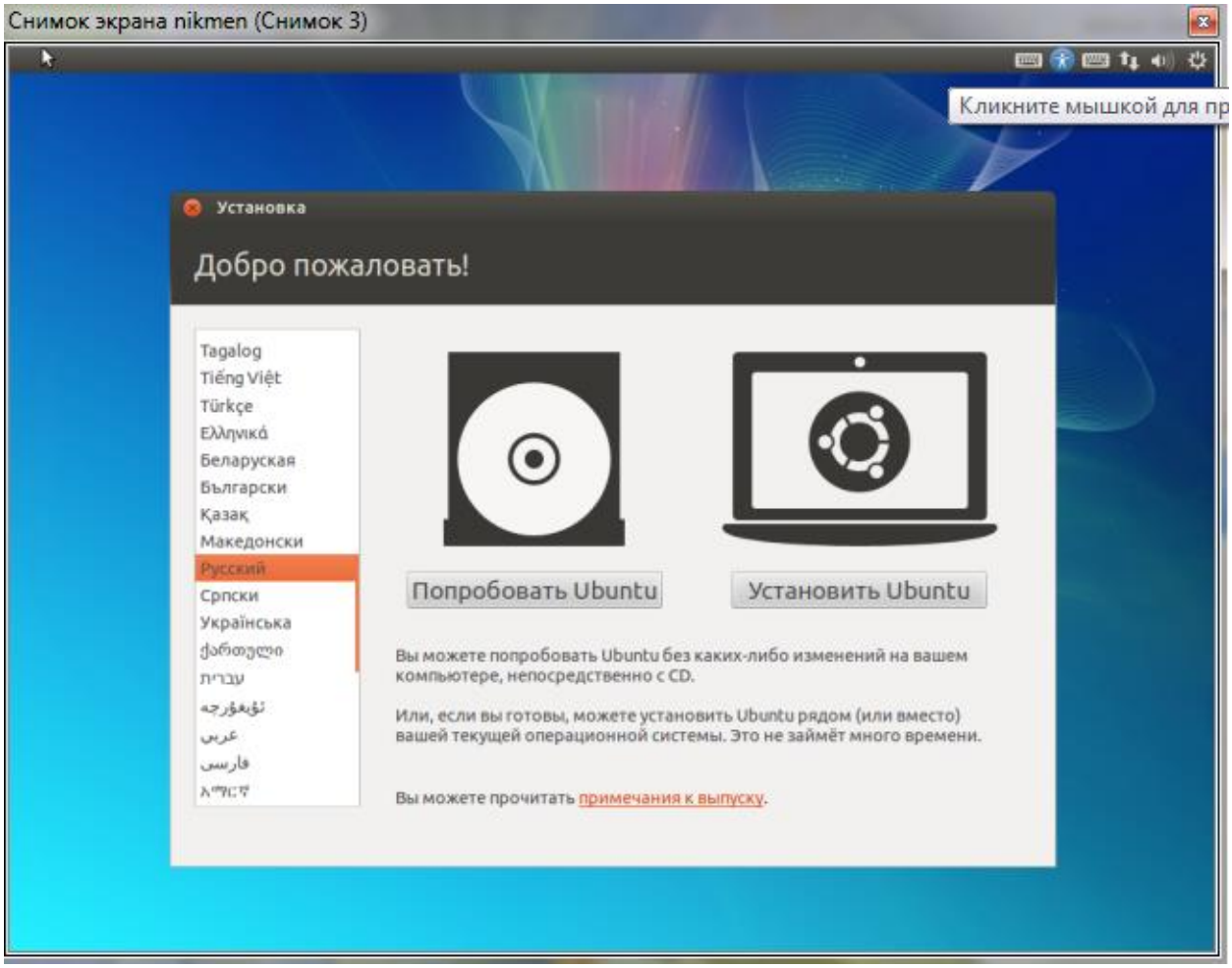
Указать размер жесткого диска (не менее 10Гб)



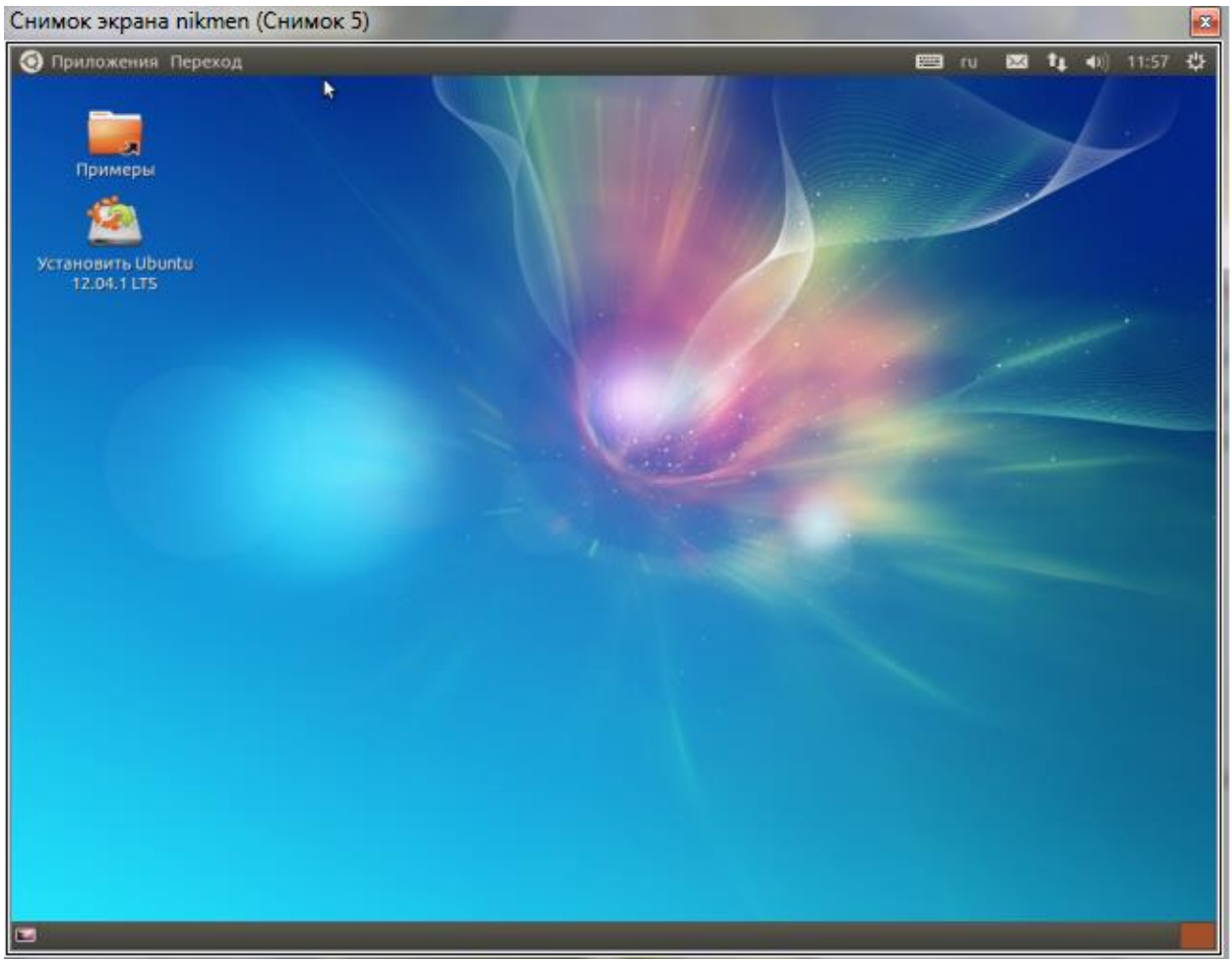
Дождаться установки виртуальной машины. Открыть окно свойств виртуальный машины и перейти в раздел носители. Выбрать IDE контроллер – пусто. Кликнуть по иконки диска в правой панели и выполнить «Выбрать образ загрузочного диска». Указать путь к образу диска Ubuntu и сохранить настройки.



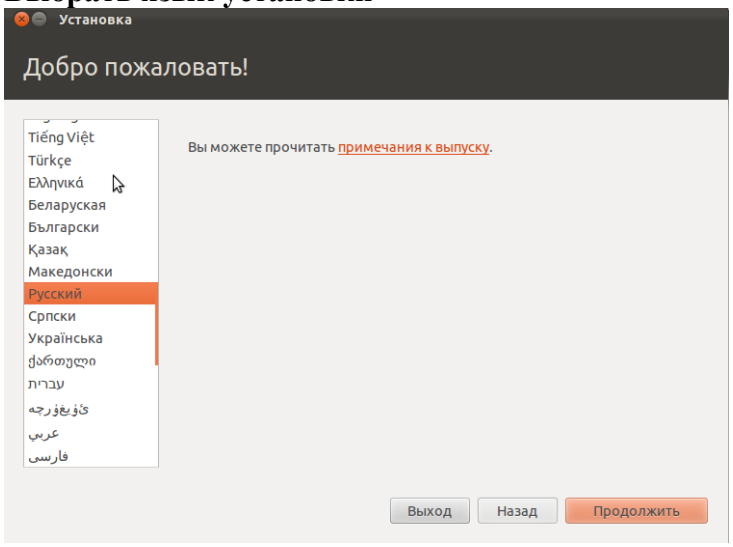
2)Запустить виртуальную машину. Выбрать в списке русский язык.



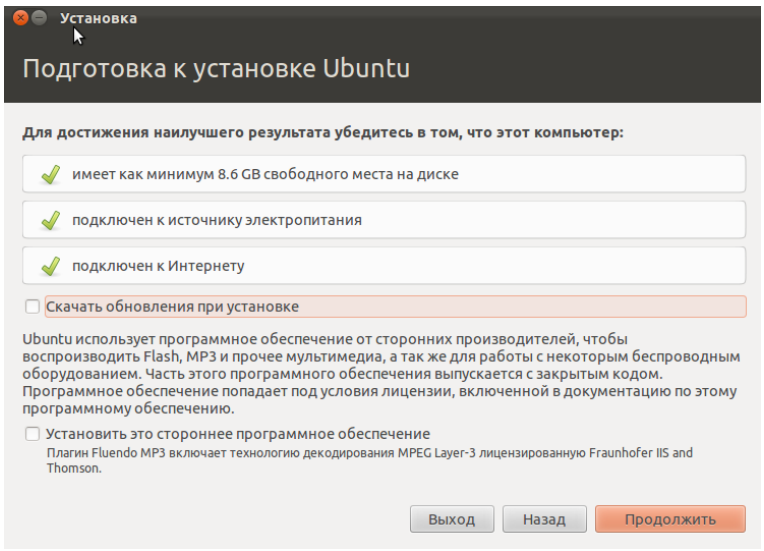
Из предложенных вариантов выбрать «Запустить Ubuntu без установки»
Дождаться загрузки операционной системы. Запустить установка кликнув по ярлыку «Установить Ubuntu»



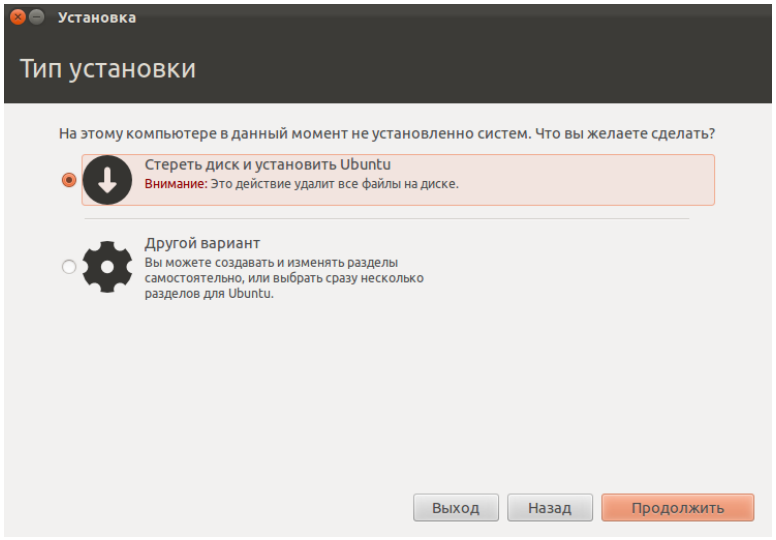
Выбрать язык установки



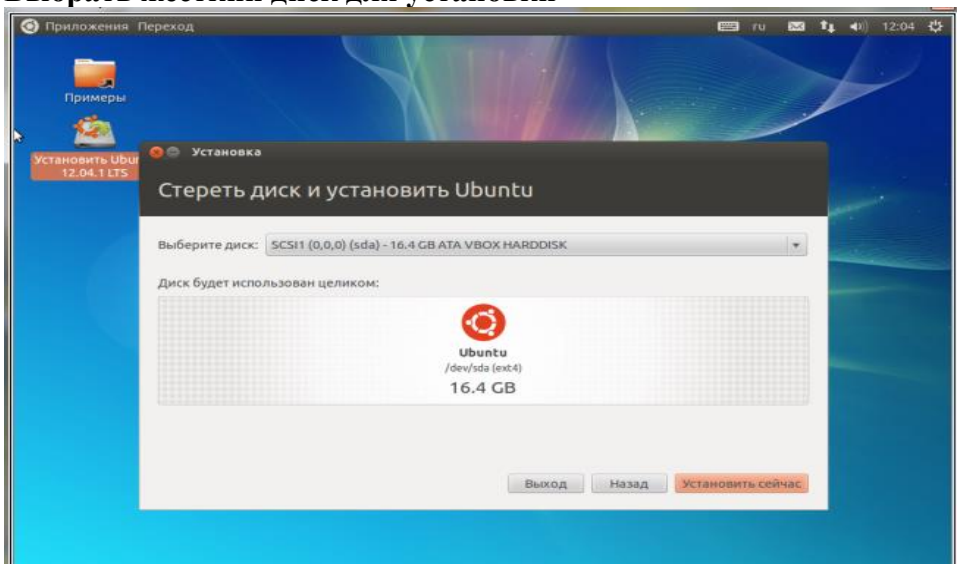
Подтвердить требования к установке



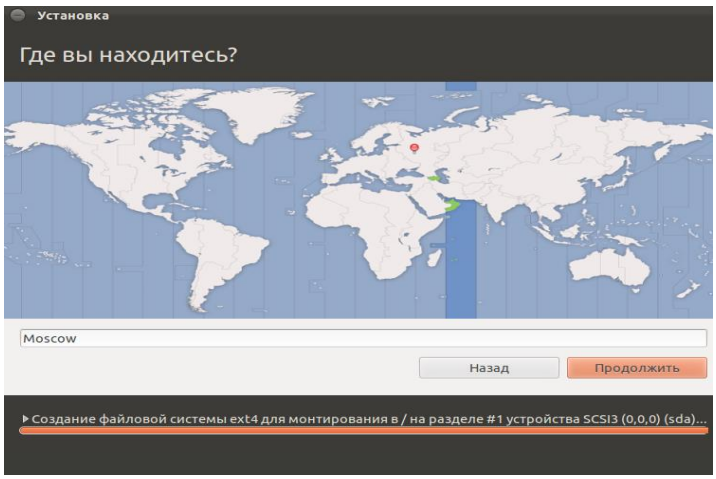
Выбрать тип установки



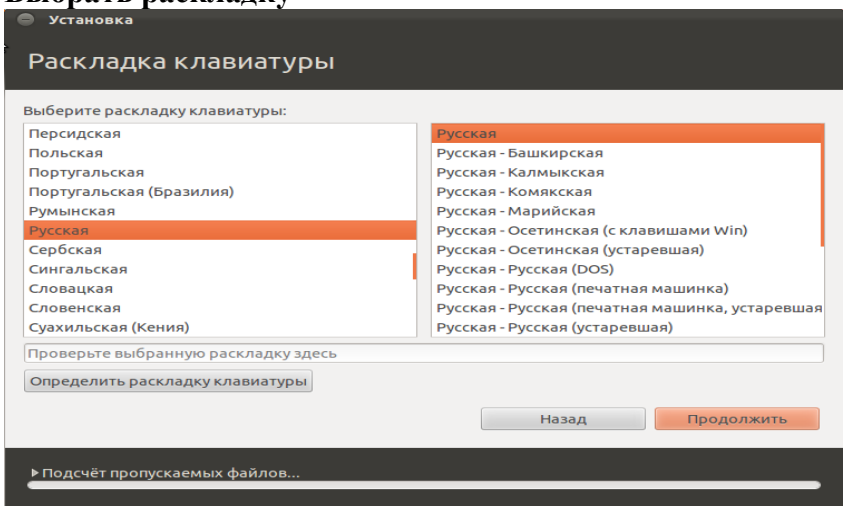
Выбрать жесткий диск для установки



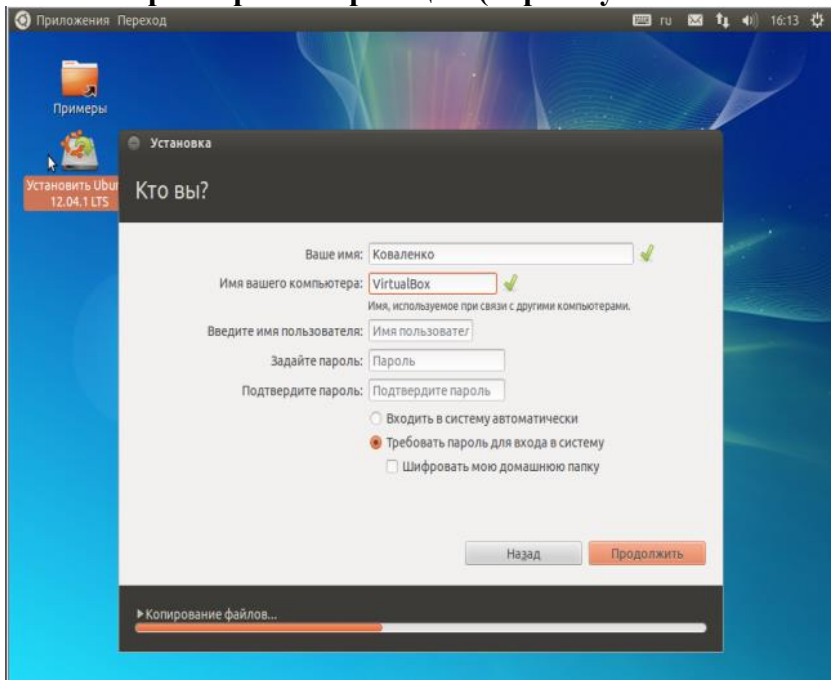
Указать региональные параметры



Выбрать раскладку



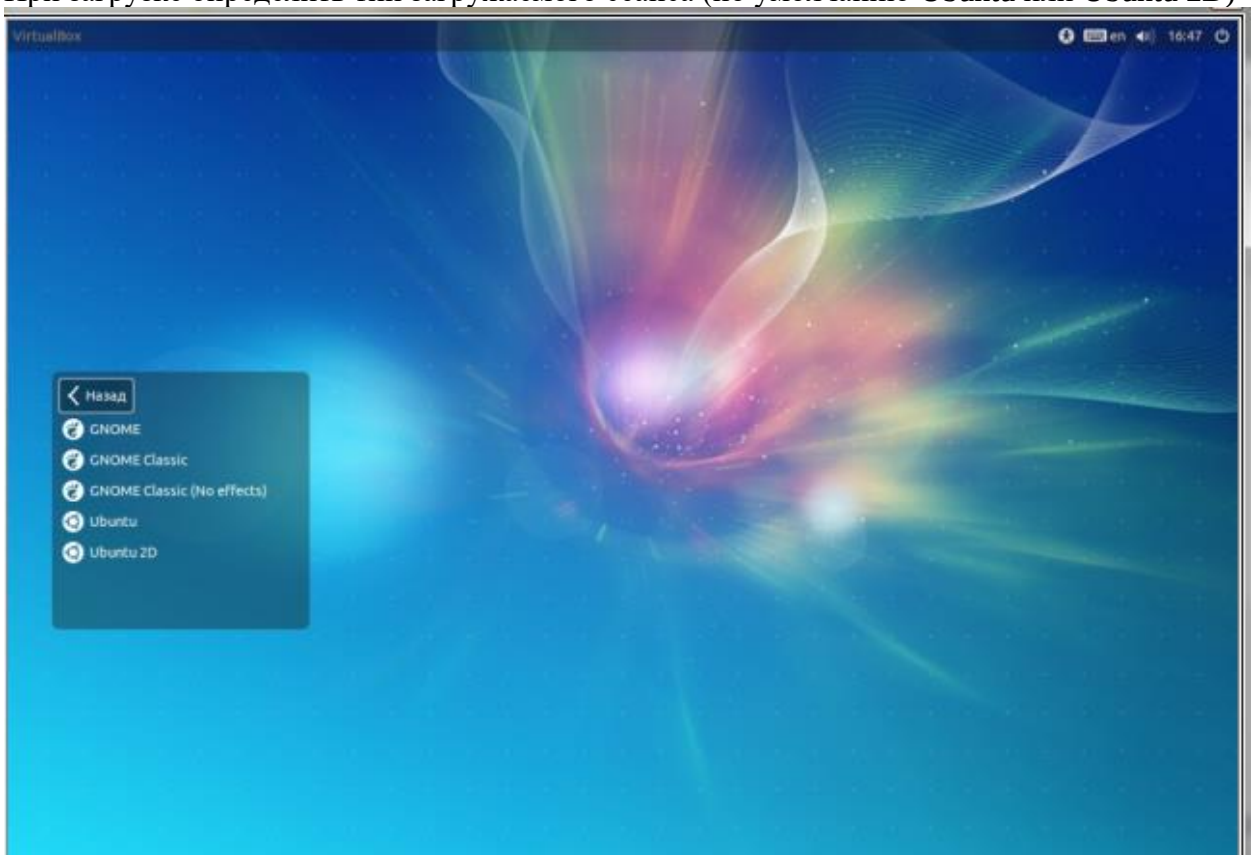
Ввести параметры авторизации (пароль установить 1234)



Дождаться установки операционной системы и перезагрузить виртуальную машины для входа в установленную операционную систему.

3)Установить дополнения гостевой операционной системы:

выбрав в меню VirtualBox «Устройства» - «Установить дополнения гостевой ОС».
Дождаться установки и перезагрузить операционную систему.
При загрузке определить тип загружаемого сеанса (по умолчанию Ubuntu или Ubuntu 2D)



Выводы:
Получен опыт в создании виртуальных машин в **Oracle VM VirtualBox** знакомство с операционной системой **Ubuntu.**

Лабораторная работа 1. Изучение протокола сетевого уровня модели OSI на примере IP

1.1 Цель работы: знать назначение и классификацию программного обеспечения вычислительных сетей, основные возможности сетевых операционных сред, уметь использовать некоторые сетевые прикладные программные пакеты для решения сетевых задач.

1.2 Указания по оформлению отчета:

Отчет должен содержать: титульный лист, цель работы; ответы на контрольные вопросы; выводы.

Указания по сдаче зачета преподавателю

Для сдачи зачета необходимо:

1) предъявить отчет;
2) ответить на контрольные вопросы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Компьютерной сетью называют совокупность узлов (компьютеров, терминалов, периферийных устройств), имеющих возможность информационного взаимодействия друг с другом с помощью специального коммуникационного оборудования и программного обеспечения.

Средства передачи и обработки информации ориентированы в ней на коллективное использование общесетевых ресурсов – информационных, программных, аппаратных.

Компьютерные сети могут работать в различных режимах: обмена данными между абонентами сети, запроса и выдачи информации, сбора информации пакетной обработки данных по запросам пользователей с удаленных терминалов, в диалоговых режимах.

Таким образом, с появлением сетей ЭВМ разрешены две очень важные проблемы:

1) обеспечение в принципе неограниченного доступа к ЭВМ пользователей независимо от территориального расположения,

2) возможность оперативного перемещений больших массивов информации на любые расстояния, позволяющий своевременно получать данные для принятия тех или иных решений.

Использование вычислительных сетей дает предприятию следующие возмож-ности:

1. Разделение дорогостоящих ресурсов;
2. Улучшение доступа к информации;
3. Быстрое и качественное принятие решений;

4. Совершенствование коммуникаций;

5. Свобода в территориальном размещении компьютеров.

Программное обеспечение сетей ЭВМ в расширенном варианте составляют:

1) сетевые операционные системы;

2) сетевые драйвера, протоколы, службы и другое дополнительное программное обеспечение сетевых интерфейсов;

3) прикладное сетевое программное обеспечение.

Под сетевыми операционными системами понимают такие операционные системы, которые обеспечивают пользователям распределенный доступ к сетям ЭВМ.

Во вторую группу входит большой круг всевозможного программного обеспечения в основном изготовителя данного интерфейса (сетевой платы, модема и т.п.) для обеспечения правильной работы сетевого устройства.

При этом под драйвером понимается программа, непосредственно взаимодействующая с интерфейсом - сетевым адаптером и операционной системой (ОС). Драйвер сетевого адаптера взаимодействует с ОС через систему протоколов и служб, которые могут находится как в самих ОС, так и поставляться вместе с устройством.

При этом под сетевым протоколом понимается набор правил поведения сетевых узлов при передаче-приеме информации.

Под сетевыми службами понимается набор программного обеспечения сетевого обеспечения узкоспециального назначения, например:

- клиенты сетей - позволяют подключаться, обозревать и пользоваться сетевыми ресурсами соответствующих сетей,
- службы контроля трафика сетей,
- службы использования доступа к разделяемым ресурсам,
- доменные службы и др.

Круг прикладного сетевого программного обеспечения составляют всевозможные сетевые приложения.

Каждый компьютер работает под управлением собственной операционной систе-мы, а какая-либо «общая» операционная система, распределяющая работу между компьютерами сети, отсутствует. Взаимодействие между компьютерами сети про-исходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства — принтеры, модемы, факс-аппараты и т.д. Разделение локальных ресурсов каждого компьютера между всеми пользователями сети — основная цель создания вычислительной сети.

Каким же образом сказывается на пользователе тот факт, что его компьютер подключен к сети? Прежде всего, он может пользоваться не только файлами, дисками, принтерами и другими ресурсами своего компьютера, но и аналогичными ресурсами других компьютеров, подключенных к той же сети. Правда, для этого недостаточно снабдить компьютеры сетевыми адаптерами и соединить их кабельной системой. Необходимы еще некоторые добавления к операционным системам этих компьютеров. На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров. Обычно такие модули называются программными серверами

(server), так как их главная задача — обслуживать (serve) запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными клиентами (client). Собственно же сетевые адаптеры и каналы связи решают в сети достаточно простую задачу — они передают сообщения с запросами и ответами от одного компьютера к другому, а основную работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционных систем.

Пара модулей «клиент – сервер» обеспечивает совместный доступ пользователей к определенному типу ресурсов, например к файлам. В этом случае говорит, что пользователь имеет дело с файловой службой (service). Обычно сетевая операционная система поддерживает несколько видов сетевых служб для своих пользователей — файловую службу, службу печати, службу электронной почты, службу удаленного доступа и т. п.

Термины «клиент» и «сервер» используются не только для обозначения про-граммных модулей, но и компьютеров, подключенных к сети. Если компьютер предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет — клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы всегда представляют собой распределенные программы, состоящие из нескольких взаимодействующих частей, причем каждая часть, как правило, выполняется на отдельном компьютере сети.

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы - приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая - работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются сетевыми приложениями.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Охарактеризовать сетевые операционные системы по следующей схеме:

1) платность,
2) доступ к исходному коду,
3) многоплатформенность,

4) мультизадачность,
5) количество пользователей,
6) функции управления сетью,
7) интерфейс работы,
8) потребляемые ресурсы.

2. Ответьте на контрольные вопросы

Контрольные вопросы

1. Что понимают под программным обеспечением сетей ЭВМ?
2. Что дает предприятию использование компьютерных сетей?
3. Классификация сетевого программного обеспечения.
4. Что называют операционной системой?
5. Что входит в группу прикладного программного обеспечения?
6. По каким критериям можно охарактеризовать сетевую операционную систему?
7. Что называют сетевым драйвером?
8. Что называют сетевым протоколом?
9. Перечислить сетевые операционные системы.
10. Что такое сетевые службы?
11. Что называют стандартным программным обеспечением ЭВМ?
12. Что такое технология «клиент-сервер»?

Практическое занятие 2 Изучение протокола транспортного уровня модели OSI на примере TCP

Диагностические утилиты TCP/IP

В состав TCP/IP входят диагностические утилиты, предназначенные для проверки конфигурации стека и тестирования сетевого соединения.

Утилита	Применение
hostname	Выводит имя локального хоста. Используется без параметров.
ipconfig	Выводит значения для текущей конфигурации стека TCP/IP: IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System)
ping	Осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом.
tracert	Осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.
arp	Выводит для просмотра и изменения таблицу трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol - определяет локальный адрес по IP-адресу)
route	Модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP.
netstat	Выводит статистику и текущую информацию по соединению TCP/IP.
nslookup	Осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS.

1. Проверка правильности конфигурации TCP/IP с помощью ipconfig.

При устранении неисправностей и проблем в сети TCP/IP следует сначала проверить правильность конфигурации TCP/IP. Для этого используется утилита ipconfig.

Эта команда полезна на компьютерах, работающих с DHCP (Dynamic Host Configuration Protocol), так как дает пользователям возможность определить, какая конфигурация сети TCP/IP и какие величины были установлены с помощью DHCP.

Синтаксис:

ipconfig [/all | /renew[adapter] | /release]

Параметры:

all выдает весь список параметров. Без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;

renew[adapter] обновляет параметры конфигурации DHCP для указанного сетевого адаптера;

release[adapter] освобождает выделенный DHCP IP-адрес;
adapter – имя сетевого адаптера;

displaydns выводит информацию о содержимом локального кэша клиента DNS, используемого для разрешения доменных имен.

Таким образом, утилита ipconfig позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляется IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;
- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0 .

2. Тестирование связи с использованием утилиты ping.

Утилита ping (Packet Internet Grouper) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование конкретного хоста. Использование ping лучший способ проверки того, что между локальным компьютером и сетевым хостом существует маршрут. Хостом называется любое сетевое устройство (компьютер, маршрутизатор), обменивающееся информацией с другими сетевыми устройствами по TCP/IP.

Команда ping проверяет соединение с удаленным хостом путем отправки к этому хосту эхо-пакетов ICMP и прослушивания эхо-ответов. Ping ожидает каждый посланный пакет и печатает количество переданных и принятых пакетов. Каждый принятый пакет проверяется в соответствии с переданным сообщением. Если связь между хостами плохая, из сообщений ping станет ясно, сколько пакетов потеряно.

По умолчанию передается 4 эхо-пакета длиной 32 байта (возможны и другие варианты значения по умолчанию) - периодическая последовательность символов алфавита в верхнем

регистре. Ping позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д.. При получении ответа в поле time указывается, за какое время (в миллисекундах) отправленный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 секунде, то все значения данного поля будут меньше 1000 миллисекунд. Если вы получаете сообщение «Request time out» (Превышен интервал ожидания), то, возможно, если увеличить время ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с помощью ключа –w.

Ping можно использовать для тестирования как имени хоста (DNS или NetBIOS), так и его IP-адреса. Если ping с IP-адресом выполнилась успешно, а с именем – неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Утилита ping используется следующими способами:

1) Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи (loopback address):

ping 127.0.0.1

Если тест успешно пройден, то вы получите следующий ответ:

Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128

2) Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера:

ping IP-адрес_локального_хоста

3) Чтобы проверить, что шлюз по умолчанию функционирует и что можно установить соединение с любым локальным хостом в локальной сети, задается IP-адрес шлюза по умолчанию:

ping IP-адрес_шлюза

4) Для проверки возможности установления соединения через маршрутизатор в команде ping задается IP-адрес удаленного хоста:

ping IP-адрес_удаленного хоста

Синтаксис:

ping [-t] [-a] [-n count] [-l length] [-f] [-i ttl] [-v tos] [-r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] destination-list

Параметры:

-t выполняет команду ping до прерывания. Control-Break - посмотреть статистику и продолжить. Control-C - прервать выполнение команды;

-a позволяет определить доменное имя удаленного компьютера по его IP-адресу;

-n count посылает количество пакетов ECHO, указанное параметром count;

-l length посылает пакеты длиной length байт (максимальная длина 8192 байта);

-f посылает пакет с установленным флагом «не фрагментировать». Этот пакет не будет фрагментироваться на маршрутизаторах по пути своего следования;

-i ttl устанавливает время жизни пакета в величину ttl (каждый маршрутизатор уменьшает ttl на единицу);

-v tos устанавливает тип поля «сервис» в величину tos;

-r count записывает путь выходящего пакета и возвращающегося пакета в поле записи пути. Count - от 1 до 9 хостов;

-s count позволяет ограничить количество переходов из одной подсети в другую (хопов). Count задает максимально возможное количество хопов;

-j host-list направляет пакеты с помощью списка хостов, определенного параметром host-list. Последовательные хосты могут быть отделены промежуточными маршрутизаторами (гибкая статическая маршрутизация). Максимальное количество хостов в списке, позволенное IP, равно 9;

-k host-list направляет пакеты через список хостов, определенный в host-list. Последовательные хосты не могут быть разделены промежуточными маршрутизаторами (жесткая статическая маршрутизация). Максимальное количество хостов – 9;

-w timeout указывает время ожидания (timeout) ответа от удаленного хоста в миллисекундах (по умолчанию – 1сек);

destination-list указывает удаленный хост, к которому надо направить пакеты ping.

Пример использования утилиты ping:

C:\WINDOWS>ping –n 10 www.netscape.com

Обмен пакетами с www.netscape.com [205.188.247.65] по 32 байт:

Ответ от 205.188.247.65: число байт=32 время=194мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=240мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=173мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=250мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=187мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=239мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=263мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=230мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=185мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=406мс TTL=48
Статистика Ping для 205.188.247.65:
Пакетов: послано = 10, получено = 10, потеряно = 0 (0% потерь)
Приблизительное время передачи и приема:
Наименьшее = 173мс, наибольшее = 406мс, среднее =236мс

В случае невозможности проверить доступность хоста утилита выводит информацию об ошибке. Ниже приведен пример ответа утилиты ping при попытке послать запрос на несуществующий хост.

Обмен пакетами с 172.16.6.21 по 32 байт:

Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.

Статистика Ping для 172.16.6.21:
Пакетов: отправлено = 4, получено = 0, потеряно = 4 (100% потерь),

Приблизительное время передачи и приема:
наименьшее = 0мс, наибольшее = 0мс, среднее = 0мс

Утилита сообщает не об отсутствии хоста, а о том, что за отведенное время не был получен ответ на посланный запрос. Причиной этого не обязательно является отсутствие хоста в сети. Проблема может крыться в сбоях связи, перегрузке или неправильной настройке маршрутизаторов и т. п. Ошибка «сеть недоступна» (network unreachable) прямо указывает на проблемы маршрутизации.

3. Изучение маршрута между сетевыми соединениями с помощью утилиты tracert.

Tracert - это утилита трассировки маршрута. Она использует поле TTL (time-to-live, время жизни) пакета IP и сообщения об ошибках ICMP для определения маршрута от одного хоста до другого.

Утилита tracert может быть более содержательной и удобной, чем ping, особенно в тех случаях, когда удаленный хост недостижим. С помощью нее можно определить район проблем со связью (у Internet-провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько далеко будет отслежен маршрут. Если возникли проблемы, то утилита выводит на экран звездочки (*), либо сообщения типа «Destination net unreachable», «Destination host unreachable», «Request time out», «Time Exeeded».

Утилита tracert работает следующим образом: посылается по 3 пробных эхо-пакета на каждый хост, через который проходит маршрут до удаленного хоста. На экран при этом выводится время ожидания ответа на каждый пакет (Его можно изменить с помощью параметра -w). Пакеты посылаются с различными величинами времени жизни. Каждый маршрутизатор, встречающийся по пути, перед перенаправлением пакета уменьшает величину TTL на единицу. Таким образом, время жизни является счетчиком точек промежуточной доставки (хопов). Когда время жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет в компьютер-источник сообщение ICMP “Time Exeeded” (Время истекло). Маршрут определяется путем посылки первого эхо-пакета с TTL=1. Затем TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока пакет не достигнет удаленного хоста, либо будет достигнута максимально возможная величина TTL (по умолчанию 30, задается с помощью параметра -h).

Маршрут определяется путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто уничтожают пакеты с истекшим TTL и не будут видны утилите tracert.

Синтаксис:

tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] имя_целевого_хоста

Параметры:

-d указывает, что не нужно распознавать адреса для имен хостов;

-h maximum_hops указывает максимальное число хопов для того, чтобы искать цель;

-j host-list указывает нежесткую статическую маршрутизацию в соответствии с host-list;

-w timeout указывает, что нужно ожидать ответ на каждый эхо-пакет заданное число мсек.

4. Утилита arp.

Основная задача протокола ARP – трансляция IP-адресов в соответствующие локальные адреса. Для этого ARP-протокол использует информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса. В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин.) запись не была востребована, то она удаляется из кэша.

Синтаксис:

arp [-s inet_addr eth_addr] | [-d inet_addr] | [-a]

Параметры:

-s занесение в кэш статических записей;

-d удаление из кэша записи для определенного IP-адреса;

-a просмотр содержимого кэша для всех сетевых адаптеров локального компьютера;

inet_addr - IP-адрес;

eth_addr - MAC-адрес.

5. Утилита route.

Утилита **route** предназначена для работы с локальной таблицей маршрутизации. Она имеет следующий **синтаксис:**

route [-f] [-p] [команда [узел] [MASK маска] [шлюз] [METRIC метрика] [IF интерфейс]]

Параметры:

-f Очистка таблицы маршрутизации.

-p При указании совместно с командой ADD создает постоянную запись, которая сохраняется после перезагрузки компьютера. По умолчанию записи таблицы маршрутов не сохраняются при перезагрузке.

команда одна из четырех команд:

PRINT - вывод информации о маршруте;

ADD - добавление маршрута;

DELETE - удаление маршрута;

CHANGE - изменение маршрута.

узел	адресуемый узел
маска	маска подсети; по умолчанию используется маска 255.255.255.255
шлюз	адрес шлюза
метрика	метрика маршрута;
интерфейс	идентификатор интерфейса, который будет использован для пересылки пакета

Для команд PRINT и DELETE возможно использование символов подстановки при указании адресуемого узла или шлюза. Параметр шлюза для этих команд может быть опущен. При добавлении и изменении маршрутов утилита route осуществляет проверку введенной информации на соответствие условию (УЗЕЛ & МАСКА) == УЗЕЛ. Если это условие не выполняется, то утилита выдает сообщение об ошибке и не добавляет или не изменяет маршрут.

Утилита осуществляет поиск имен сетей в файле networks. Поиск имен шлюзов осуществляется в файле hosts. Оба файла расположены в папке %systemroot%\system32\drivers\etc. Наличие и заполнение этих файлов не обязательно для нормального функционирования утилиты route и работы маршрутизации.

Хотя в большинстве случаев на рабочей станции это не требуется, можно вручную редактировать таблицы маршрутизации.

Пример использования утилиты route:

Добавление статического маршрута:

route add 172.16.6.0 MASK 255.255.255.0 172.16.11.1 METRIC 1 IF 0x1000003

6. Утилита netstat.

Утилита netstat позволяет получить статическую информацию по некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит сведения о текущих сетевых соединениях. Особенно она полезна на брандмауэрах, с ее помощью можно обнаружить нарушения безопасности периметра сети.

Синтаксис:

netstat [-a] [-e] [-n] [-s] [-p protocol] [-r]

Параметры:

-a выводит перечень всех сетевых соединений и прослушивающихся портов локального компьютера;

-e выводит статистику для Ethernet-интерфейсов (например, количество полученных и отправленных байт);

-n выводит информацию по всем текущим соединениям (например, TCP) для всех сетевых интерфейсов локального компьютера. Для каждого соединения выводится информация об IP-адресах локального и удаленного интерфейсов вместе с номерами используемых портов;

-s выводит статистическую информацию для протоколов UDP, TCP, ICMP, IP. Ключ «/more» позволяет просмотреть информацию постранично;

-r выводит содержимое таблицы маршрутизации.

7. Утилита nslookup.

Утилита **nslookup** предназначена для диагностики службы DNS, в простейшем случае - для выполнения запросов к DNS-серверам на разрешение имен в IP-адреса. В общем случае утилита позволяет просмотреть любые записи DNS-сервера:

A – каноническое имя узла, устанавливает соответствие доменного имени ip-адресу.

SOA – начало полномочий, начальная запись, единственная для зоны;

MX – почтовые серверы (хосты, принимающие почту для заданного домена);

NS – серверы имен (содержит авторитетные DNS-серверы для зоны);

PTR – указатель (служит для обратного преобразования ip-адреса в символьное имя хоста)

и т. д.

Утилита nslookup достаточно сложна и содержит свой собственный командный интерпретатор.

В простейшем случае (без входа в командный режим) утилита **nslookup** имеет следующий

Синтаксис:

nslookup хост [сервер]

Параметры:

Хост DNS-имя хоста, которое должно быть преобразовано в IP-адрес.

Сервер Адрес DNS-сервера, который будет использоваться для разрешения имени. Если этот параметр опущен, то будут последовательно использованы адреса DNS-серверов из параметров настройки протокола TCP/IP.

Примеры использования утилиты nslookup:

1. Получение списка серверов имен для домена yandex.ru без входа в командный режим (с использованием ключей).

```
C:\> nslookup -type=ns yandex.ru
Server: dns01.catv.ext.ru
Address: 217.10.44.35
Non-authoritative answer:
yandex.ru    nameserver = ns4.yandex.ru
yandex.ru    nameserver = ns5.yandex.ru
yandex.ru    nameserver = ns2.yandex.ru
yandex.ru    nameserver = ns1.yandex.ru

ns2.yandex.ru  internet address = 213.180.199.34
ns5.yandex.ru  internet address = 213.180.204.1
```

2. Получение записи SOA домена yandex.ru с авторитетного сервера с использование командного интерпретатора nslookup.

```
C:\>nslookup
Default Server: dns04.catv.ext.ru
Address: 217.10.39.4
> set type=SOA
> server ns2.yandex.ru
Default Server: ns2.yandex.ru
Address: 213.180.199.34
> yandex.ru
Server: ns1.yandex.ru
Address: 213.180.193.1
>yandex.ru
   primary name server = ns1.yandex.ru
   responsible mail addr = sysadmin.yandex-team.r
   serial = 2009022707
```

refresh = 1800 (30 mins)

retry = 900 (15 mins)

expire = 2592000 (30 days)

default TTL = 900 (15 mins)

yandex.ru nameserver = ns5.yandex.ru

yandex.ru nameserver = ns1.yandex.ru

yandex.ru nameserver = ns4.yandex.ru

yandex.ru nameserver = ns2.yandex.ru

ns1.yandex.ru internet address = 213.180.193.1

ns2.yandex.ru internet address = 213.180.199.34

ns4.yandex.ru internet address = 77.88.19.60

ns5.yandex.ru internet address = 213.180.204.1

> exit

3. Получение адреса почтового сервера для домена yandex.ru.

C:\ >nslookup

Default Server: dns01.catv.ext.ru

Address: 217.10.44.35

> set q=mx

> yandex.ru

Server: dns01.catv.ext.ru

Address: 217.10.44.35

Non-authoritative answer:

yandex.ru MX preference = 10, mail exchanger = mx2.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx3.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx1.yandex.ru

yandex.ru nameserver = ns2.yandex.ru

yandex.ru nameserver = ns1.yandex.ru

yandex.ru nameserver = ns4.yandex.ru

yandex.ru nameserver = ns5.yandex.ru

mx1.yandex.ru internet address = 77.88.21.89

mx2.yandex.ru internet address = 93.158.134.89

mx3.yandex.ru internet address = 213.180.204.89

ns2.yandex.ru internet address = 213.180.199.34

ns4.yandex.ru internet address = 77.88.19.60

ns5.yandex.ru internet address = 213.180.204.1

>

Указав ключ type=any, можно получить все записи о узле или домене. Ключи querytype, t, q эквивалентны type.

Задания на лабораторную работу

1. Изучите методические указания к лабораторной работе.
2. Выполните упражнения.
3. Оформите отчет по лабораторной работе, описав выполнение упражнений и дав краткие ответы на контрольные вопросы.

Упражнение 1. Получение справочной информации по командам.

Выведите на экран справочную информацию по всем рассмотренным утилитам (см. таблицу п.1). Для этого в командной строке введите имя утилиты без параметров и дополните **/?**.

Сохраните справочную информацию в отдельном файле.

Изучите ключи, используемые при запуске утилит.

Упражнение 2. Получение имени хоста.

Выведите на экран имя локального хоста с помощью команды hostname. Сохраните результат в отдельном файле.

Упражнение 3. Изучение утилиты ipconfig.

Проверьте конфигурацию TCP/IP с помощью утилиты ipconfig. Заполните таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	

Адрес WINS-сервера	

Упражнение 4. Тестирование связи с помощью утилиты ping.

1. Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверьте функционирование основного шлюза, послав 5 эхо-пакетов длиной 64 байта.
3. Проверьте возможность установления соединения с удаленным хостом.
4. С помощью команды ping проверьте адреса (взять из списка локальных ресурсов на сайте aspu.ru) и для каждого из них отметьте время отклика. Попробуйте изменить параметры команды ping таким образом, чтобы увеличилось время отклика. Определите IP-адреса узлов.

Упражнение 5. Определение пути IP-пакета.

С помощью команды tracert проверьте для перечисленных ниже адресов, через какие промежуточные узлы идет сигнал. Изучите ключи команды.

a) aspu.ru

b) mathmod.aspu.ru

c) yarus.aspu.ru

Упражнение 6: Просмотр ARP-кэша.

С помощью утилиты arp просмотрите ARP-таблицу локального компьютера.

Внести в кэш локального компьютера любую статическую запись.

Упражнение 7: Просмотр локальной таблицы маршрутизации.

С помощью утилиты route просмотреть локальную таблицу маршрутизации.

Упражнение 8. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

С помощью утилиты netstat выведите перечень сетевых соединений и статистическую информацию для протоколов UDP, TCP, ICMP, IP.

Упражнение 9. Получение DNS-информации с помощью nslookup.

1) Узнайте ip-адреса узлов, сайтов ростовских предприятий.

2) Узнайте авторитетные (компетентные) сервера для этих узлов.

3) Получите запись SOA с одного из этих серверов для домена выбранного вами домена.

Контрольные вопросы

1. Раскрыть термины: хост, шлюз, хоп, время жизни пакета, маршрут, маска сети, авторитетный/неавторитетный (компетентный) DNS-сервер, порт TCP, петля обратной связи, время отклика.
2. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
3. Каким образом команда ping проверяет соединение с удаленным хостом?
4. Каково назначение протокола ARP?
5. Как утилита ping разрешает имена узлов в ip-адреса (и наоборот)?
6. Какие могут быть причины неудачного завершения ping и tracert? (превышен интервал ожидания для запроса, сеть недоступна, превышен срок жизни при передаче пакета).
7. Всегда ли можно узнать символьное имя узла по его ip-адресу?
8. Какой тип записи запрашивает у DNS-сервера простейшая форма nslookup?

Лабораторная работа 2 Изучение протокола прикладного уровня модели OSI на примере HTTP

MAC-адресация

MAC-адрес (от англ. Media Access Control — управление доступом к среде) — это уникальный идентификатор, сопоставляемый с различными типами оборудования для компьютерных сетей. Большинство сетевых протоколов канального уровня используют одно из трёх пространств MAC-адресов, управляемых IEEE: MAC-48, EUI-48 и EUI-64. Адреса в каждом из пространств теоретически должны быть глобально уникальными. Не все протоколы используют MAC-адреса, и не все протоколы, использующие MAC-адреса, нуждаются в подобной уникальности этих адресов.

В широковещательных сетях (таких, как сети на основе Ethernet) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу. Таким образом, MAC-адреса формируют основу сетей на канальном уровне, которую используют протоколы более высокого (сетевого) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, ARP и RARP в сетях TCP/IP).

Адреса типа MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDDI, WiMAX и др. Они состоят из 48 бит, таким образом, адресное пространство MAC-48 насчитывает 248 (или 281 474 976 710 656) адресов. Согласно подсчётам IEEE, этого запаса адресов хватит по меньшей мере до 2100 года.

EUI-48 от MAC-48 отличается лишь семантически: в то время как MAC-48 используется для сетевого оборудования, EUI-48 применяется для других типов аппаратного и программного обеспечения.

Идентификаторы EUI-64 состоят из 64 бит и используются в FireWire, а также в IPv6 в качестве младших 64 бит сетевого адреса узла.

Структура MAC-адреса

Стандарты IEEE определяют 48-разрядный (6 октетов) MAC-адрес, который разделен на четыре части.

Первые 3 октета (в порядке их передачи по сети; старшие 3 октета, если рассматривать их в традиционной бит-реверсной шестнадцатиричной записи MAC-адресов) содержат 24-битный уникальный идентификатор организации (OUI)[1], или (Код MFG - Manufacturing, производителя), который производитель получает в IEEE. При этом используются только младшие 22 разряда (бита), 2 старшие имеют специальное назначение:

первый бит указывает, для одиночного (0) или группового (1) адресата предназначен кадр

следующий бит указывает, является ли MAC-адрес глобально (0) или локально (1) администрируемым.

Следующие три октета выбираются изготовителем для каждого экземпляра устройства. За исключением сетей системной сетевой архитектуры SNA.

Таким образом, глобально администрируемый MAC-адрес устройства глобально уникален и обычно «зашит» в аппаратуру.

Администратор сети имеет возможность, вместо использования «зашитого», назначить устройству MAC-адрес по своему усмотрению. Такой локально администрируемый MAC-адрес выбирается произвольно и может не содержать информации об OUI. Признаком локально администрируемого адреса является соответствующий бит первого октета адреса

Среди людей, плохо разбирающихся в сетях, существует распространенное мнение, что MAC-адрес железно вшит в сетевую карту и сменить его нельзя или можно только с помощью программаторов. На самом деле, это не так. MAC-адрес легко меняется программным путем, так как значение, указанное через драйвер, имеет более высокий приоритет, чем зашитый в плату. Поскольку многие DHCP-серверы, которые раздают динамические IP, обычно делают привязку по MAC-адресу (то есть при неизменном MAC они будут выдавать одинаковый IP), то смена MAC-адреса через драйвер поможет сменить и локальный IP адрес. Постоянно изменяя свой MAC-адрес, пользователь становится почти неуязвимым для администратора сети, поскольку для идентификации компьютера можно использовать только IP, MAC-адрес и сетевое имя. Если эти три параметра будут меняться, то становится невозможно определить, разные это клиенты или один и тот же. При этом администратор может жестко привязать выдаваемые IP адреса к списку определенных MAC-адресов пользователей — таким образом, DHCP может отказать устройству с незнакомым MAC в выдаче адреса.

Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена.

В терминологии TCP/IP под локальным адресом понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной интерсети. В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP предполагалось наличие разных типов локальных адресов. Если подсетью интерсети является локальная сеть, то локальный адрес - это MAC-адрес. Однако протокол IP может работать и над протоколами более высокого уровня, например над протоколом IPX или X.25. В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.25. Компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка-точка».

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100. IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера

узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Internet Network Information Center, InterNIC), если сеть должна работать как составная часть Internet. Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Символьные доменные имена. Символьные имена в IP-сетях называются доменными и строятся по иерархическому признаку. Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами.

1.2. Классы IP-адресов

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30 - традиционная десятичная форма представления адреса;

10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая - к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому классу относится тот или иной IP-адрес. На рисунке 1 показана структура IP-адресов различных классов.

Класс A. 0ccccccc yyyyyyyy yyyyyyyy yyyyyyyy

Класс B. 10cccccc cccccccc yyyyyyyy yyyyyyyy

Класс C. 110ccccc cccccccc cccccccc yyyyyyyy

Класс D. 1110aaaa aaaaaaaa aaaaaaaa aaaaaaaa

Класс E. 11110zzz zzzzzzzz zzzzzzzz zzzzzzzz

Рисунок 1 – Классы IP-адресов (c - бит, входящий в номер сети; y - бит, входящий в номер узла; a - бит, входящий в адрес группы multicast; z - бит, входящий в зарезервированный адрес)

Если адрес начинается с 0, то сеть относят к классу A и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) Количество узлов в сетях класса A может достигать 2^{24}, то есть 16 777 216 узлов.

Если первые два бита адреса равны 10, то сеть относится к классу B. В сетях класса B под номер сети и под номер узла отводится по 16 бит. Таким образом, сеть класса B является сетью средних размеров с максимальным числом узлов 2^{16}, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть класса C. В этом случае под номер сети отводится 24 бита, а под номер узла - 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 2^8, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

1.3. Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP.

- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет.

- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным. сообщением (limited broadcast).

- Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется широковещательным сообщением (broadcast).

При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса C под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название loopback.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интерсети - они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер которой указан в адресе назначения.

Уже упоминавшаяся форма группового IP-адреса - multicast - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы multicast не обязательно должны принадлежать одной сети. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей. Если такие средства найдут широкое применение, то Internet сможет создать серьезную конкуренцию радио и телевидению.

1.4. Использование масок в IP-адресации

Важным элементом разбиения адресного пространства Internet являются подсети. Подсеть – это подмножество сети, не пересекающееся с другими подсетями. Это означает, что сеть организации может быть разбита на фрагменты, каждый из которых будет составлять подсеть. Реально, каждая подсеть соответствует физической локальной сети (например, сегменту Ethernet). Подсети используются для того, чтобы обойти ограничения физических сетей на число узлов в них и максимальную длину кабеля в сегменте сети. Например, сегмент тонкого Ethernet имеет максимальную длину 185 м и может включать до 32 узлов. Самая маленькая сеть класса С может состоять из 254 узлов. Для того, чтобы достичь этой цифры надо объединить несколько физических сегментов сети. Сделать это можно либо с помощью физических устройств (например, репитеров), либо при помощи машин-шлюзов. В первом случае разбиение на подсети не требуется, так как логически сеть выглядит как одно целое. При использовании шлюза сеть разбивается на подсети.

Разбиение сети на подсети использует ту часть IP-адреса, которая закреплена за номерами хостов. Администратор сети может замаскировать часть IP-адреса и использовать её для назначения номеров подсетей. Фактически, способ разбиения адреса на две части, теперь будет применяться к адресу хоста из IP-адреса сети, в которой организуется разбиение на подсети.

Маска подсети – это четыре байта, которые накладываются на IP-адрес для получения номера подсети. Например, маска 255.255.255.0 позволяет разбить сеть класса В на 254 подсети по 254 узла в каждой. Подсети не только решают, но и создают ряд проблем. Например, происходит потеря адресов, но уже не по причине физических ограничений, а по причине принципа построения адресов подсети. Так, при выделении трех битов на адрес подсети, приводит к образованию не восьми, а только шести подсетей, так как номера 0 и 7 нельзя использовать в силу специального значения IP-адресов, состоящих из нулей или из единиц.

Для стандартных классов сетей маски имеют следующие значения:

- класс А - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс В - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);

- класс С- 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, адрес 185.23.44.206 попадает в диапазон 128-191, то есть адрес относится к классу В. Следовательно, номером сети являются первые два байта, дополненные двумя нулевыми байтами - 185.23.0.0, а номером узла - 0.0.44.206. Если этот адрес ассоциировать с маской 255.255.255.0, то номером подсети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде:

IP-адрес 129.64.134.5 - 10000001.01000000.10000110.00000101

Маска 255.255.128.0 - 11111111.11111111.10000000.00000000

Если использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

10000001. 01000000. 10000000. 00000000 или в десятичной форме записи - номер сети 129.64.128.0, а номер узла 0.0.6.5.

Практическая часть

Определить тип используемых в компьютере расположенного в аудитории MAC и IP адреса, а также маску под сети. Для этого вызываем меню выполнить комбинацией клавиш windows+R, затем в водим в появившимся окошке команду cmd и нажимаем выполнить. В появившейся командной строке вводим команду ipconfig /all.

Из вышедшего окошка выписываем данные в лабораторную работу:

1. Физический адрес – mac адрес
2. Ip – адрес
3. Маску подсети.

Определить из каких частей состоят выписанные мас и ip адреса.

Контрольные вопросы

1. Мас адрес и его структура.
2. ip- адрес и его структура.
3. Для чего применяется маска подсети.
4. Какие есть специальные ip – адреса.

Практическое занятие 3 Сетевые диагностические утилиты операционных систем семейства Linux на примере Ubuntu

ЦЕЛЬ РАБОТЫ:

1. Знать принципы анализа сетевого трафика.
2. Научиться использовать сетевой анализатор (сниффер Wireshark).
3. Научиться анализировать сетевой трафик на примере протоколов ARP, IP и ICMP.

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

- обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);
- подключением сниффера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер;
- через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;
- через атаку на канальном (2-й) или сетевом (3-й) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети.

Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.
- Отлаживать протоколы сетевых приложений.
- Локализовать неисправность или ошибку конфигурации.
- Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи.
- Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие.
- Перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

Постепенно из инструментов, предназначенных только для диагностики, сниферы постепенно превратились в средства для исследований и обучения. Например, они постоянно используются для изучения динамики и взаимодействий в сетях. В частности, они позволяют легко и наглядно изучать тонкости сетевых протоколов. Наблюдая за данными, которые посылает протокол, вы можете глубже понять его функционирование

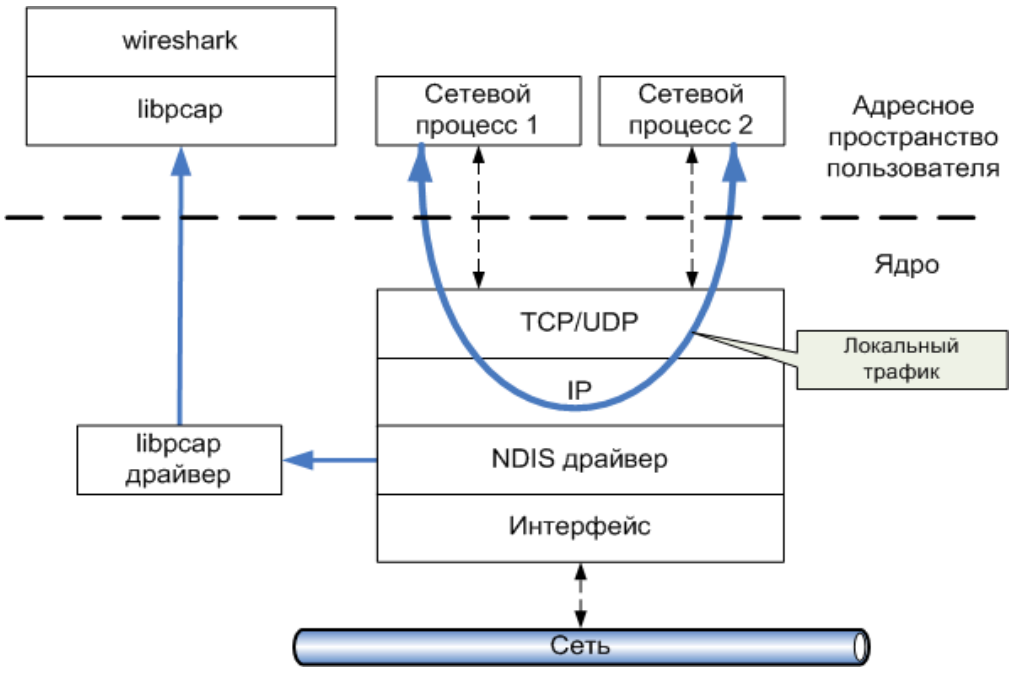


Рис. 1. Принцип «захвата» снифферем сетевого

на практике, а заодно увидеть, когда некоторая конкретная реализация работает не в соответствии со спецификацией.

На сегодняшний момент существует достаточно большое количество хороших реализаций снифферов. Некоторое из них:

- Tcpdump (http://www.tcpdump.org/) – консольный вариант сниффера. Портирован почти подо все наиболее распространенные ОС;
- Wireshark (http://www.wireshark.org/) до недавнего момента был известен под названием Ethreal;
- WinDump http://www.winpcap.org/windump;
- IP Sniffer
- и др.

Сниффер Wireshark

Программа Wireshark является одной из самых удобных реализаций снифферов. Портирована на большое количество платформ. Распространяется абсолютно бесплатно. Имеет смысл использовать данный сниффер для изучения и анализа сетевых протоколов.

Но для начала рассмотрим базовый принцип работы сниффера как раз на примере Wireshark.

Базовый принцип работы снифферов

Давайте рассмотрим с вами рис. 1. На нем изображена схематично структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс сниффера работают в пользовательском режиме.

На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

Основными компонентами сниффера являются: драйвер для захвата пакетов (libpcap драйвер), интерфейсная библиотека (libpcap) и интерфейс пользователя (Wireshark). Библиотека libpcap (реализация под ОС Windows носит название WinPcap - http://www.winpcap.org) – универсальная сетевая библиотека, самостоятельно реализующая большое количество сетевых протоколов и работающая непосредственно с NDIS (Network Driver Interface Specification) драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark.

Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию ВСЕХ данных проходящих через драйвер сетевого интерфейса. Изменения в сами данные не вносятся!

Основной нюанс использования сниффера заключается в том, что он не позволяет производить анализ локального трафика, т.к. он не проходит через драйвер сетевого устройства (см. рис 1.). Т.е., если вы захотите проанализировать сниффером трафик между 2-ми сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то у вас ждет разочарование. Однако, например при использовании виртуальных машин, сниффер будет работать без проблем, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры, поэтому трафик идет через драйвера как и в нормальной ситуации при взаимодействии с другими физическими сетевыми машинами.

Также к недостаткам большинства снифферов стоит отнести и тот факт, что, позволяя анализировать трафик, проходящий через сетевой интерфейс, они не могут указать, какое именно приложение генерирует или получает его. Это объясняется те, что информация об этом хранится на сетевом (например, IP) уровне сетевого стека, а большинство снифферов использует собственную реализацию стека протоколов (например, библиотеку WinPcap), которая (как уже было показано) работает непосредственно с драйверами устройств.

Также, снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

Использование программы Wireshark

Данный сниффер позволяет в режиме реального времени захватывать пакеты из сети, и анализировать их структуру. Также можно анализировать структуру пакетов из файла, содержащего трафик, полученный, например, программой «tcpdump» (unix/linux).

На рис. 2. изображено основное окно программы Wireshark. В стандартном режиме окно сниффера делится на 3 фрейма (панели): список захваченных пакетов, «анализатор» протоколов и исходные данные пакетов. Размер каждого фрейма можно менять по своему усмотрению.

Рассмотрим эти панели подробнее.

Верхняя панель содержит список пакетов, захваченных из сети. Список можно отсортировать по любому полю (в прямом или обратном порядке) – для этого нажать на заголовок соответствующего поля.

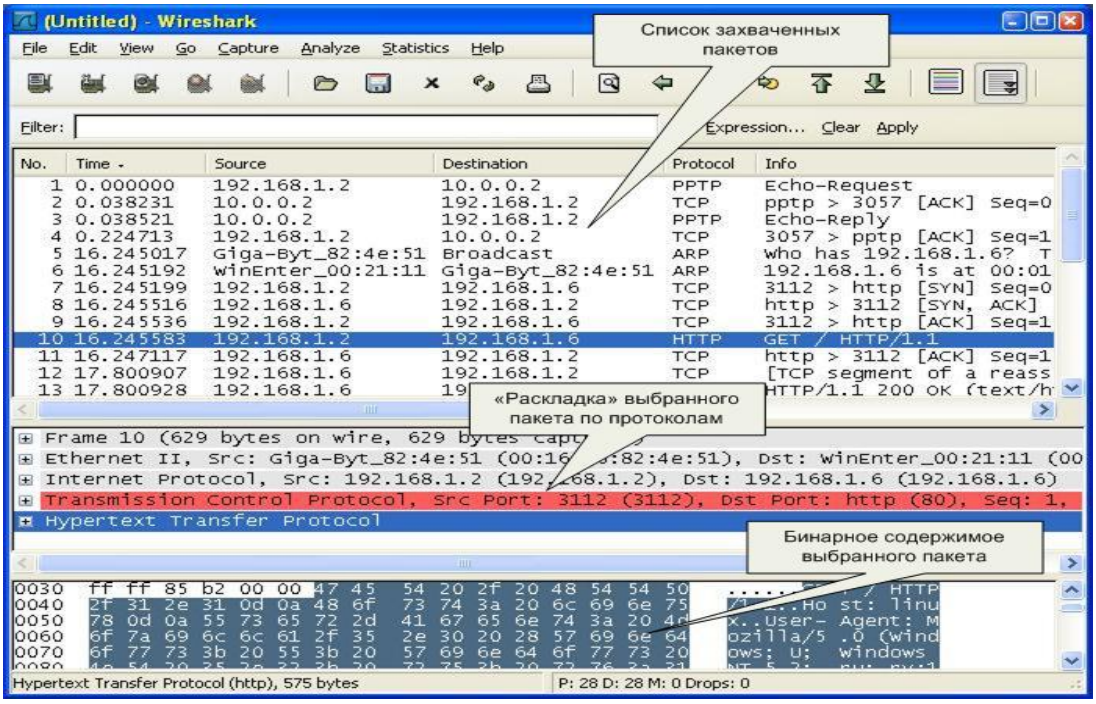


Рис. 2. Основное окно сниффера Wireshark

Каждая строка содержит следующие поля (по умолчанию):

- порядковый номер пакета (No.);
- время поступления пакета (Time);
- источник пакета (Source);
- пункт назначения (Destination);
- протокол (Protocol);
- информационное поле (Info).

Список отображаемых полей настраивается в Edit/Perferencis/Columns. Для того, чтобы изменения возымели эффект необходимо перезапустить программу, предварительно нажав кнопку Save.

При нажатии правой кнопки мыши на том или ином пакете, появится контекстное меню. Нажатием на среднюю кнопку мыши можно помечать группу интересующих нас пакетов.

Средняя панель содержит т.н. «дерево протоколов» для выбранного в верхнем окне пакета. В этой панели в иерархическом виде для выбранного в верхнем окне захваченного пакета отображается вложенность протоколов в соответствии с моделью взаимодействия открытых систем OSI. По нажатию на правую кнопку мыши вызывается контекстное меню. При «раскрытии» каждого из протокола нажатием на значек «+» слева, выводятся поля данных соответствующих протоколов.

Нижняя панель содержит шестнадцатеричное представление выбранного пакета. При выборе того или иного поля в средней панели автоматически будет подсвечиваться соответствующий участок 16-ого представления.

Захват пакетов

Для начала захвата пакетов необходимо задать параметры захвата. В частности, указать сетевой интерфейс, с которого и будет осуществляться захват пакетов. Это действие

доступно через меню как «Capture–>Options» или комбинации клавиш CTRL+K (см. рис. 3). Интерфейс, задаваемый в поле ***«Interface:»*** можно выбрать из соответствующего поля. В примере на рис. 3. Показано, что доступны 3 интерфейса: физический сетевой адаптер («Marvel…»), и интерфейсы для виртуальных каналов, в частности, установленного VPN-соединения («WAN (PPP/SLIP)…»). В большинстве случае подходит выбор интерфейса сетевого адаптера.

В качестве дополнительных параметров захвата можно указать следующие:

- ***«Capture Filter»*** – фильтр захвата (будем рассматривать далее). По нажатию на соответствующую кнопку можно применить тот или иной фильтр отбора (из ранее сохраненных). Если таковых не имеется, его можно указать явно в строке редактирования.
- ***«Update list of packets in real time»*** **–** обновление списка захваченных пакетов в режиме реального времени.
- ***«Stop Capture»*** – набор параметров, позволяющих задать то или иное значение при достижении, которого процесс захвата пакетов прекратится.
- ***«Name Resolution»*** – набор параметров разрешения имен позволяет определить какие из способов разрешения имен должны использоваться.

Для начала мониторинга сетевой активности нужно нажать «Start». После выбора интерфейса, который нас интересует, в дальнейшем можно начинать и останавливать захват пакетов через соответствующие команды в меню «Capture».

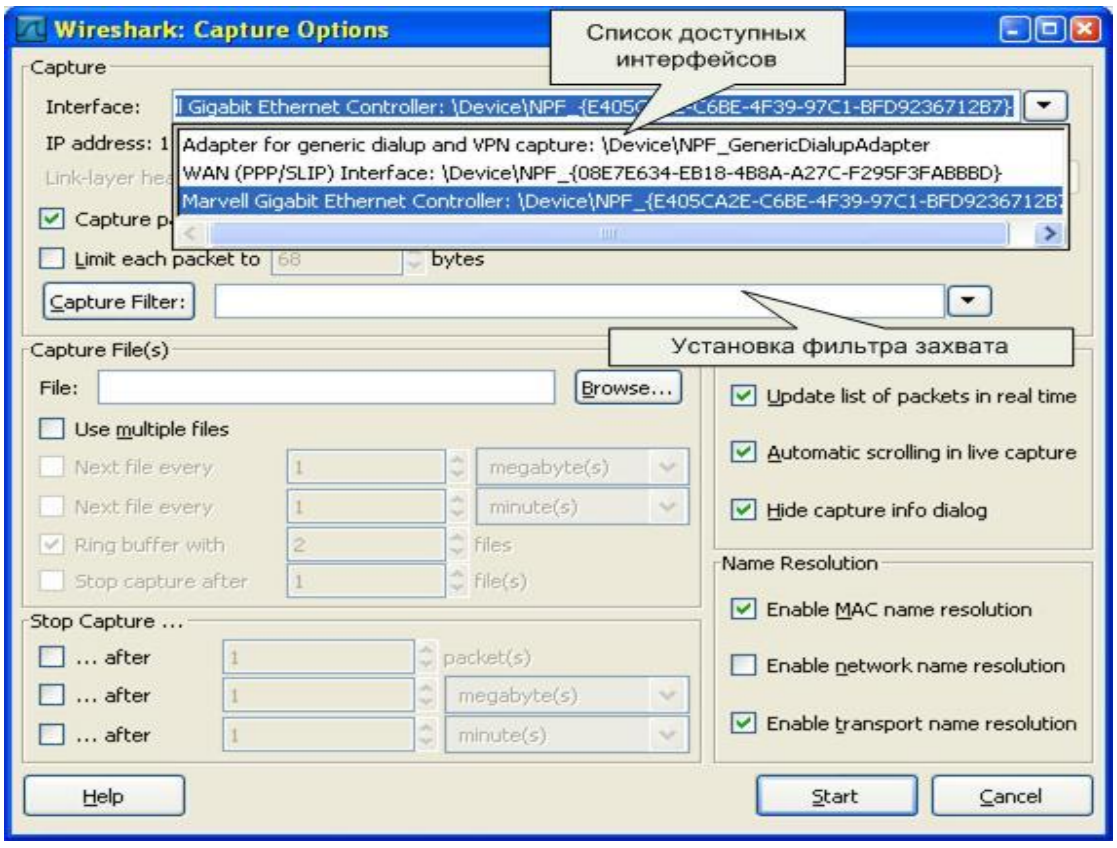


Рис. 3. Выбор интерфейса и параметров захвата пакетов

Фильтрация пакетов

Если запустить сниффер без дополнительных настроек, он будет «захватывать» все пакеты, проходящие через сетевые интерфейсы (см. рис. 1). Вообще, для общего

ознакомления с процессами, происходящими в сети, очень полезно пронаблюдать активность сетевых протоколов в реальных условиях работы системы в сети. Пронаблюдать все разнообразие протоколов, запросов, ответов и др. событий.

При целенаправленном использовании сниффера очень часто необходимо выборочно отображать или захватывать пакеты по некоторым заданным критериям. Для этих целей служат фильтры отображения и захвата, соответственно.

Типы фильтрации трафика

Существует два варианта фильтрации пакетов: на этапе захвата и на этапе отображения пользователю. В первом случае эффективность работы сниффера и потребляемые им системные ресурсы значительно ниже, нежели во втором случае. Это объясняется тем, что при достаточно интенсивном сетевом трафике и и продолжительном времени захвата все пакеты должны бить захвачеты и сохранены либо в память, либо на дисковое устройство. Самые простые подсчеты могу показать, что даже для 100-мегабитной сети системных ресурсов хватит на непродолжительное время. Фильтрация захвата уже на момент получения пакета гораздо эффективнее, однако в таком случае она должна быть реализована на уровне самих драйверов захвата. Данный факт, естественно, усложняет реализацию сниффера. Wireshark поддерживает оба варианта фильтрации. Рассмотрим

Фильтры отображения

Фильтры отображения представляют собой достаточно мощное средство отображения трафика. Фильтры задаются в строке, располагающейся вверху основного экрана («Filter:»). Простейший фильтр отображения, позволяет отобрать пакеты по тому или иному протоколу. Для этого в строке требуется указать название протокола (например HTTP) и нажать кнопку «Apply». После этого в верхнем окне останутся пакеты, принадлежащие этому протоколу. Кнопкой «Reset» действие фильтра отменяется.

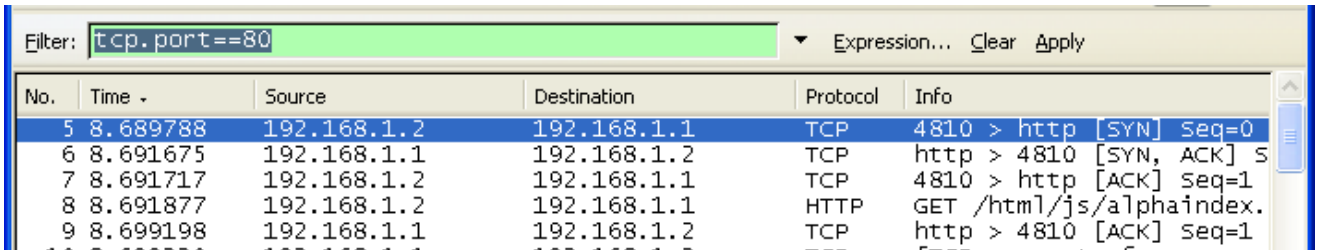
Для работы с фильтрами можно вызвать окно «Analyze/Display Filters». Можно сохранять созданные выражения под определенными именами для последующего использования и т.д.

С помощью логических операций (синтаксис языка Си) можно составлять логические выражения. Логическая истина - 1, ложь - 0.

Список поддерживаемых логических операций:

eq	==	равенство
ne	!=	не равно
gt	>	больше чем
Lt	<	меньше чем
ge	>=	больше равно
Le	<=	меньше равно

Например: tcp.port == 80 (см. рис. 4).



Мастер построения фильтров отображения доступен через кнопку «Expression…» (см. рис. 5).

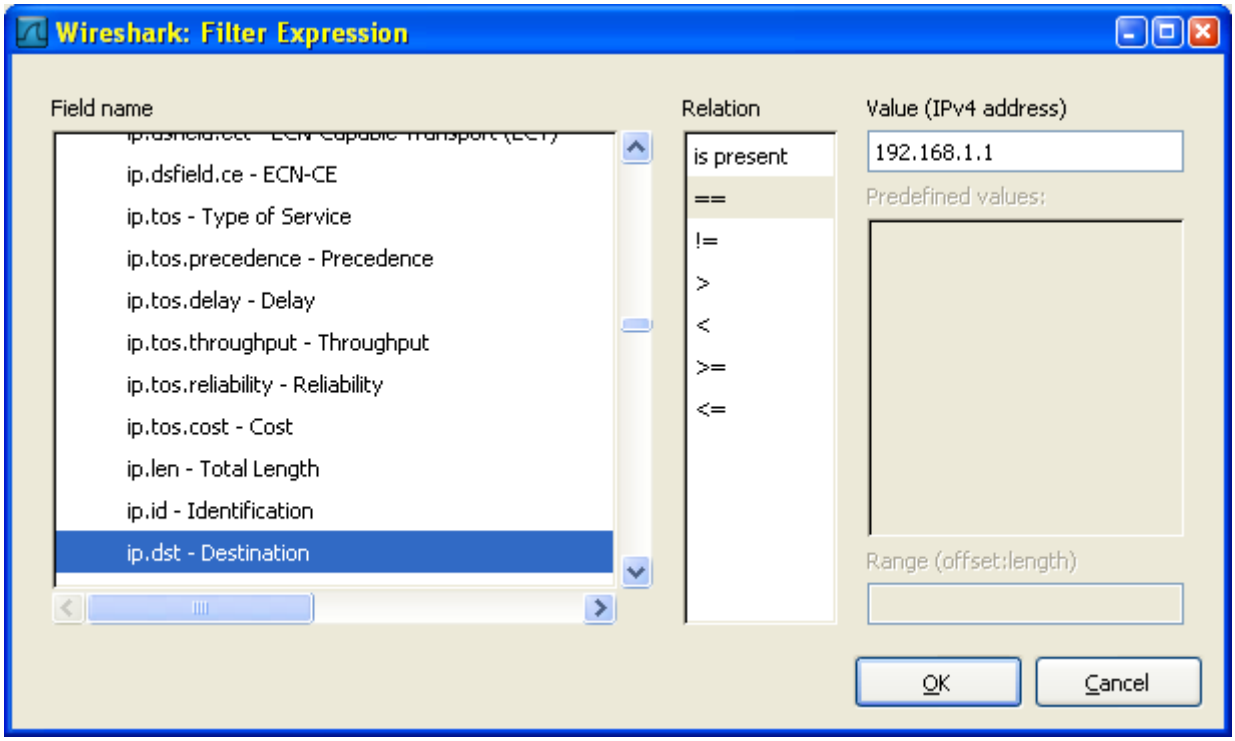


Рис. 5. Построение фильтров отображения

Фильтры захвата

С помощью данных фильтров можно захватывать из сети только те пакеты, которые подходят под критерий отбора. Если не задано никакого фильтра (по умолчанию), то будут захватываться все пакеты. В противном случае только пакеты, для которых указанное выражение будет истинным. Синтаксис фильтров захвата несколько отличается от синтаксиса фильтров отображения. Выражение состоит из одного или более примитивов

разделенных пробельными символами. На рис. 6 приведен пример фильтра для захвата пакетов, адресованных на 80-й порт (http) узла с ip-адресом 10.197.0.11.

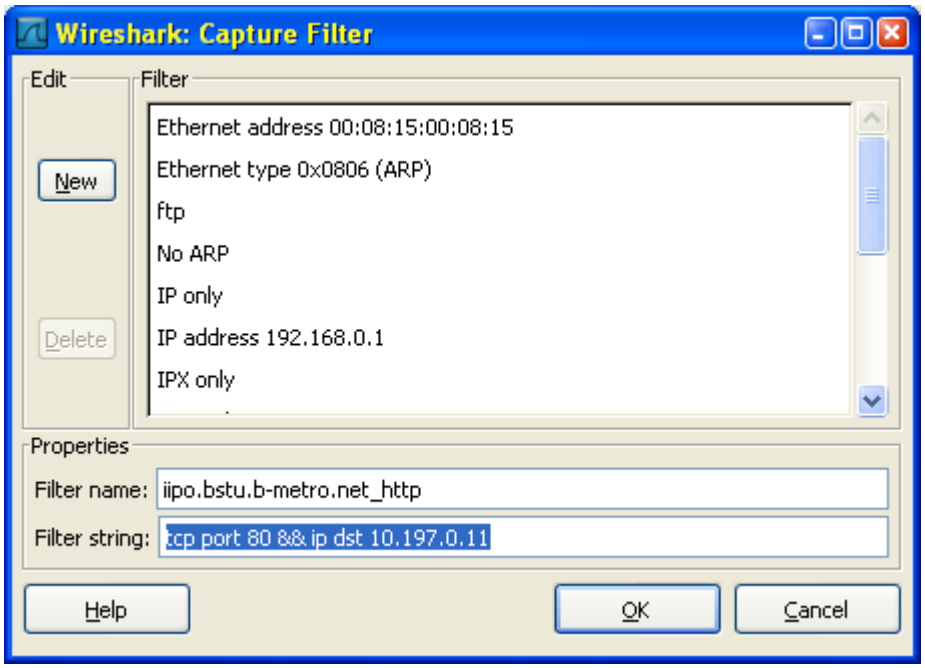


Рис. 6. Пример фильтра захвата

Существует три различных типа примитивов: ***type, dir, proto***.

Спецификатор ***type*** определяет тип параметра. Возможные параметры: **host**; **net**; **port**.

Например:

- host linux
- net 192.168.128
- port 80

Если не указано никакого типа предполагается что это параметр **host**.

Спецификатор ***dir*** определят направление передачи. Возможные направления: **src**; **dst**; **src or dst**; **src and dst**.

Например:

- src linux
- dst net 192.168.128
- src or dst port http

Если не определено направление то предполагается направление «**src or dst**». Для протоколов типа point-to-point используются спецификаторы inbound и outbound.

Спецификатор ***proto*** определят тип протокола, которому принадлежит пакет.

Возможные протоколы: **ether**; **fddi**; **tr**; **ip/ipv6**; **arp/rarp**; **decent**; **tcp**; **udp**.

Например:

- ether src linux
- arp net 192.168.128
- tcp port 80

Если протокол не определен, то будут захватываться пакеты всех протоколов. То есть: «src linux» означает «(ip or arp or rarp) src linux»; «net ctam» означает «(ip or arp or rarp) net ctam»; «port 53» означает «(tcp or udp) port 53».

Также существует несколько специальных спецификаторов, которые не попадают в описанные выше случаи:

- *gateway*;

- *broadcast*;
- *less*;
- *greater*;
- *арифметические выражения*.

Сложные фильтры захвата строятся с использованием логических выражений.

Список операций:

not	!	отрицание
and	&&	конкатенация (логическое И)
or	||	альтернатива (логическое ИЛИ)

Примеры фильтров захвата

Ниже рассмотрены некоторые примеры построения фильтров захвата.

- Захват всех пакетов на сетевом интерфейсе хоста 192.168.1.2:
  host 192.168.1.2
- Захват трафика между хостом host1 И хостами host2 ИЛИ host3:
  host host1 and (host2 or host3)
- Захват всех IP-пакетов между хостом host1 и каждым хостом за исключением hostX:
  ip host host1 and not hostX
- Захват пакетов ни сгенерированных ни адресованных локальными хостами:
  ip and not net localnet
- Захват IP-пакетов размером больше чем 576 байт, проходящих через шлюз snup:
  gateway snup and ip[2:2] > 576
- Захват всех ICMP пакетов, за исключением пакетов ping:
  icmp[0] != 8 and icmp[0] != 0

Статистическая обработка сетевого трафика

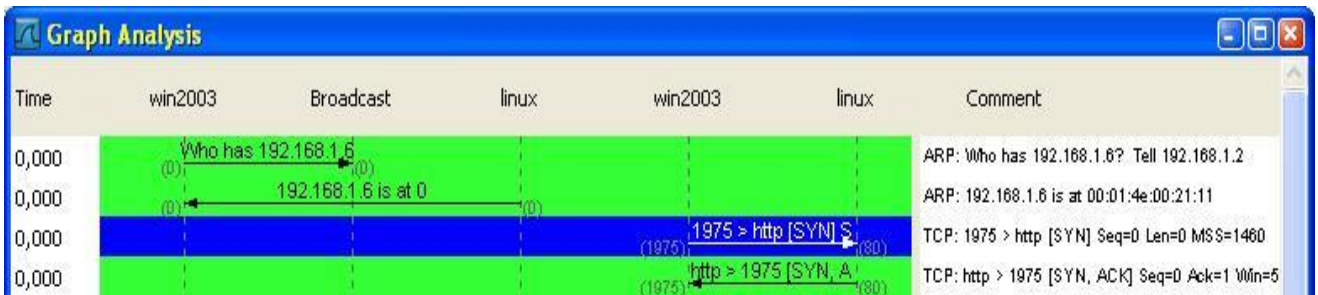
Сниффер Wireshark позволяет выполнять различную статистическую обработку полученных данных. Все доступные операции находятся в меню «Statistics».

Общая статистика – количество полученных/переданных пакетов, средняя скорость передачи и т.д. доступны через пункт «Statistics->Summary».

Получить информацию по статистике обработанных протоколов в полученных пакетах можно через пункт «Statistics->Protocol Hierarchy».

Статистику по типу ip-пакетов, их размеру и порту назначения можно получить выбрав подпункты меню «IP-address…», «Packet length» и «Port type» соответственно.

Одной из наиболее интересных возможностей является генерация диаграммы взаимодействия между узлами, которая доступна пунктом меню «Flow Graph…» . В результате можно наблюдать в достаточно наглядной форме процесс взаимодействия на уровне протоколов. Например, на рис. 7 приведена диаграмма взаимодействия при получении узлом win2003 статической web-странички с сервера http://linux.



Заключение

Программы-снифферы – это незаменимый инструмент для изучения того, что происходит в сети. Если знать, что в действительности посылается или принимается «по проводам», то трудные, на первый взгляд, ошибки удается легко найти и исправить. Сниффер представляет собой также важный инструмент для исследований динамики сети, а равно средство обучения.

ПРИМЕР ИССЛЕДОВАНИЯ ПРОТОКОЛА С ИСПОЛЬЗОВАНИЕМ СНИФФЕРА

В качестве примера исследования некоторого протокола с использованием сниффера рассмотрим протокол ARP.

Протокол ARP

ARP (англ. Address Resolution Protocol — протокол разрешения адресов) — сетевой протокол, предназначенный для преобразования IP-адресов (адресов сетевого уровня) в MAC-адреса (адреса канального уровня) в сетях TCP/IP. Он определён в ***RFC 826***.

Данный протокол очень распространенный и чрезвычайно важный. Каждый узел сети имеет как минимум два адреса, физический адрес и логический адрес. В сети Ethernet для идентификации источника и получателя информации используются оба адреса. Информация, пересылаемая от одного компьютера другому по сети, содержит в себе физический адрес отправителя, IP-адрес отправителя, физический адрес получателя и IP-адрес получателя. ARP-протокол обеспечивает связь между этими двумя адресами. Существует четыре типа ARP-сообщений: ARP-запрос (ARP request), ARP-ответ (ARP reply), RARP-запрос (RARP-request) и RARP-ответ (RARP-reply). Локальный хост при помощи ARP-запроса запрашивает

физический адрес хоста-получателя. Ответ (физический адрес хоста-получателя) приходит в виде ARP-ответа. Хост-получатель, вместе с ответом, шлет также RARP-запрос, адресованный отправителю, для того, чтобы проверить его IP адрес. После проверки IP адреса отправителя, начинается передача пакетов данных.

Перед тем, как создать подключение к какому-либо устройству в сети, IP-протокол проверяет свой ARP-кеш, чтобы выяснить, не зарегистрирована ли в нём уже нужная для подключения информация о хосте-получателе. Если такой записи в ARP-кеше нет, то выполняется широковещательный ARP-запрос. Этот запрос для устройств в сети имеет следующий смысл: «Кто-нибудь знает физический адрес устройства, обладающего следующим IP-адресом?» Когда получатель примет этот пакет, то должен будет ответить: «Да, это мой IP-адрес. Мой физический адрес следующий: ...» После этого отправитель обновит свой ARP-кеш, и будет способен передать информацию получателю.

RARP (англ. Reverse Address Resolution Protocol – обратный протокол преобразования адресов) – выполняет обратное отображение адресов, то есть преобразует аппаратный адрес в IP-адрес.

Протокол применяется во время загрузки узла (например компьютера), когда он посылает групповое сообщение-запрос со своим физическим адресом. Сервер принимает это сообщение и просматривает свои таблицы (либо перенаправляет запрос куда-либо ещё) в поисках соответствующего физическому IP-адреса. После обнаружения найденный адрес отсылается обратно на запросивший его узел. Другие станции также могут «слышать» этот диалог и локально сохранить эту информацию в своих ARP-таблицах.

RARP позволяет разделять IP-адреса между не часто используемыми хост-узлами. После использования каким либо узлом IP-адреса он может быть освобождён и выдан другому узлу. RARP является дополнением к ARP, и описан в ***RFC 903***.

Для просмотра ARP-кеша можно использовать одноименную утилиту arp с параметром «-a». Например:

```
D:\>arp -a
Interface: 192.168.1.2 --- 0x10003
 Internet Address   Physical Address   Type
 192.168.1.1      00-15-e9-b6-67-4f   dynamic
   192.168.1.6      00-01-4e-00-21-11   dynamic
```

Из данного результата команды arp видно, что в кеше на данный момент находится 2 записи и видны соответственно ip-адреса машин и MAC-адреса их сетевых адаптеров.

Записи в ARP-кеше могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кеша. Хост-отправитель автоматически послал запрос получателю, не уведомляя при этом пользователя. Записи в ARP-кеш можно добавлять вручную, создавая статические (static) записи кеша. Это можно сделать при помощи команды:

arp -s <IP адрес> <MAC адрес>

Также можно удалять записи из ARP-кеша. Это осуществляется путем следующего вызова:

arp -d <IP адрес>

После того, как IP-адрес прошёл процедуру разрешения адреса, он остается в кеше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кеше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кеше просуществует до 10 минут. После этого запись будет удалена из кеша и будет отправлен повторный ARP-запрос.

ARP изначально был разработан не только для IP протокола, но в настоящее время в основном используется для сопоставления IP- и MAC-адресов.

Посмотрим же на практике как работает протокол ARP/RARP. Для этого воспользуемся сниффером для захвата сетевого трафика.

Рассмотрим пример работы протокола ARP при обращении к машине с адресом 192.168.1.5, выполнив запрос с машины с адресом 192.168.1.2. Для успешного эксперимента предварительно очистим arp-кеш командой

arp -d 192.168.1.5

Для фильтрации ARP/RARP трафика воспользуемся фильтром захвата. В нашем случае это будет простой фильтр

arp or rarp

Далее запустим захват трафика командой «Start» и выполним обращение к заданной машине, например, «пропинговав» ее:

D:\>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:

Reply from 192.168.1.5: bytes=32 time<1ms TTL=64

Reply from 192.168.1.5: bytes=32 time<1ms TTL=64

Reply from 192.168.1.5: bytes=32 time<1ms TTL=64

Reply from 192.168.1.5: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.5:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Так как на момент начала работы утилиты ping в arp-кеше не было информации о MAC-адресе соответствующего узла, то первоначально система должна выполнить определение это самого MAC-адреса, сгенерировав ARP-запрос и отослав его в сеть широковещательным пакетом. После чего она будет ожидать ответа от заданного узла.

Посмотрим же, что мы получим на практике. После остановки сниффера мы должны увидеть результат схожий с тем, что отображен на рис. 8. В нашем случае мы видим 2 захваченных пакета: ARP-запрос и ARP-ответ.

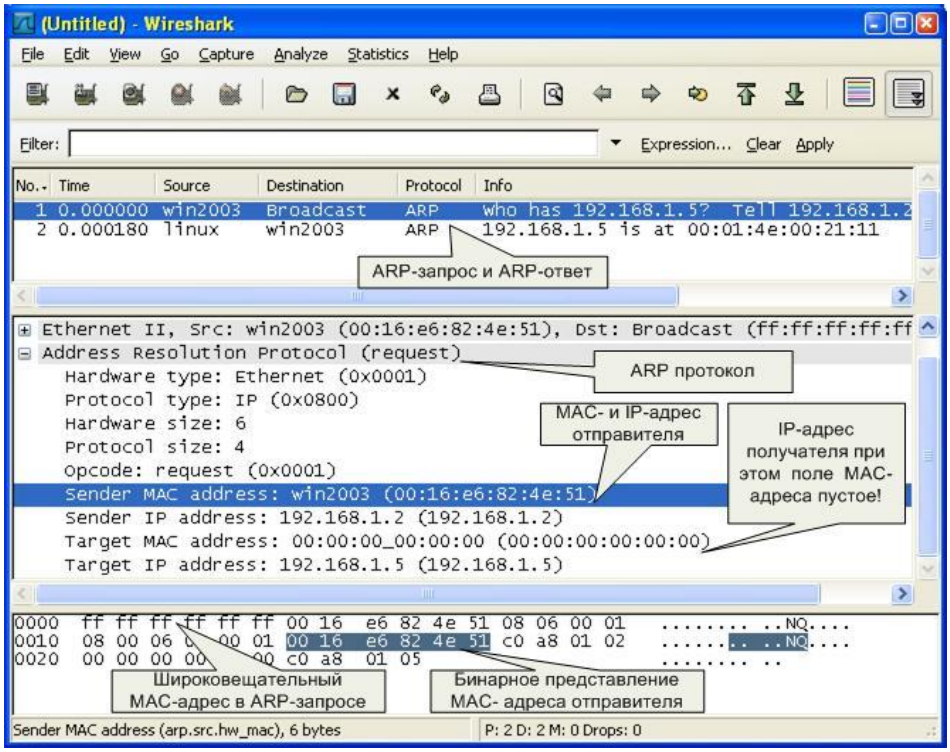


Рис. 8. Анализ ARP-запроса

Проанализируем полученные пакеты. Сначала рассмотрим ARP-запрос (пакет №1). Выделив пакет курсором, мы получаем его раскладку по протоколам (Ethernet+ARP) в среднем окне. Wireshark очень наглядно «раскладывает» заголовок протокола по полям.

Мы можем видеть, что в пакете указаны MAC- и IP-адреса отправителя («Sender MAC address» и «Sender IP address» соответственно). Это параметры машины, с которой выполняется запрос. В данном случае запрос направлен на получения («Opcode: request» – запрос) MAC-адреса машины, у которой IP-адрес («Protocol type: IP») 192.168.1.5 («Target IP address»). При этом поле «Target MAC address» обнулено. Так как получатель ARP-запроса на момент запроса не известен, Ethernet-пакет отправляется всем машинам в данном локальном сегменте, о чем сигнализирует MAC адрес Ethernet-пакета «ff:ff:ff:ff:ff:ff».

Примечание. Обратите внимание, что пакет представляет собой бинарную последовательность и сниффер выполняет большую работу по преобразованию полей из бинарного представления в удобочитаемый вариант.

Все работающие машины в сети получают пакет с ARP-запросом, анализируют его, а ответ отсылает только та машина, чей IP-адрес соответствует IP-адресу в запросе. Таким образом, второй полученный пакет является ARP-ответом (см. рис. 9). Это следует из параметра поля «Opcode: reply». Обратите внимание, что данный пакет был отправлен именно той машиной, чей MAC-адрес нас и интересовал («Sender IP address: 192.168.1.5»). При этом поле «Sender MAC address» заполнено значением «00:01:4E:00:21:11».

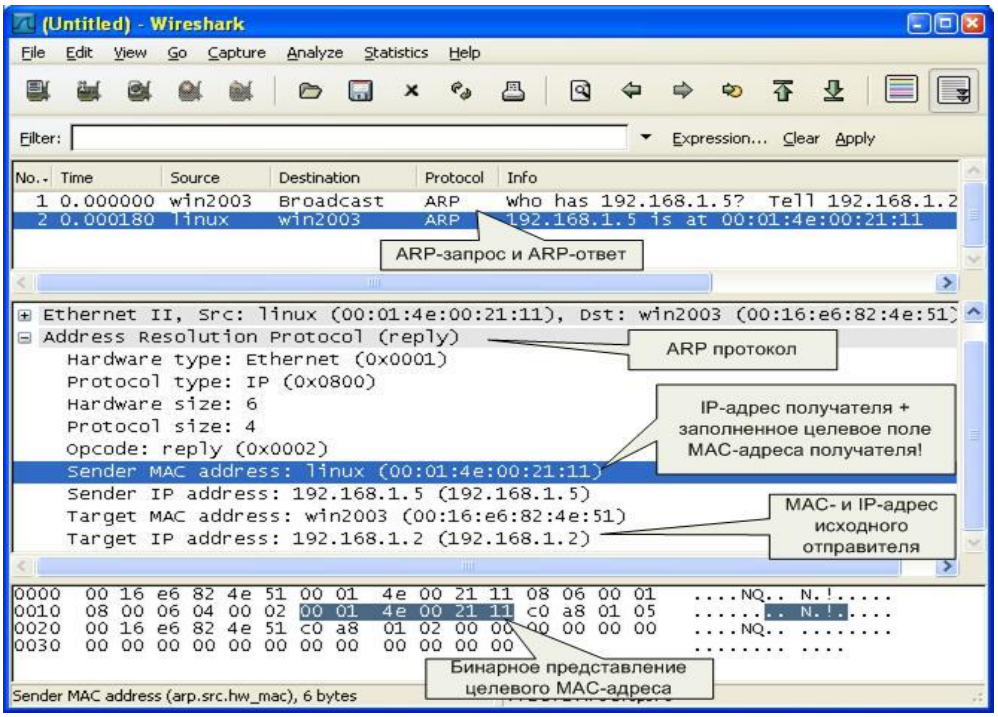


Рис. 9. Анализ ARP-ответа

Примечание. Обратите внимание на поле «Info» в списке захваченных пакетов. Сниффер и тут упрощает анализ сетевого трафика, подсказывая назначение пакетов ☺

Теперь мы можем повторно просмотреть ARP-кеш и сверить данные в нем с данными, которые мы узнали из анализа пакетов ARP-запрос/ответа:

```
D:\>arp -a
Interface: 192.168.1.2 --- 0x10003
 Internet Address   Physical Address   Type
 192.168.1.5      00-01-4e-00-21-11   dynamic
```

Стоит также отметить, что в реальных условиях в локальной сети с большим количеством машин arp/rarp трафик бывает гораздо более интенсивным.

Задание на практическую работу

1. Изучить интерфейс программы Wireshark (\\corp.mgkit.ru\dfs\work\wireshark)
2. Захватить 100 произвольных пакетов. Определить статистические данные:
   - процентное соотношение трафика разных протоколов в сети;
   - среднюю скорость кадров/сек;
   - среднюю скорость байт/сек;
   - минимальный, максимальный и средний размеры пакета;
   - степень использования полосы пропускания канала (загрузку сети).
3. Зафиксировать 20 IP-пакетов. Определить статистические данные:
   - процентное соотношение трафика разных протоколов стека tcp/ip в сети;
   - средний, минимальный, максимальный размеры пакета.
4. Выполнить анализ ARP-протокола по примеру из методических указаний.
5. На примере любого IP-пакета указать структуры протоколов Ethernet и IP, Отметить поля заголовков и описать их.
6. Проанализировать и описать принцип работы утилиты *ping*. При этом описать все протоколы, используемые утилитой. Описать все поля протоколов. Составить диаграмму взаимодействия машин при работе утилиты *ping*.

Контрольные вопросы

1. Каковы основные цели мониторинга сетевого трафика?
2. Чем отличается мониторинг трафика от фильтрации?
3. Каково назначении класса программ-снифферов?
4. Какие основные функции выполняют снифферы?
5. Зачем используются фильтры отображения и фильтры захвата сниффера Wireshark? В чем их отличие?
6. Какие базовые функции статистической обработки захваченных пакетов имеет сниффер Wireshark?
7. Какие задачи рассчитан решать протокол ARP?

Лабораторная работа 3 Анализ и исследование TCP/IP соединений

Понятие и назначение прокси-серверов.
Принцип работы прокси-сервера.
Настройка Nat-прокси на базе Windows Server.

ЦЕЛЬ РАБОТЫ

1. Выяснить назначение принцип работы Proxy-серверов.
2. Изучить механизм работы NAT-прокси.
3. Уметь настраивать NAT-прокси на базе Windows Server.

ВВЕДЕНИЕ

Одной из важных задач при организации работы локальных сетей (и не только локальных) является организация доступа к ресурсам глобальных сетей (в частности, *Internet*). Чаще всего реализовать отдельный канал в глобальную сеть для каждой клиентской машины не представляется возможным. В таких случаях задача решается организацией канала через некоторую машину (в дальнейшем просто *сервер*), которая имеет сетевой интерфейс в глобальную сеть, с установленным специализированным ПО.

На данной лабораторной работе мы с вами познакомимся с механизмами реализации доступа во внешние сети.

БАЗОВЫЕ ПОНЯТИЯ

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 какой-либо 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 кеша (в случаях, если прокси имеет свой кеш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях.

Прокси-сервера используют для того, чтобы обеспечить эффективный и безопасный доступ в Интернет. Их устанавливают в различных организациях для обеспечения взаимодействия локальной сети с глобальной сетью Интернет. 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 (например, модема), 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.

Важное отличие прокси от маршрутизатора (в IP-сетях) в том, что при использовании маршрутизатора IP-пакеты остаются без изменений – в них сохраняются исходные IP–адреса компьютеров как отправителя так и получателя. А прокси всегда работает от своего адреса, а адреса клиентов не передаются, т.к. чаще всего являются локальными и невалидными в сети Internet.

На рис. 1. представлена простейшая схема сети с прокси сервером.

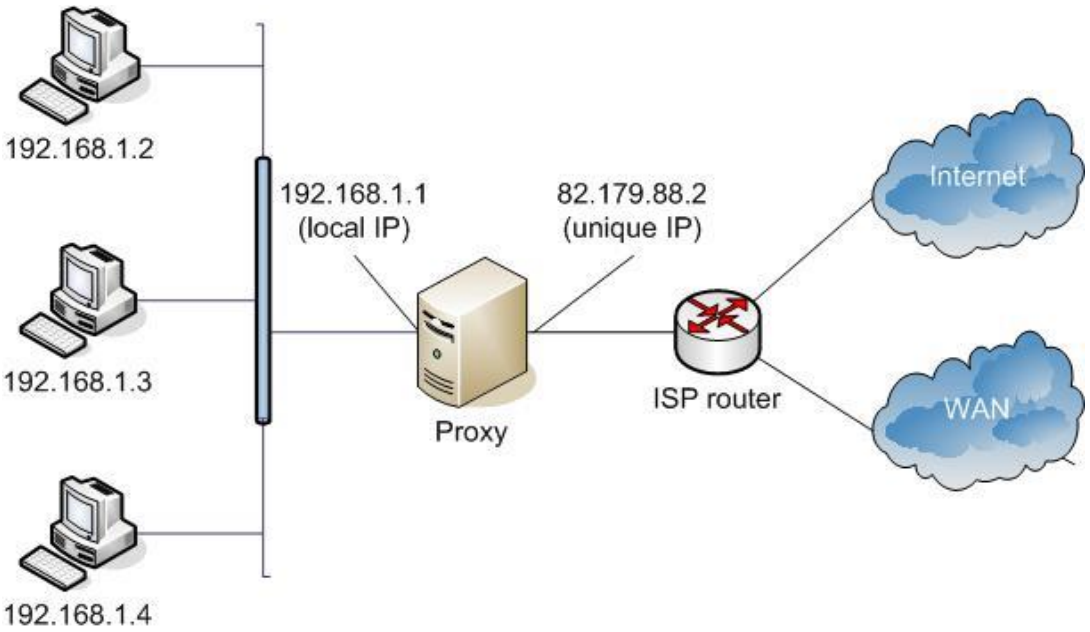


Рис. 1. Схема сети с прокси-сервером

Виды и функции прокси-серверов

Мы дали обобщенное понятие прокси–серверов. Давайте теперь рассмотрим, какие конкретные реализации прокси существуют и в чем их особенность.

NAT-прокси (Network Address Translation) – самый простой вид прокси. Это, по сути, специализированный маршрутизатор, подменяющий адреса. Простейшая реализация NAT входит в состав Windows 2000+. Она называется «Общий доступ к подключению интернета» (Internet Connection Sharing) и включается галочкой в свойствах модемного соединения. Этот прокси работает прозрачно для пользователя, никаких специальных настроек в программах не требуется. Но он достаточно ограничен в возможностях. Другие реализации NAT-прокси 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. Далее мы более детально мы рассмотрим механизм NAT.

Специализированные прокси. В общем случае для каждого их сетевых прикладных протоколов необходимо реализовывать свои прокси. Обычно прокси – это сетевой сервис, который настраивается для работы на некоторый порт. Клиентские программы должны знать этот порт и, естественно, уметь работать с прокси. Далее приведены некоторые специализированные типы прокси и их функции.

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

- Возможно сохранение полученных файлы на диске сервера. Впоследствии, если запрашиваемый файл уже скачивался, то при повторном запросе можно «мгновенно» выдать его с диска без обращения в Интернет, что приводит также к экономии внешний трафик. Эта возможность называется кэшированием. Прокси с поддержкой данной функциональности называют кеширующими. Часто это одна из наиболее приоритетных функций. Обычно на практике экономия трафика достигает порядка 10-15%.
- Можно ограничивать доступ к ресурсам. В частности, производить авторизацию пользователя при запросе ресурсов. Также можно 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. Условия и действия в принципе могут быть достаточно сложными.
- 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. 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.

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 HTTP-прокси, 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 ftp-клиенты предпочитают и специальный прокси (FTP-gate), работающий с обеими сторонами по FTP. Хотя встречаются и вносящие путаницу названия. Например, в программе CuteFTP и FAR FTP-gate называют firewall, хотя FireWall в общем случае – это вообще не прокси.

HTTPS-прокси – фактически часть HTTP-прокси. S в названии означает «secure», т.е. безопасный. Не смотря на то, что программно это 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 или на более низком уровне. 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). В частности, в таком случает невозможна и антивирусная фильтрация трафика!!! 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.

Mapping-прокси – это способ заставить работать через прокси те программы, которые умеют работать с интернетом только напрямую. При настройке такого прокси создается как бы «копия» целевого сервера, но доступная через один из портов прокси-сервера для всех клиентов локальной сети. Таким образом, устанавливает локальное «отображение» заданного сервера.

Socks-прокси. Большинство прокси-серверов являются узкоспециализированными и рассчитаны на конкретные прикладные протоколы (см. выше), то есть для новых протоколов требуются новые прокси-серверы.

Socks (современная версия Socks5) обладает двумя ключевыми особенностями, выделяющими его из группы прокси-серверов:

- Он не зависит от высокоуровневых протоколов (HTTP, FTP, POP3, SMTP, NNTP и т. д.), так как осуществляет представительство клиентов на более низком уровне (TCP и UDP).
- Приложение в локальной сети может попросить Socks-сервер выступить в роли сервера от лица клиента. То есть приложение в локальной сети сможет принимать соединения извне, несмотря на отсутствие реального IP-адреса.

Через Socks можно заставить работать даже приложения, которые и понятия не имеют о прокси. Фактически Socks-сервер является программно-управляемым mapping-proxy, причем с описанным единым интерфейсом. Все mapping-proxy так или иначе программно управляются, но под руководством администратора сети (человека), и отображения статичны. А Socks-сервер управляется прикладными программами, и отображения устанавливаются, только когда они нужны, и на то время, пока они нужны.

Прозрачный прокси-сервер

Данная функциональность поддерживается большинством современных прокси-серверов. До внедрения этой функции пользователи локальной сети должны были указывать адрес и порт прокси в LAN настройках браузера. При использовании прозрачного прокси запросы незаметно для пользовательских приложений транслируются на стороне сервера порт прокси-сервера. В остальном работа ни чем не отличается от работы с обычным прокси-сервером.

Открытый прокси-сервер

В отличие от обычных прокси-серверов, которыми пользуются ограниченное количество лиц (обычно в зоне ответственности владельца прокси-сервера), открытый прокси-сервер позволяет практически любому узлу сети обращаться к другим узлам сети, скрывая свой истинный адрес от узла-получателя.

Открытые прокси-сервера могут использоваться для обеспечения (частичной) анонимности в Интернете, так как скрывают IP-адрес пользователя, направляя все пользовательские запросы через себя. Поэтому такие прокси-сервера еще называют анонимными. При этом сам прокси-сервер может вести логи обращений.

Потенциальное использование открытых прокси-серверов для скрытия адреса пользователя приводит к тому, что сайты некоторых интернет-сервисов запрещают доступ к своим ресурсам с открытых прокси-серверов. Например, почтовые службы mail.yandex.ru отказываются работать с пользователями открытых прокси-серверов.

Механизм работы NAT-прокси

NAT – это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Был разработан в 1994 году для использования совместно с протоколом IPv4. Механизм NAT определён в RFC 1631, RFC 2663 и RFC 3022. Преобразование адресов методом NAT может производиться почти любым маршрутизирующим устройством: маршрутизатором, сервером доступа, межсетевым экраном. Например, практически во всех сетевых ADSL-модемах имеется встроенный NAT-прокси.

Суть механизма состоит в замене обратного (source) адреса при прохождении пакета в одну сторону и обратной замене адреса назначения (destination) в ответном пакете. Наряду с адресами source/destination могут также заменяться номера портов source/destination. Информация о произведенной трансляции адресов сохраняется в течении некоторого времени в таблице трансляции. На данный момент существует 3 базовых концепции трансляции адресов: статическая (Static Network Address Translation), динамическая (Dynamic

Address Translation), маскарадная (NAPT, PAT). Пример статической трансляции представлен на
рис. 2.

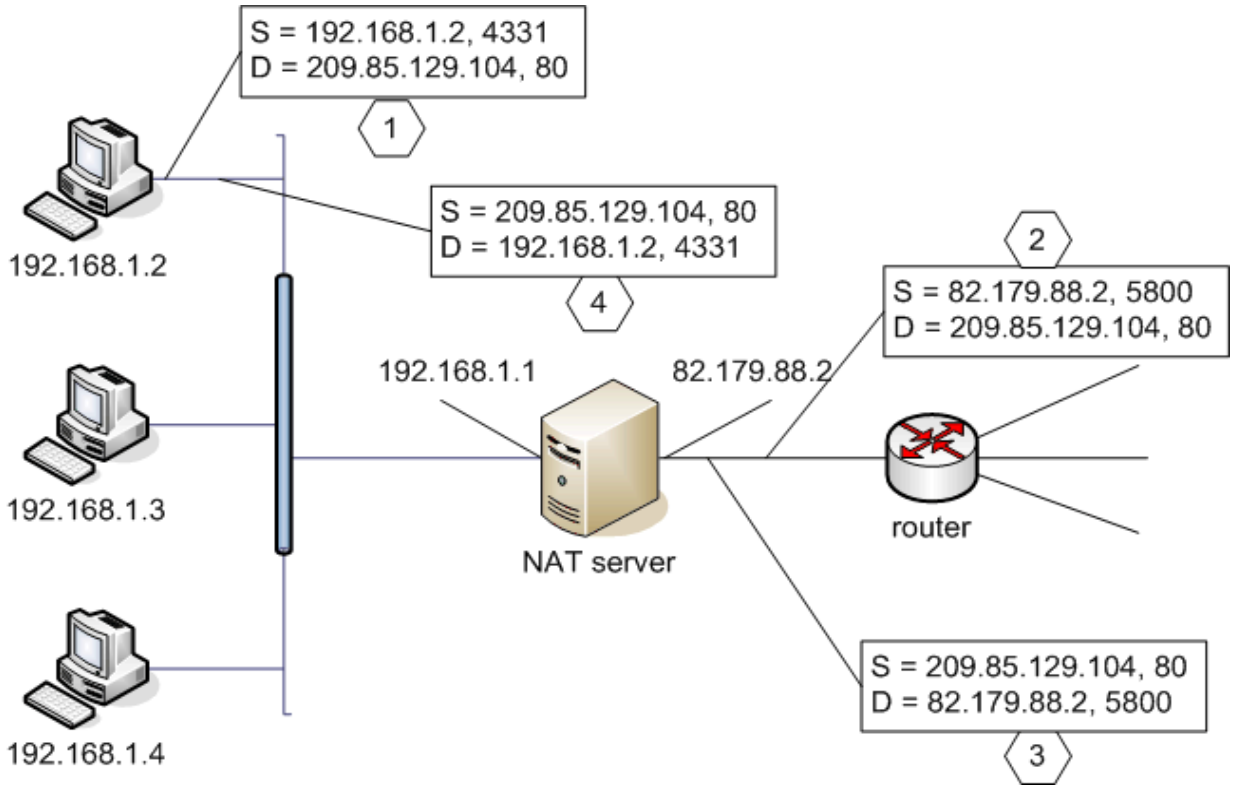


Рис. 2. Принцип работы механизма трансляции адресов

В данном примере у нас имеется локальная сеть, подключенная через NAT-прокси (например, на базе ADSL модема) с локальными адресом 192.168.1.1 и внешним интерфейсом 82.179.88.2 к глобальной сети Internet. Пользователи локальной сети 192.168.1.0/24 имеют прозрачный доступ в *Internet* через данный сервер.

Клиент с адресом 192.168.1.2 обращается к web-серверу по адресу 209.85.129.104 (www.google.com) на 80-м порту. Пакет (1) с порта 4331 отправляется на NAT-прокси (192.168.1.1). Прокси получает пакет, генерирует новый номер порта 5880, которым заменяет оригинальный номер порта отправителя, а IP-адрес клиента заменяет на адрес своего внешнего сетевого интерфейса (82.179.88.2) и отправляет пакет (2) на адрес 209.85.129.104, при этом сохраняя в таблице трансляции (см. таб. 1) данные об IP-адресах и номерах портов. Генерируя новый номер порта, NAT-прокси может выбирать произвольный номер, которого нет в таблице трансляции. Целевой web-сервер, не имея представления о том, что пакет был «проксирован», отправляет на адрес 82.179.88.2 и порт 5800 ответный пакет (3). Получив дейтаграмму, NAT-прокси по указанному в пакете IP-адресу (82.179.88.2) и номеру порта (5800) находит в таблице трансляции исходный адрес клиента (192.168.1.2) и номер порта (4331). После чего заменяет в пришедшем пакете оба поля найденными в таблице и перенаправляет пакет (4) клиенту.

Таблица 1.

Таблица трансляции сетевых адресов	
Сторона глобальной сети	Сторона локальной сети
82.179.88.2, 5800 …	192.168.1.2, 4331 …

NAT выполняет две важных функции:

1. Позволяет сэкономить IP-адреса (актуально для протокола IPv4), транслировав несколько внутренних IP-адресов в один внешний публичный IP-адрес (или в несколько внешних, но меньше, чем внутренних).
2. Позволяет предотвратить или ограничить обращение снаружи ко внутренним хостам, оставляя возможность обращения изнутри наружу. При инициации соединения изнутри сети создаётся трансляция. Ответные пакеты, поступающие снаружи, соответствуют созданной трансляции и поэтому пропускаются. Если для пакетов, поступающих снаружи, соответствующей трансляции не существует (а она может быть созданной при инициации соединения или статической), они не пропускаются.

Сложности при использовании NAT:

1. Необходимо индивидуально отслеживать работу ряда протоколов, например, ICMP (т.к. в данном протоколе отсутствует понятие портов), фрагментированные варианты IP-пакетов.
2. Не все протоколы прикладного уровня могут «преодолеть» NAT. Некоторые не в состоянии работать, если на пути между взаимодействующими хостами есть трансляция адресов. Но большинство современных реализаций NAT справляются с данной ситуацией, соответствующим образом заменяя IP-адреса не только в заголовках IP, но и на более высоких уровнях (например, в командах протоколов FTP или H.323).
3. Из-за трансляции адресов «много в один» появляются дополнительные сложности с идентификацией пользователей. Необходимо хранить полные логи трансляций.
4. Трансляции адресов достаточно сложна при использовании шифрованного IP-трафика (IPSec).

Несмотря на ряд недостатков, трансляция сетевых адресов является важной составляющей Internet. Более детальную информацию о работе NAT можно найти в соответствующих RFC и дополнительных источниках [2-4].

Маскарадная трансляция

Маскарадная трансляция адресов, известная также как NAPT (Network Address Port Translation) или «Virtual Servers» – применяется в случаях, когда необходимо реализовать доступ к локальной машине из глобальной сети. Примерная схема работы NAPT приведен на рис. 3.

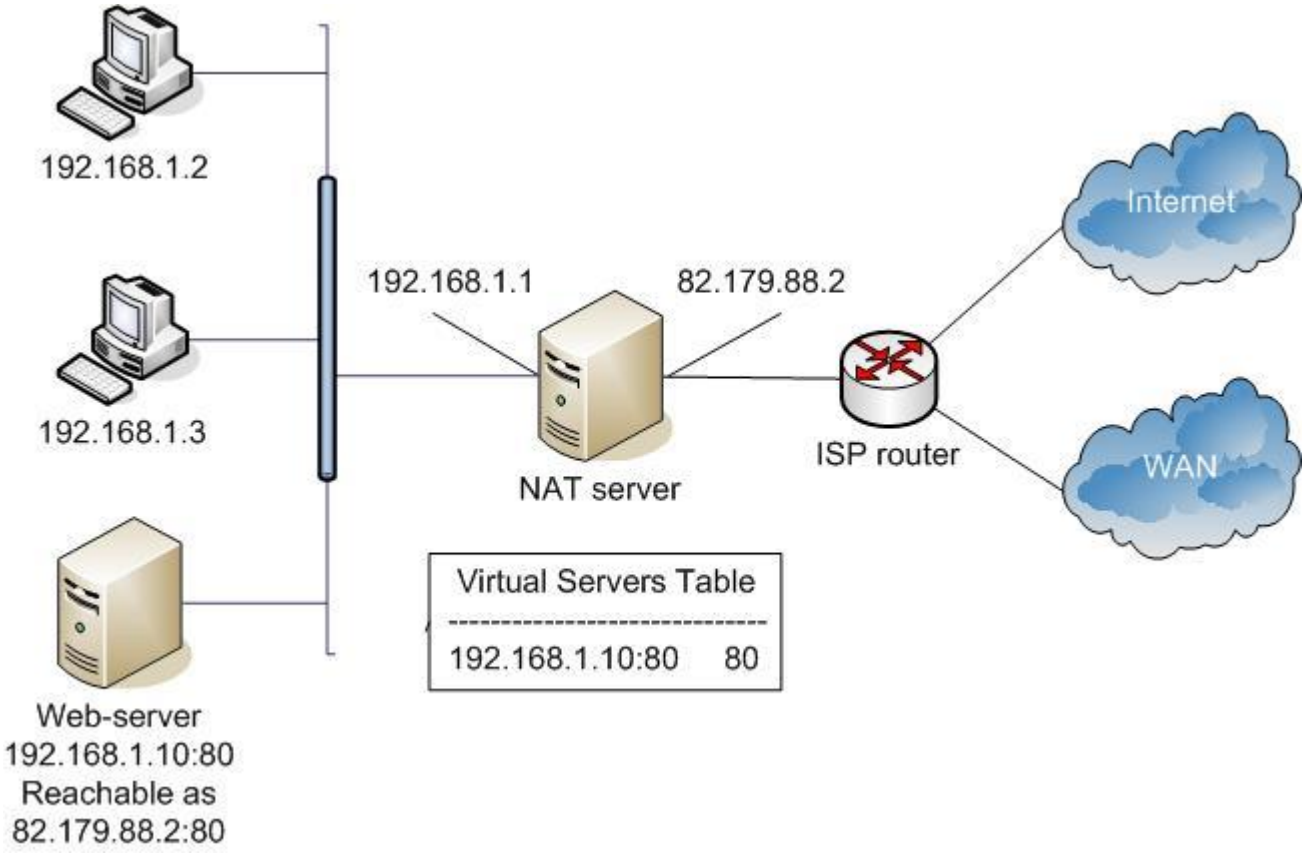


Рис. 3. Схема работы NAPT

На схеме показана ситуация «публикации» локального web-сервера с адресом 192.168.1.10:80 на внешний интерфейс 82.179.88.2:80. В данном случае при включении режима маскарадинга сервер открывает на интерфейсе 82.179.88.2 порт 80 на ожидание входящих подключений. При поступлении запросов на 82.179.88.2:80 NAT-сервер транслирует адреса пакета по рассмотренному выше алгоритму и перенаправляет пакеты на локальную машину с адресом 192.168.1.10 на 80-й порт. В дальнейшем работа севера с клиентов ничем не отличается от механизма NAT.

В общем случае внешний порт и порт локального сервера могут не совпадать (port-translation).

Естественно, данный механизм позволяет одновременно транслировать запросы с одного порта только на одну назначенную машину в локальной сети.

Механизм маскарадинга является классическим приемом, используемым на ADSL-модемах (например, для работы в пиринговых сетях).

NAPT и NAT часто используются совместно.

Реализации Proxy-серверов

На данный момент существует большое количество практических реализаций прокси серверов. Наиболее популярными для платформы Window NT являются ISA Server, Usergate,

WinGate, WinRoute и др. В серверных версиях Windows также имеется встроенный достаточно функциональный NAT-маршрутизатор.

Для платформы Unix/Linux самыми известным прокси-сервером считается squid (http://www.squid-cache.org/).

Настройка и тестирование NAT-прокси на базе Windows 2003 server

В качестве практического примера рассмотрим с вами настройку имеющегося в Windows 2003 Server встроенного NAT-прокси.

Настройку и тестирование прокси-серверов будем выполнять в сети аналогичной рис. 4. (ВНИМАНИЕ: реальные адреса сетевых интерфейсов необходимо выбирать в зависимости от учебной лаборатории! При необходимости нужно проконсультироваться с преподавателем)

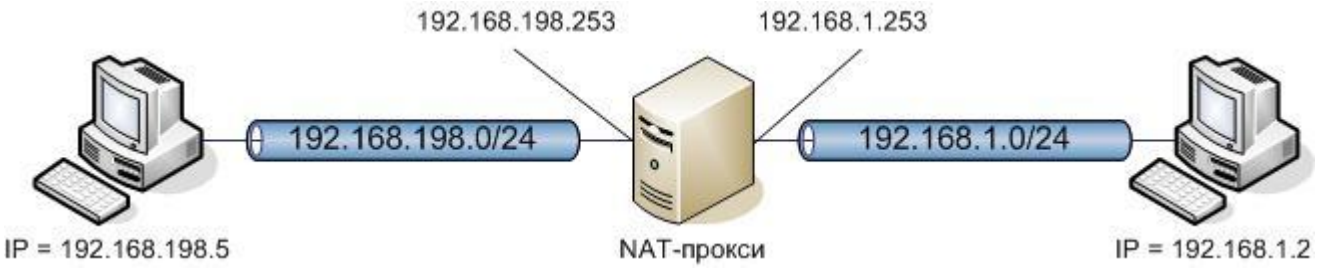


Рис. 4. Простейшая сеть для настройки и тестирования NAT

Поставим задачу следующим образом: необходимо настроить на сервере NAT-прокси таким образом, чтобы для машин из подсети 192.168.198.0/4 (рис. 4, левая подсеть) осуществлялась трансляция адресов в некоторую внешнюю «подсеть» 192.168.1.0/24 (рис. 4, правая подсеть). Дополнительно необходимо настроить на машине 192.168.198.5 telnet или web сервер и посредством технологии NAPT «опубликовать» данные сервисы для доступа из вне.

Настройка NAT-прокси осуществляется при помощи оснастки «Routing and Remote Access» (см. рис. 5).

Настройка NAT достаточна проста и не требует особых пояснений. На рис. 5-6 показано, как добавить функциональность NAT-маршрутизатора на сервер. На рис. 7 показана настройка основных параметров NAT: задание внешнего интерфейса и опциональное включение встроенного сетевого фильтра. В дальнейшем для каждого из интерфейсов можно вызвать диалог настроек (рис. 8). Для внешнего интерфейса настроек больше, в частности, имеется настройка NAPT (см. рис. 9). Пример настройки NAPT для протокола telnet приведен на рис. 10.

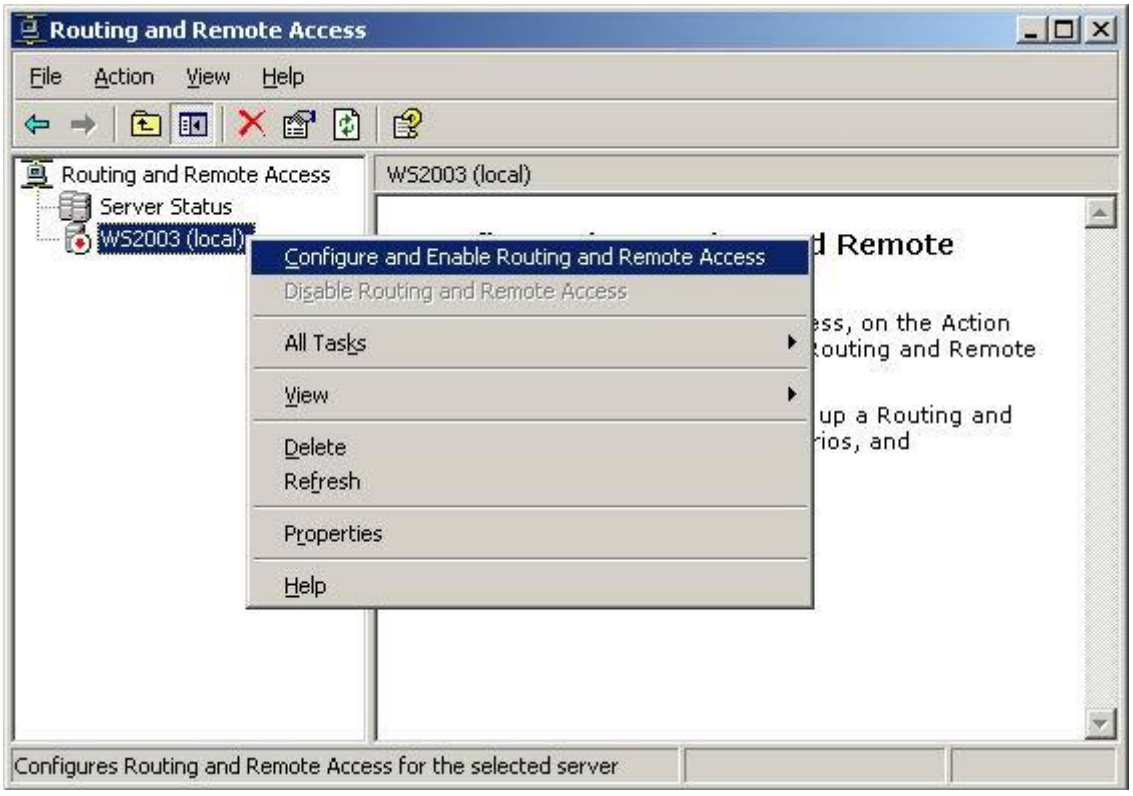


Рис. 5. Добавление NAT-прокси

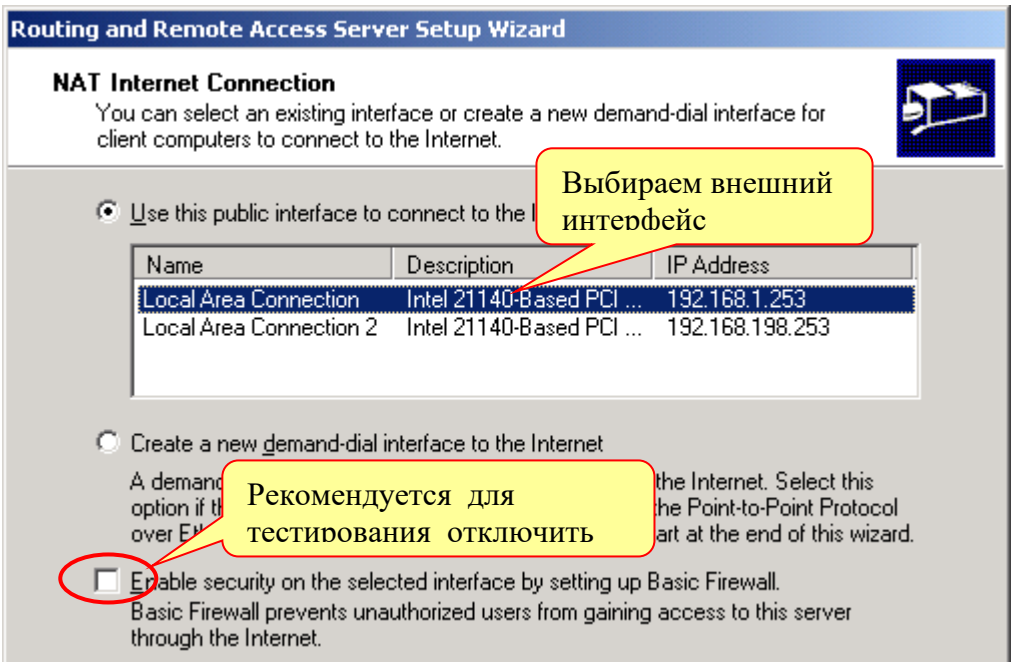


Рис. 7. Задание базовых параметров NAT

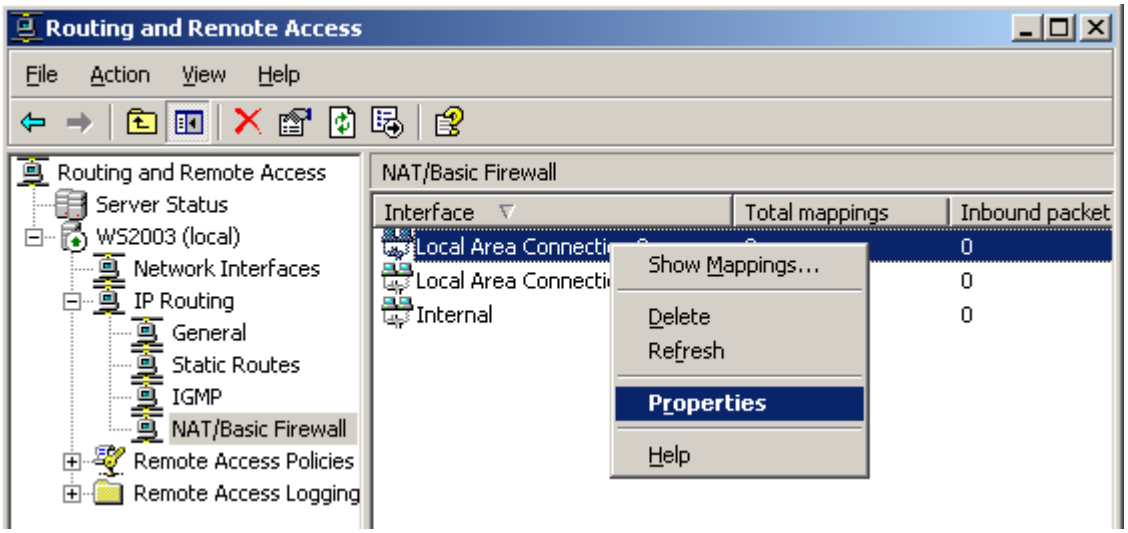


Рис. 8. Доступ к конфигурации NAT

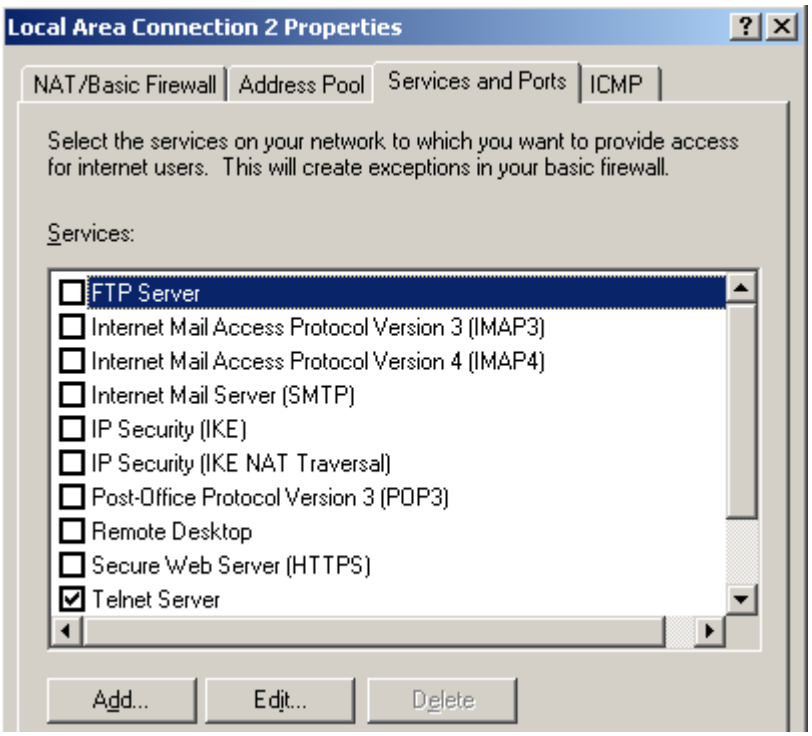


Рис. 9. Настройка NAPT для внешнего интерфейса

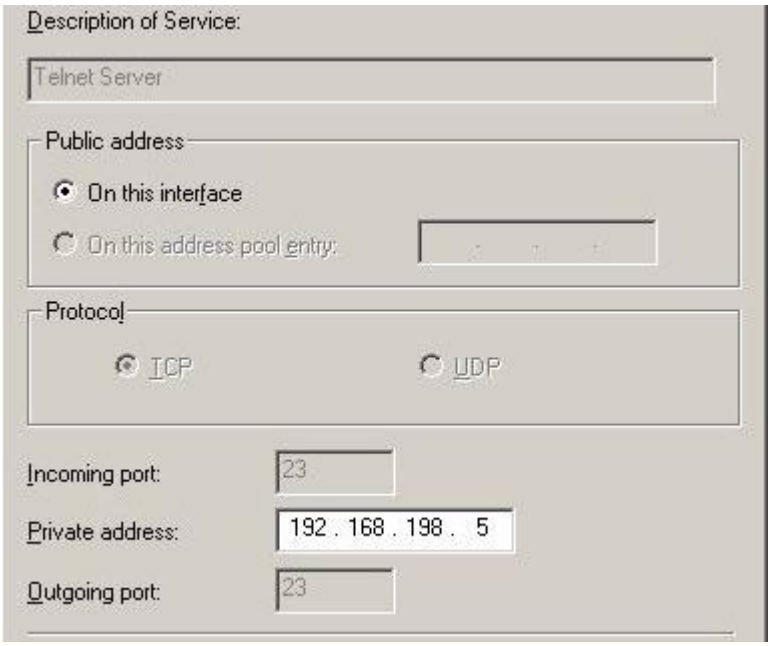


Рис. 10. Конфигурирование NAPT

Обратите внимание, что для прозрачной работы с NAT-прокси необходимо настроить маршрутизацию по умолчанию на сервер с настроенным NAT.

Контрольные вопросы

1. Что такое прокси-сервер?
2. Перечислите цели и задачи применения прокси серверов.
3. В чем принципиальное отличие специализированного прокси-сервера от неспециализированного?
4. Что такое NAT-маршрутизатор (прокси)?
5. Объясните основной принцип работы NAT-маршрутизатора.
6. Каковы преимущества и недостатки использования NAT-маршрутизатора в сравнении со специализированными прокси-серверами?
7. Для чего применяется механизм NAPT? В чем его отличие от NAT?

Задание на лабораторную работу

1. Настроить NAT-маршрутизатор на базе Windows Server.
2. Настроить NAPT для доступа с HTTP web-серверу в локальной сети.
3. Протестировать работу маршрутизатора с использованием любого браузера.
4. С использование снифферов, установленных как на сервере, так и на клиентских машинах, зафиксировать трафик при работе через NAT. Сопоставить полученные результаты с приведенным описанием принципа работы NAT-маршрутизатора.
5. Сделать выводы. Подготовить отчет.

Список рекомендованной литературы

1. Что такое прокси-сервер, и зачем он нужен. http://www.eserv.ru/WhatIsProxyServer
2. The IP Network Address Translator (NAT). RFC 1631, RFC 2663, RFC 3022
3. Cisco System Inc., «How NAT Works», http://www.cisco.com/warp/public/556/nat-cisco.shtml
4. Cisco System Inc., «The Trouble with NAT», http://www.cisco.com/web/about/ac123/ac147/ac174/ac182/about_cisco_ipj_archive_article09186a00800c83ec.html

Практическое занятие 4 Изучение диагностических утилит для администрирования

Ознакомиться с основными сетевыми сервисами и связанными с ними портами. Научиться использовать команду netstat для контроля за состоянием локальных портов.

Взаимодействие компьютеров между собой, а также с другим активным сетевым оборудованием, в TCP/IP-сетях организовано на основе использования сетевых служб, которые обеспечиваются специальными процессами сетевой операционной системы (ОС) — демонами в UNIX-подобных ОС, службами в ОС семейства Windows и т. п. Примерами сетевых сервисов являются веб-серверы (в т.ч. сайты всемирной паутины), электронная почта, FTP-серверы для обмена файлами, приложения IP-телефонии и многое другое.

Специальные процессы операционной системы (демоны, службы) создают «слушающий» сокет и «привязывают» его к определённому порту (пассивное открытие соединения), обеспечивая тем самым возможность другим компьютерам обратиться к данной службе. Клиентская программа или процесс создаёт запрос на открытие сокета с указанием IP-адреса и порта сервера, в результате чего устанавливается соединение, позволяющее взаимодействовать двум компьютерам с использованием соответствующего сетевого протокола прикладного уровня.

Порт (англ. port) — натуральное число, записываемое в заголовках протоколов транспортного уровня модели OSI (TCP, UDP, SCTP, DCCP). Используется для определения процесса-получателя пакета в пределах одного хоста.

Номер порта для «привязки» службы выбирается в зависимости от его функционального назначения. За присвоение номеров портов определённым сетевым службам отвечает IANA. IANA (от англ. Internet Assigned Numbers Authority — «Администрация адресного пространства Интернет») — функция управления пространствами IP-адресов, доменов верхнего уровня, а также регистрирующая типы данных MIME и параметры прочих протоколов Интернета. Исполняется компанией Public Technical Identifiers, которая находится под контролем ICANN. IANA отвечает за распределение всех зарезервированных имён и номеров, которые используются в протоколах, определённых в RFC. 1 октября 2016 года официально истёк срок действия договора о выполнении функций администрации адресного пространства интернета (IANA) между ICANN и Национальным управлением по телекоммуникациям и информации (NTIA) Министерства торговли США. При этом координирующая роль в исполнении функций IANA перешла в руки международного интернет-сообщества в связи с завершением срока действия договора с правительством США.

Номера портов находятся в диапазоне 0—65535 и разделены на 3 категории:

Номера портов	**Категория**	**Описание**
0—1023	Общеизвестные порты	Номера портов назначены IANA и на большинстве систем могут быть использованы исключительно процессами системы (или пользователя root) или прикладными программами, запущенными привилегированными пользователями. **Не должны использоваться** без регистрации IANA. Процедура регистрации определена в разделе 19.9 RFC 4340 (англ.).

1024—49151	Зарегистрированные порты	Номера портов включены в каталог IANA и на большинстве систем могут быть использованы процессами обычных пользователей или программами, запущенными обычными пользователями. **Не должны использоваться** без регистрации IANA. Процедура регистрации определена в разделе 19.9 RFC 4340.
49152—65535	Динамические порты	Предназначены для временного использования (например, для тестирования приложений до регистрации IANA), а также в качестве клиентских (используемых для частных служб внутри закрытых сетей). Эти порты **не могут быть зарегистрированы**.

Локальная копия списка входит в установочный пакет сетевых операционных систем. Файл локальной копии списка обычно называется services и в различных операционных системах «лежит» в разных местах:

Windows 98/ME
C:\Windows\services

Windows NT/XP/7
C:\Windows\system32\drivers\etc\services

UNIX-подобные ОС
/etc/services

В большинстве операционных систем можно посмотреть состояние сетевых служб при помощи команды (утилиты)

netstat -an

В ОС семейства Windows результат работы этой команды выглядит примерно так:

```
Активные подключения
 Имя  Локальный адрес    Внешний адрес     Состояние
 TCP  0.0.0.0:135      0.0.0.0:0       LISTENING
 TCP  0.0.0.0:445      0.0.0.0:0       LISTENING
 TCP  127.0.0.1:1026     0.0.0.0:0       LISTENING
 TCP  127.0.0.1:12025    0.0.0.0:0       LISTENING
 TCP  127.0.0.1:12080    0.0.0.0:0       LISTENING
 TCP  127.0.0.1:12110    0.0.0.0:0       LISTENING
 TCP  127.0.0.1:12119    0.0.0.0:0       LISTENING
 TCP  127.0.0.1:12143    0.0.0.0:0       LISTENING
 TCP  192.168.0.16:139    0.0.0.0:0       LISTENING
 TCP  192.168.0.16:1572   213.180.204.20:80   CLOSE_WAIT
 TCP  192.168.0.16:1573   213.180.204.35:80   ESTABLISHED
 UDP  0.0.0.0:445      *:*
 UDP  0.0.0.0:500      *:*
 UDP  0.0.0.0:1025      *:*
 UDP  0.0.0.0:1056      *:*
 UDP  0.0.0.0:1057      *:*
 UDP  0.0.0.0:1066      *:*
 UDP  0.0.0.0:4500      *:*
 UDP  127.0.0.1:123     *:*
 UDP  127.0.0.1:1900     *:*
 UDP  192.168.0.16:123    *:*
 UDP  192.168.0.16:137    *:*
 UDP  192.168.0.16:138    *:*
 UDP  192.168.0.16:1900   *:*
```

В UNIX-подобных ОС результат работы команды netstat -an имеет примерно такой вид:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address      Foreign Address     State
tcp    0   0 0.0.0.0:37       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:199       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:2601      0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:3306      0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:2604      0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:2605      0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:13       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:179       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:21       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:22       0.0.0.0:*        LISTEN
tcp    0   0 0.0.0.0:1723      0.0.0.0:*        LISTEN
tcp    0   0 10.0.0.254:1723     10.0.0.243:2441     ESTABLISHED
tcp    0   0 192.168.19.34:179    192.168.19.33:33793   ESTABLISHED
tcp    1   0 192.168.18.250:37    192.168.18.243:3723   CLOSE_WAIT
tcp    0   0 10.0.0.254:1723     10.0.0.218:1066     ESTABLISHED
tcp    1   0 192.168.18.250:37    192.168.18.243:2371   CLOSE_WAIT
tcp    0   0 10.0.0.254:1723     10.0.0.201:4346     ESTABLISHED
tcp    0   0 10.0.0.254:1723     10.0.0.30:2965     ESTABLISHED
tcp    0   48 192.168.19.34:22    192.168.18.18:43645   ESTABLISHED
tcp    0   0 10.0.0.254:38562    10.0.0.243:22      ESTABLISHED
tcp    0   0 10.50.1.254:1723    10.50.1.2:57355     ESTABLISHED
tcp    0   0 10.50.0.254:1723    10.50.0.174:1090    ESTABLISHED
tcp    0   0 192.168.10.254:1723   192.168.13.104:65535  ESTABLISHED
tcp    0   0 10.0.0.254:1723     10.0.0.144:65535    ESTABLISHED
tcp    0   0 10.0.0.254:1723     10.0.0.169:2607     ESTABLISHED
tcp    0   0 10.0.0.254:1723     10.0.0.205:1034     ESTABLISHED
udp    0   0 0.0.0.0:1812      0.0.0.0:*
udp    0   0 0.0.0.0:1813      0.0.0.0:*
udp    0   0 0.0.0.0:161       0.0.0.0:*
udp    0   0 0.0.0.0:323       0.0.0.0:*
udp    0   0 0.0.0.0:123       0.0.0.0:*
raw    0   0 192.168.10.254:47    192.168.13.104:*    1
raw    0   0 10.0.0.254:47      10.0.0.120:*      1
raw    0   0 10.10.204.20:47     10.10.16.110:*     1
raw    0   0 192.168.10.254:47    192.168.11.72:*     1
raw    0   0 10.0.0.254:47      10.0.0.144:*      1
raw    0   0 10.0.0.254:47      10.0.0.205:*      1
raw    0   0 10.50.0.254:47     10.50.0.174:*      1
raw    0   0 10.0.0.254:47      10.0.0.170:*      1
raw    0   0 10.0.0.254:47      10.0.0.179:*      1
```

Состояние (State) LISTEN (LISTENING) показывает пассивно открытые соединения («слушающие» сокеты). Именно они и предоставляют сетевые службы. ESTABLISHED — это установленные соединения, то есть сетевые службы в процессе их использования.

В случае обнаружения проблем с той или иной сетевой службой, для проверки её доступности используют различные средства диагностики, в зависимости от их наличия в данной ОС.

Одно из самых удобных средств — команда (утилита) tcptraceroute (разновидность traceroute), которая использует TCP-пакеты открытия соединения (SYN|ACK) с указанным сервисом (по умолчанию — web-сервер, порт 80) интересующего хоста и показывает информацию о времени прохождения данного вида TCP-пакетов через маршрутизаторы, а также информацию о доступности службы на интересующем хосте, либо, в случае проблем с доставкой пакетов — в каком месте пути они возникли.

В качестве альтернативы можно использовать отдельно traceroute для диагностики маршрута доставки пакетов (недостаток — использование UDP-пакетов для диагностики) и telnet или netcat на порт проблемной службы для проверки её отклика.

Traceroute — это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях TCP/IP. Traceroute может использовать

разные протоколы передачи данных в зависимости от операционной системы устройства. Такими протоколами могут быть UDP, TCP, ICMP или GRE. Компьютеры с установленной операционной системой Windows используют ICMP-протокол, при этом операционные системы Linux и маршрутизаторы Cisco — протокол UDP.

Traceroute входит в поставку большинства современных сетевых операционных систем. В системах Microsoft Windows эта программа носит название tracert, а в системах GNU/Linux, Cisco IOS и Mac OS — traceroute.

Рассмотрим пример работы программы в операционной системе Windows. Программа tracert выполняет отправку данных указанному узлу сети, при этом отображая сведения о всех промежуточных маршрутизаторах, через которые прошли данные на пути к целевому узлу. В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки. Необходимо отметить, что программа работает только в направлении от источника пакетов и является весьма грубым инструментом для выявления неполадок в сети. В силу особенностей работы протоколов маршрутизации в сети Интернет, обратные маршруты часто не совпадают с прямыми, причём это справедливо для всех промежуточных узлов в трейсе. Поэтому ICMP ответ от каждого промежуточного узла может идти своим собственным маршрутом, затеряться или прийти с большой задержкой, хотя в реальности с пакетами, которые адресованы конечному узлу, этого не происходит. Кроме того, на промежуточных маршрутизаторах часто стоит ограничение числа ответов ICMP в единицу времени, что приводит к появлению ложных потерь.

Для определения промежуточных маршрутизаторов traceroute отправляет целевому узлу серию ICMP-пакетов (по умолчанию 3 пакета), с каждым шагом увеличивая значение поля TTL («время жизни») на 1. Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем traceroute повторяет отправку серии пакетов, но уже с TTL, равным 2, что заставляет первый маршрутизатор уменьшить TTL пакетов на единицу и направить их ко второму маршрутизатору. Второй маршрутизатор, получив пакеты с TTL=1, так же возвращает «time exceeded in transit».

Процесс повторяется до тех пор, пока пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершённым.

На оконечном хосте IP-датаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа срок истёк, а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые traceroute датаграммы содержат UDP-пакет с заведомо неиспользуемым номером порта на адресуемом хосте. Номер порта будет равен 33434 + (максимальное количество транзитных участков до узла) — 1. В пункте назначения UDP-модуль, получая подобные датаграммы, возвращает ICMP-сообщения об ошибке «порт недоступен». Таким образом, чтобы узнать о завершении работы, программе traceroute достаточно обнаружить, что поступило ICMP-сообщение об ошибке этого типа.

Использование утилиты tracert

Запуск программы производится из командной строки. Для этого вы должны войти в неё. Для операционных систем семейства Windows существует несколько способов запуска командной строки:

Пуск — Выполнить — В графе «Открыть» написать «cmd» и нажать Ок.

Сочетание клавиш Win (кнопка с логотипом Windows) + R (должны быть нажаты одновременно) — В графе «Открыть» написать «cmd» и нажать Ок.

Пуск — Все программы (или просто «Программы», зависит от версии операционной системы) — Стандартные — Командная строка.

В открывшемся окне написать:

```
tracert example.net
```

Где *tracert* — обращение к программе, а example.net — любой домен или IPv4 адрес.

```
C:\Documents and Settings\Administrator>tracert ru.wikipedia.org

Трассировка маршрута к rr.esams.wikimedia.org [91.198.174.2]
с максимальным числом прыжков 30:

1    1 ms  <1 ms  <1 ms vpn4.kras.gldn [10.10.1.14]
2    2 ms  <1 ms  <1 ms C7604-BRAS4-FTTB.ranetka.ru [80.255.150.41]
3    1 ms   1 ms   4 ms C76-External.ranetka.ru [80.255.128.162]
4    1 ms  <1 ms  <1 ms pe-1.Krasnoyarsk.gldn.net [195.239.173.37]
5   79 ms  79 ms  98 ms cat01.Stockholm.gldn.net [194.186.157.62]
6  131 ms  131 ms  132 ms ams-ix.2ge-2-1.br1-knams.wikimedia.org [195.69.145.176]
7  131 ms  131 ms  131 ms te-8-2.csw1-esams.wikimedia.org [91.198.174.254]
8  133 ms  134 ms  133 ms rr.esams.wikimedia.org [91.198.174.2]

Трассировка завершена.
```

В UNIX/Linux системах существуют режимы, в которых запуск программы возможен только от имени суперпользователя root (администратора). К числу этих режимов относится важный режим трассировки с помощью ICMP (ключ -I).

Во всех остальных случаях, traceroute может работать от имени обычного рядового пользователя. При этом, параметры по умолчанию различаются от дистрибутива к дистрибутиву, хотя в справке традиционно пишется ключ -U (UDP) в качестве такового. В отдельных RedHat-based дистрибутивах фактически в качестве умолчания используется -I, поэтому в случае, если команда из следующего примера выдаст сообщение о недостатке прав, попробуйте явно указать ключ -U.

```
[user@localhost ~]$ traceroute www.ru
traceroute to www.ru (194.87.0.50), 30 hops max, 38 byte packets
 1  mygateway.ar7            (192.168.1.1)    0.777 ms  0.664 ms  0.506 ms
 2  L0.ghsdr04             (213.227.224.91)  15.661 ms  15.867 ms 31.426 ms
 3  213.227.224.1            (213.227.224.1)  16.797 ms  18.221 ms 16.756 ms
 4  dg                  (213.186.216.161) 53.068 ms  39.163 ms 38.283 ms
 5  br13                 (213.186.193.43)  40.156 ms  39.768 ms 42.803 ms
 6  aggr                 (62.221.40.169)  37.884 ms  38.712 ms 37.207 ms
 7  edge-3GE-216dot1q.kiev.ucomline.net (213.130.30.182)  39.723 ms  38.039 ms 41.261 ms
 8  ae0-202.RT771-001.kiv.retn.net    (81.222.15.1)    40.029 ms  37.088 ms 40.039 ms
 9  ae0-3.RT502-001.msk.retn.net    (81.222.15.1)    128.932 ms 122.043 ms 121.612 ms
10  GW-Demos.retn.net          (81.222.8.46)    120.023 ms 121.135 ms 119.493 ms
11  iki-1-vl10.demos.net        (194.87.0.83)    119.074 ms 119.784 ms 123.607 ms
12  www.ru               (194.87.0.50)    120.358 ms 122.545 ms 119.399 ms
```

Часто встречается заблуждение, что traceroute, как и ping, работает только по протоколу ICMP. В связи с этим начинающие администраторы, разрешив в файерволе протокол ICMP, получают рабочий ping и нерабочий traceroute. Для исправления такой ситуации необходимо дополнительно разрешить в файерволе UDP-пакеты на порты выше

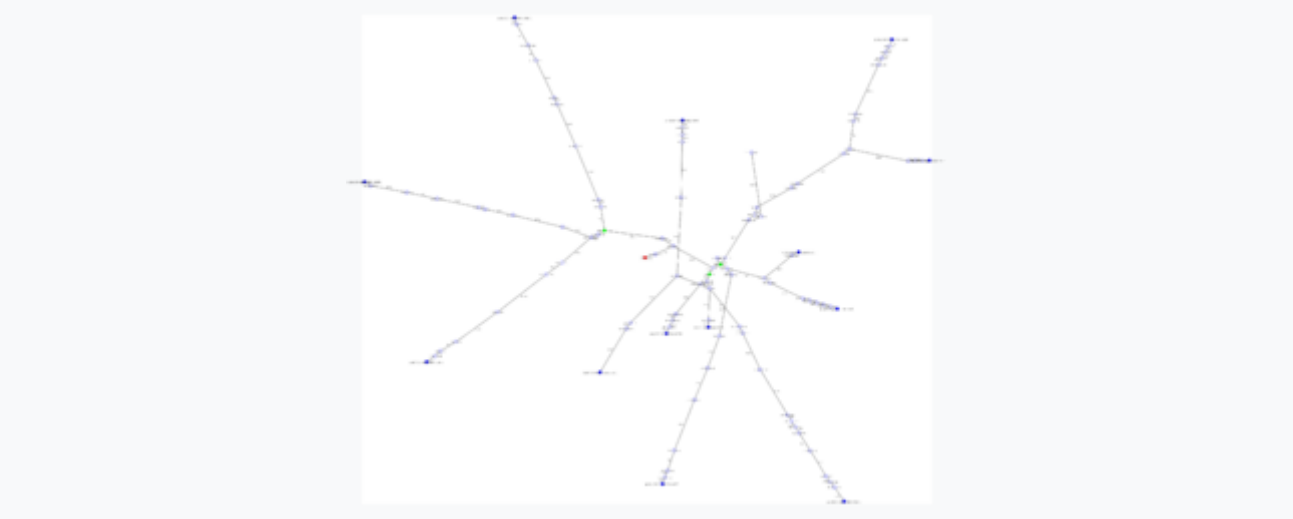
33434 (в некоторых источниках указано, что достаточно указать диапазон портов от 33434 до 33534).

tracepath — похожая на traceroute программа, но может строить асимметричные трассы и имеет некоторые другие отличия.

mtr — Интерактивная программа, способная постоянно выводить обновлённую статистику по трассе.

tracemap — программа, позволяющая выполнять трассировку пути на несколько хостов сразу и представить полученные данные в виде графической карты.

tcptraceroute (tracetcp) — аналогичная traceroute программа, но предназначена для диагностики TCP соединений; вместо UDP-пакетов использует TCP-пакеты открытия соединения (SYN|ACK) с указанным сервисом (по умолчанию — web-сервер, порт 80) интересующего хоста; в результате получаем информацию о времени прохождения данного вида TCP-пакетов через маршрутизаторы и информацию о доступности сервиса на интересующем хосте, либо, в случае проблем с доставкой пакетов — видим, в каком месте трассы они возникли.



Полученная с помощью tracemap графическая карта трассировки пути на корневые серверы DNS с хоста

ЗАДАНИЕ

1. Найдите локальную версию списка распределения портов на своём компьютере.
2. При помощи утилиты netstat проверьте состояние портов на своём компьютере.
3. При помощи утилиты *tracert* проведите трассировку узлов google.com, yandex.ru, rnd.mts.ru и ещё двум узлам на своё усмотрение. Дайте интерпретацию полученным результатам.
4. Ответьте на 2 вопроса из приеденного ниже списка. Номера вопросов определите остаток от деления вашего номера по журналу на 11 и следующий (предыдущий, если следующего нет) вопрос
5. Оформите отчёт по лабораторной работе с включением в него скриншотов выполняемых операций, пояснений сделанного и ответов на вопросы.

Контрольные вопросы

1.Какую роль выполняет ОС при организации сетевой работы?
2.Что понимают под термином «сетевая оболочка»?
3.Что означает термин «сетевой сервис»?
4.Может ли использоваться сетевая оболочка в сетевых операционных системах

5. В чем сходство и различие сетевой и распределенной ОС?
6. Какие два признака характеризуют термин «сетевая ОС»?
7. Какие основные функциональные компоненты можно выделить в сетевой ОС?
8. Дайте определение сетевой службы.
9. Является ли сетевая служба только клиентской или только серверной?
10. Назовите принципиальное различие между клиентом и сервером.
11. Перечислите основные требования, предъявляемые к современным операционным системам

Лабораторная работа 4 Установка, настройка и конфигурирование Web-сервера IIS

Теоретическая часть

Понятие серверного приложения

В практикуме рассматривается служба Интернета WWW (World Wide Web — Всемирная паутина), которая управляет передачей гипертекстовых страниц и регламентируется протоколом HTTP (Hyper Text Transfer Protocol). На компьютере-сервере, поддерживающем эту службу, должна быть установлена программа, которая также называется Web-сервером (кратко — сервером). В специальной папке компьютера-сервера хранятся Web-страницы, которые могут быть запрошены с компьютера-клиента общеизвестными способами (например, набором имени страницы в адресной строке браузера или активизацией гиперссылки).

Web-страницы могут быть двух видов: страницы, написанные на языке HTML (с расширением .htm или .html) и серверные приложения. HTML-страницы отсылаются сервером клиенту без пред варительной обработки. Серверные приложения создаются с помощью специальных технологий; расширение файла приложения показывает, какая именно технология использовалась: .asp — ASP-технологии, .aspx — технологии ASP.NET, .php — PHP-технологии, и т.п. При запросе клиентом серверное приложение преобразуется Web-сервером в HTML-файл, и этот файл отсылается клиенту. Интерпретация HTML-файлов (независимо от того, был он получен из серверного приложения или нет) осуществляется на компьютере-клиенте программой-браузером.

Серверное приложение может быть программой на алгоритмическом языке, или текстом на языке HTML, в который включены фрагменты (называемые сценариями или скриптами) на алгоритмическом языке. Обработка серверного приложения Web-сервером представляет собой трансляцию серверного приложения в HTML-текст. В процессе трансляции могут быть использованы данные из запроса клиента; например, сведения компьютере или браузере клиента, а также данные, посылаемые клиентом в соответствии с решаемой задачей. Эти данные могут существенно повлиять на вид ответной Web-страницы. Таким образом, серверные приложения представляют собой динамические, интерактивные Web-страницы, формируемые на сервере.

Трансляцию серверных приложений осуществляет специальный программный модуль, входящий в состав Web-сервера или подключенный к нему. Такой модуль, включенный в IIS-сервер и осуществляющий трансляцию asp-приложений, называется Script host.

Заметим, что существуют технологии формирования интерактивных Web-страниц на компьютере-клиенте с помощью браузера. Настоящий практикум посвящен именно серверным приложениям.

Передача данных задачи пользователя на сервер

Основным способом передачи данных от клиента к серверу является использованием HTML-форм. Формы содержат интерфейсные элементы (элементы управления). Примеров таких элементов могут быть текстовые окна для ввода данных, списки (селекторы) для выбора значений, флажки, радиокнопки. Предполагается, что читатель знаком с кодированием этих элементов на языке HTML. С каждым элементом формы связаны имя (атрибут элемента NAME) и значение (как правило, атрибут VALUE). Для передачи данных на сервер форма обязательно должна содержать элемент управления submit. Этот элемент представляет собой кнопку, при нажатии которой данные формы автоматически включаются в запрос и запрос отправляется на сервер.

При использовании формы для отправки данных на сервер тэг <FORM> обязательно

должен содержать два атрибута — ACTION и METHOD. В атрибуте ACTION записывается URL серверного приложения, формирующего ответную Web-страницу. При нажатии кнопки submit указанный URL включается в стартовую строку отправляемого запроса. Если серверное приложение находится в том же виртуальном каталоге сервера, из которого была вызвана Web-страница, содержащая форму, или в подчиненных ему папках, то вместо полного URL можно указать путь к серверному приложению относительно виртуального каталога.

Атрибут METHOD определяет метод передачи данных от клиента к серверу. В нашем практикуме мы будем рассматривать два метода — GET и POST. Метод GET обеспечивает присоединение данных формы к URL серверного приложения через знак вопроса (?), и расширенный таким образом URL, как уже было сказано выше, при нажатии кнопки submit включается в стартовую строку запроса. Метод POST означает, что данные формы включаются в тело запроса. Для обоих методов включаемые в запрос данные имеют вид: имя элемента формы = значение элемента. Такие пары вида имя = значение отделяются друг от друга символом «&». Отметим, что все символы, входящие в имя и значение, кроме латинских букв и пробела, при включении в запрос автоматически заменяются своим шестнадцатеричным кодом, перед которым ставится символ «%». Латинские буквы не подлежат перекодировке, а пробел заменяется символом «+».

Сравним два рассмотренных метода передачи данных. Метод POST меньше, чем GET, ограничивает объем передаваемых данных и предпочтительней с точки зрения безопасности (так как данные нельзя прочитать в адресной строке браузера). Однако, используя GET, можно не только передавать данные полей формы, но и «вручную» присоединить данные к URL (после символа «?»), например, при запросе страницы из адресной строки браузера или в гиперссылке.

Структура простейшего asp-приложения

Файл asp-приложения обязательно имеет расширение .asp. Он содержит текст на языке HTML, в который вставлены сценарии на алгоритмическом языке. Сценарий ограничен парами символов «<%» и «%>», первая пара играет роль открывающей, а вторая — закрывающей скобки. Существуют другие способы ограничения сценариев, они будут рассмотрены позже. Алгоритмический язык сценариев указывается в инструкции <%@ Language = язык%>. Эта инструкция располагается в первой строке файла и относится ко всем включенным в него сценариям. Стандартным языком сценариев является VBScript; при его использовании указанная инструкция может быть упущена. Код, полученный в результате интерпретации сценария, вставляется на место сценария в HTML-файле. Конструкции языка VBScript, используемые в данном пособии, приведены в Приложении 1.

ASP-технология предоставляет широкий спектр возможностей для извлечения данных из запроса, поступившего на сервер, и формирования ответной Web-страницы. Основные возможности будут рассмотрены в настоящем пособии. При создании ответной Web-страницы очень часто используется оператор Response.Write, который выводит строку символов в формируемый HTML-текст. Отметим, что Response — это объект, содержащий основные средства формирования динамической Web-страницы, а Write — метод этого объекта. Приведенное ниже простейшее приложение выводит фразу «HELLO!» в окно браузера:

```
<%@ Language = VBScript%>.
<HTML>
<HEAD> <TITLE> Первый пример</TITLE> </HEAD>
<BODY>
  <% Response.Write “HELLO!” %>
</BODY>
</HTML>
```

Если сценарий состоит из вывода одной строки, то его можно сделать еще короче, заменив оператор Response.Write символом «=». Так, сценарий в приведенном выше примере можно заменить следующим: <% = “HELLO!” %>.

Рассмотрим приложение, которое выводит в окно браузера время формирования ответной Web-страницы (Time — встроенная функция VBScript):

```
<HTML>
<HEAD> <TITLE> Узнай время</TITLE> </HEAD>
<BODY>
  Точное время на стороне сервера:<%=Time %>
</BODY>
</HTML>
```

Обратите внимание, что для вызова серверного приложения надо обязательно сформировать запрос от клиента к серверу, даже если вы работаете в отладочном режиме «обратной петли», и Ваш компьютер является одновременно и сервером, и клиентом. Например, можно набрать адрес серверного приложения в окне браузера. Серверные приложения нельзя вызывать, как обычные приложения, двойным кликом на пиктограмме в окне папок Проводник или Мой компьютер.

Рассмотрим простые примеры asp-приложений, обрабатывающих данные HTML-форм. Ниже приведен файл (назовем его concat.htm), который формирует на стороне клиента простейший запрос, обеспечивающий передачу на сервер значений двух строк из полей редактирования формы в окне браузера:

```
<HTML>
<HEAD><TITLE> Ввод значений a и b для передачи на сервер</TITLE>
</HEAD>
<BODY>
<FORM ACTION="concat.asp" METHOD=POST NAME="forma">
Первое значение <input type="text" name="a" value=""> <br>
Второе значение <input type="text" name="b" value=""> <br>
<input type="submit" name="plus" value="результат">
</FORM>
</BODY>
</HTML>
```

Обратите внимание, что этот файл не является asp-приложением, о чем говорит и его расширение. В теге <FORM> указано имя серверного приложения (concat.asp) для обработки этих строк. Это приложение осуществляет конкатенацию (сцепление) полученных строк и формирует ответную Web-страницу, содержащую поля редактирования с исходными значениями строк и результатом сцепления. Содержание файла concat.asp:

```
<HTML>
<HEAD> <TITLE> Результат конкатенации<TITLE > </HEAD >
<BODY>
<%
    a=Request("a") ‘в переменную a считывается строка из элемента a
    b=Request("b") ‘в переменную b считывается строка из элемента b
    c=a+b ‘ c принимает значение результата сцепления a и b
‘ ниже в поля редактирования выводятся значения a, b, c
%>
<FORM>
Первое слагаемое <input type="text" value=<% Response.Write a %> > <br>
Второе слагаемое <input type="text" value=<% Response.Write b %> > <br>
Сумма <input type="text" value=<% Response.Write c %> > <br>
</FORM>
```

</BODY>
</HTML>

Заметим, что оператор имя=Request("имя") извлекает значение данного из запроса. Имя в правой части оператора (в данной ситуации кавычки обязательны!) — это имя элементы формы, значение которого передано в запросе; имя в левой части — это имя ячейки оперативной памяти сервера. Естественно, имена в левой и правой частях могут не совпадать. При выборе одинаковых имен легче читается программный код.

Операция «+» в языке VBScript (как, например, и в языке Pascal) выполняется в зависимости от контекста: над строками как конкатенация, над числами как сложение. Если в файле concat.asp оператор c=a+b заменить оператором c=CDbl(a)+ CDbl(b), то с будет не результатом сцепления строк, а суммой чисел a и b, так как CDbl — это функция преобразования данного в вещественное число. Если, кроме того, в сценарии <% Response.Write c %> с изменить на CStr(c), то серверное приложение будет выводить сумму двух чисел (CStr — функция преобразования в строку).

4. 1.4. Установка IIS-сервера

IIS-сервер – серверный программный комплекс, входящий в состав операционной системы Windows (начиная с Windows 2000). В этот комплекс входит Web-сервер и ASP-технология подготовки серверных приложений.

Для установки IIS-сервера надо выполнить следующие шаги:
Вызовите окно Мастера компонентов Windows (*Пуск/ Настройка/Панель управления /Установка и удаление программ/ Добавление и удаление компонентов Windows).* Окно мастера приведено на рис.1. В списке компонентов выделите *Internet Information Services (IIS)*

Контрольные вопросы

1. Что понимают под программным обеспечением сетей ЭВМ?
2. Что дает предприятию использование компьютерных сетей?
3. Классификация сетевого программного обеспечения.
4. Что называют операционной системой?
5. Что входит в группу прикладного программного обеспечения?
6. По каким критериям можно охарактеризовать сетевую операционную систему?
7. Что называют сетевым драйвером?
8. Что называют сетевым протоколом?
9. Перечислить сетевые операционные системы.
10. Что такое сетевые службы?
11. Что называют стандартным программным обеспечением ЭВМ?
12. Что такое технология «клиент-сервер»?

Практическое занятие 5 Установка, настройка и конфигурирование Web-сервера Apache

Цель работы: настройка и администрирование web-сервера ***Apache*** под ОС Linux.

Практическое задание выполняется в локальной сети на рабочей станции с операционной системой Linux 7 или более поздней. В лабораториях кафедры операционная система Linux работает на компьютерах под управлением программного пакета VMware. Этот пакет позволяет создавать так называемые «виртуальные машины» – мнимые компьютеры, не зависящие от выполняющейся в текущее время на данном компьютере операционной системы (ОС). Для запуска ОС Linux необходимо запустить VMware на рабочей станции, выбрать из списка требуемую операционную систему и нажать кнопку «Power ON».
После окончания загрузки для входа в систему необходимо использовать имя пользователя ***root****, пароль –* ***rootuser****.*

Порядок выполнения лабораторной работы

Подготовка и допуск к работе. К выполнению лабораторной работы допускаются студенты, которые подготовились к работе и имеют не более двух невыполненных предыдущих работ.
Перед работой студент должен:

- предъявить преподавателю полностью оформленный отчет о предыдущей работе;
- ответить на вопросы преподавателя.

К работе не допускаются студенты, которые не выполнили одно из вышеперечисленных требований.
Отчёт по работе должен содержать**:**

- текст задания;
- перечень всех использованных в лабораторной работе команд и инструкций;
- содержимое конфигурационных файлов (без комментариев!!!);
- отрывок из log файлов, демонстрирующий обращения в web-серверу;
- вывод по работе.

Администрирование и установка WEB-СЕРВЕРа

Самый распространенный web-сервер в мире – это **Apache**. По данным компании Netcraft (*http://www.netcraft.com/Survey/*) общее число web-узлов, работающих под его управлением, к концу 1998 г. достигало 2 млн (55 % общего числа узлов) и это число постоянно растет. Для сравнения – на долю серверов Microsoft приходится 25 %, Netscape – 7 %. Будучи бесплатной и открытой программой, предназначенной для бесплатных же Unix-систем (FreeBSD, Linux и др.), **Apache** по функциональным возможностям и надежности не уступает коммерческим серверам, а широкие возможности конфигурирования позволяют настроить его для работы практически с любой конкретной системой. Существуют локализации сервера для различных языков, в том числе и для русского.

Список директив (настроек), используемых для конфигурации

web-сервера

Каждая строка конфигурационного файла, кроме строк, начинающихся с #, описывает какую-то директиву конфигурации и ее значение. Строки, начинающиеся с # – это комментарии.
Более подробное описание всех директив находится в файле l2.html методического пакета к лабораторной работе, находящегося в методическом разделе сервера кафедры «Автоматика и системотехника». Описание директив, необходимых для выполнения лабораторной работы, приведено в табл 1.

Таблица 1
Описание основных директив конфигурирования web-сервера

Наименование директивы	**Описание**
ServerType standalone	Показывает тип запуска программы *httpd* – она может вызываться через *inetd (inetd)* или как отдельный демон (*standalone*). По умолчанию вызывается как отдельный демон
Port 80	Порт, по которому http-сервер будет получать запросы. По умолчанию 80
HostnameLookups off	Определяет, будут ли записываться в лог-файлы имена (*on*) или только *ip*-адреса (*off*) хостов, просматривающих сайт. По умолчанию *off*
User nobody **Group nobody**	Определяет то, под каким пользователем будет запускаться *httpd*. Для доступа к 80 порту *httpd* должен запускаться под *root*'ом. Если в директивах **User** и **Group** указаны другие пользователи, то *httpd* все свои действия выполняет с правами этих пользователей, а не *root*'а
ServerAdmin root@localhost	Здесь указывается *e-mail* администратора, отвечающего за работу сервера
ServerRoot /etc/httpd	Каталог, в котором находятся все конфигурационные файлы, лог-файлы и модули. Все пути, которые используются далее, указаны относительно этого каталога
ErrorLog logs/error_log	Местонахождение файла логов об ошибках сервера
LogLevel warn	Уровень сообщений в файлах логов. Возможные значения: *debug, info, notice, warn, error, crit, alert, emerg*
LoadModule имя_модуля modules/имя_файла_модуля.so	Команда используется для загрузки различных модулей сервера **Apache**
ClearModuleList	Очистка списка модулей
AddModule имя_файла_модуля.c	Добавления модулей в список модулей

Продолжение табл. 1

Наименование директивы	Описание
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined **LogFormat "%h %l %u %t \"%r\" %>s %b" common**	Описание различных форматов записи логов. Имя формата используется в директиве *CustomLog*

LogFormat "%{Referer}i -> %U" referer **LogFormat "%{User-agent}i" agent** **LogFormat "формат" имя**	
CustomLog logs/access_log common	Задает формат файла логов запросов. С помощью *CustomLog* можно задать файл логов любого формата и имени
PidFile /var/run/httpd.pid	Имя файла, в котором хранится *PID* запущенного *httpd*. Используется для останова сервера и перезапуска
UseCanonicalName on	Разрешает использование коротких имен в *URL*'ах. Удобно при использовании в Intranet-серверах. Если включено (*on*), преобразует, например строку *http://www/stat* в строку *http://www.твой-домен.ru/stat*, если отключено (*off*), преобразования не происходит
Timeout 300	Время в секундах, по истечении которого при отсутствии запросов сервер прерывает связь с клиентом
KeepAlive On	Разрешать (*on*) или нет(*off*) множественные запросы через одно TCP соединение
MaxKeepAliveRequests 100	Максимальное количество запросов через одно TCP соединение
KeepAliveTimeout 15	Время ожидания следующего запроса в секундах

Продолжение табл. 1

Наименование директивы	Описание
MinSpareServers 8 **MaxSpareServers 20**	Минимальное и максимальное значение загруженных процессов *httpd*. Если набрать в консоли **ps ax | grep httpd**, то можно просмотреть все процессы *httpd*, загруженные на машине. Если используется **Apache** на локальной машине, то необходимо уменьшить значение *MinSpareServers*
StartServers 10	Количество изначально запускаемых процессов
MaxClients 150	Ограничивает количество одновременно подключенных клиентов к серверу
MaxRequestsPerChild 100	Максимальное количество запросов, которые обрабатывает процесс до завершения работы
DocumentRoot /home/httpd/html	Путь к файлам сайта. Здесь хранятся те файлы, которые пользователь получает при наборе *URL*'а *http://твой.сайт.ru.* После установки там находится документация по **Apache**. Эти файлы следует заменить на файлы своего сайта
UserDir public_html	Имя каталога в домашней директории пользователя, где должны находиться файлы его веб-страницы. Обращение к ним происходит по *URL*'у *http://твой.сайт.ru/~имя_пользователя* (обрати

	внимание на ~ ("тильда"))
DirectoryIndex index.html index.shtml index.cgi	Файлы индекса каталога. Это те файлы, которые выводятся при задании в броузере *URL*'а, являющемся ссылкой на каталог. Например, *http://localhost/manual* – выведет файл *index.html*, находящийся в каталоге */home/httpd/html/manual*. Если файл индекса отсутствует в каталоге, выводится все содержимое каталога. Можно задать *DirectoryIndex default.htm*

Продолжение табл. 1

Наименование директивы	Описание
FancyIndexing on	При включении этой опции список файлов каталога будет выводиться в виде таблицы, в которой над каждой колонкой есть ссылка, при нажатии на которую список сортируется по этой колонке (например, по времени модификации или размеру)
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip **AddIconByType (TXT,/icons/text.gif) text/*** **. . .** **AddIcon /icons/binary.gif .bin .exe** **. . .** **DefaultIcon /icons/unknown.gif**	Эти директивы задают имя файла иконки в зависимости от типа, расширения и имени файла, которая будет выводиться перед именем файла в списке содержимого каталога
ReadmeName README **HeaderName HEADER**	*ReadmeName* – имя README файла, который сервер ищет по умолчанию. *HeaderName* – имя файла, который включается в листинг каталога в качестве заголовка.
IndexIgnore .??* *~ *# HEADER* README* RCS	Файлы, не включаемые в список каталога
AccessFileName .htaccess	Имя файла с правилами доступа, содержимое которого проверяется для каждого каталога
TypesConfig /etc/mime.types	Файл с описаниями типов файлов и принадлежащих к ним расширений
DefaultType text/plain	Тип файла по умолчанию для тех файлов, тип которых неизвестен
AddEncoding x-compress Z **AddEncoding x-gzip gz**	Позволяет некоторым броузерам распаковывать “сжатые” и gzip-овые архивы "на лету", т. е. при копировании файла типа *textcounter.tar.gz* получается уже распакованный файл *textcounter.tar*

Продолжение табл. 1

Наименование директивы	Описание
AddLanguage en .en	Позволяет определить язык документа. Если на

AddLanguage fr .fr **AddLanguage de .de** **AddLanguage da .da** **AddLanguage el .el** **AddLanguage it .it**	сайте есть английская и русская версии страницы, то можно создать файлы *doc.html.ru* и *doc.html.en* и в зависимости от языка, прописанного в броузере клиента, ему будет показываться одна из страниц при обращении к *doc.html*. Расширение не обязательно должно совпадать с аббревиатурой языка
LanguagePriority en fr de	Определяет приоритет языка документа
Alias /icons/ /home/httpd/icons/	Разрешает создавать *алиасы* (альтернативные имена) для каталогов, т.е. если html-страница лежит в */home/httpd/html*, то при обращении к *http://твой.сайт.ru/icons* **Apache** будет искать каталог */home/httpd/html/icons*, а после определения такого *алиаса* **Apache** будет искать каталог */home/httpd/icons*
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/	То же самое, что и *Alias*, но для серверных скриптов (**CGI**)
AddType text/html .shtml **AddHandler server-parsed .shtml**	Добавляет поддержку **SSI** для файлов с расширением **.shtml**
AddHandler imap-file map	Добавляет поддержку *imagemap*'ов
BrowserMatch "Mozilla/2" nokeepalive **BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0** **BrowserMatch "RealPlayer 4\.0" force-response-1.0** **BrowserMatch "Java/1\.0" force-response-1.0** **BrowserMatch "JDK/1\.0" force-response-1.0**	Прописывает различные параметры работы *httpd* при обращении к нему описанных броузеров

Продолжение табл. 1

Наименование директивы	Описание
<Directory /> **Options None** **AllowOverride None** **</Directory>**	Описывает параметры доступа к корневой директории. **Options** контролирует то, какие опции доступа к директории доступны. Как видно из примера – никакие (**None**). Следовательно, доступа никакого. **AllowOverride None** – запрещает изменять опции доступа посредством содержимого файла *.htaccess* каталога
<Directory /home/httpd/html> ***Options Indexes Includes FollowSymLinks*** ***AllowOverride None*** ***order allow,deny*** ***allow from all*** **</Directory>**	Устанавливает параметры доступа к каталогу */home/httpd/html*. Опции доступа (**Options**) следующие:
Indexes	если эта опция включена, то при указании *URL*'а, ссылающегося на каталог (типа *http://www.ru/nirvana* или *http://localhost/manual*), при отсутствии в каталоге файла, описанного директивой

	DirectoryIndex (*index.html,index.php3,index.cgi* и т. д.), *httpd* возвращает клиенту листинг этого каталога
Includes	разрешает выполнение **SSI** директив в файлах, описанных директивой **AddHandler server-parsed *mun*** и находящихся в каталоге */home/httpd/html*
FollowSymLinks	позволяет использовать символические ссылки на файлы или каталоги, не находящиеся в */home/httpd/html*
AllowOverride None	см. выше

Окончание табл. 1

Наименование директивы	Описание
Order allow, deny	определяет последовательность применения правил запрета и правил разрешения доступа к каталогу. Здесь сначала проверяются разрешающие правила, затем запрещающие, т. е. если разрешено, например, по *IP*, и запрещено, например, по имени *хоста*, то будет запрещен доступ к этому каталогу, т. к. запрещающие правила сработали последними
Allow from all	разрешен доступ отовсюду.
<Directory /home/httpd/cgi-bin> ***AllowOverride None*** ***Options ExecCGI*** **</Directory>**	Прописывает параметры доступа к каталогу */home/httpd/cgi-bin* (к тому, где лежат *CGI*-скрипты). **AllowOverride None** – см. выше. **Options ExecCGI** – разрешен запуск *CGI* и больше ничего

Alias /doc /usr/doc **<Directory /usr/doc>** **order deny,allow** **deny from all** **allow from localhost** **Options Indexes FollowSymLinks** **</Directory>**	**Создает *алиас /doc* для каталога */usr/doc* и описывает параметры доступа к нему. order deny,allow – сначала срабатывают запрещающие правила, затем разрешающие. deny from all – запрещен доступ всем клиентам, откуда бы то ни было. allow from localhost – разрешен доступ с машины, на которой и запущен Apache. Options Indexes FollowSymlinks – см. выше. (Если запустить Netscape на той же машине, на которой загружен Apache, и набрать URL – *http://localhost/doc* или *http://127.0.0.1/doc*, то можно увидеть листинг каталога */usr/doc*.)**

Настройка виртуальных серверов в файле httpd.conf

В большинстве случаев один *http*-сервер способен обрабатывать запросы, поступающие на различные, так называемые виртуальные, Web-серверы. Виртуальные серверы могут иметь как один и тот же IP-адрес, но разные доменные имена, так и разные IP-адреса. С точки зрения пользователя второй вариант чуть более предпочтителен, поскольку запрос к серверу, отличающемуся от основного только доменным именем, должен содержать его имя, а некоторые старые браузеры, не поддерживающие протокол HTTP/1.1 (например, Microsoft Internet Explorer 2.0), не включают в запрос эту информацию. Однако такие браузеры выходят из употребления (сейчас их уже менее 0,5 % от общего числа); с другой стороны, выделение собственного
IP-адреса каждому виртуальному серверу может быть неоправданной растратой адресного пространства компании.
Для описания адресов и доменных имен виртуальных серверов служат директивы **ServerName, ServerAlias, NameVirtualHost и VirtualHost**. Они необходимы, только если вам нужно установить более одного виртуального сервера.
Директива **ServerName**, находящаяся вне секций **VirtualHost**, определяет имя основного сервера, т. е. сервера, корневой каталог которого задан директивой **DocumentRoot** в файле *srm.conf*. Виртуальные серверы наследуют настройки основного; при необходимости специальной настройки соответствующие директивы помещаются в секции **VirtualHost**, относящейся к данному серверу. Допустимы любые директивы, которые могут встретиться в файлах *httpd.conf* и *srm.conf*, например **DocumentRoot, ErrorLog, CustomLog, Location, ServerAdmin.**

Задание на лабораторную работу

Лабораторная работа выполняется по вариантам. Задания приведены в табл. 2.

Таблица 2

Задания на лабораторную работу

Вариант	**Задание**
Подгруппа 1, 5	Сконфигурировать сервер на порт 90; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 1, 3, 4, 5; указать, чтобы в файлах log использовались буквенные имена, а не цифровые; создать дополнительный виртуальный сервер на порт 100, для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 2, 3
Подгруппа 2, 6	Сконфигурировать сервер по умолчанию; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 4, 5; указать, чтобы в файлах log использовались цифровые имена, а не буквенные; создать дополнительный виртуальный сервер на домен (буквенное имя) ***lvirn***, где ***n*** – номер компьютера; для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 1, 3
Подгруппа 3, 7	Сконфигурировать сервер на порт 80; для тестового сайта использовать содержимое каталога testserver1; переименовать файл index.html в этом каталоге в sind.htm; указать в конфигурации, чтобы данный файл загружался по умолчанию; добиться, чтобы выполнялись тесты 4, 5; указать, чтобы в файлах log использовались цифровые имена, а не буквенные; создать дополнительный виртуальный сервер на порт 100; для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 1, 2

Окончание табл. 2

Вариант	Задание
Подгруппа 4, 8	**Сконфигурировать сервер по умолчанию; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 3, 5; указать, чтобы в файлах log использовались буквенные имена, а не цифровые; создать дополнительный виртуальный сервер на домен (буквенное имя) *lvirn,* где *n* – номер компьютера, для тестового сайта использовать содержимое каталога testserver2; переименовать файл index.html в этом каталоге в sind.htm; указать в конфигурации, чтобы данный файл загружался по умолчанию; добиться, чтобы выполнялись тесты 3, 4**

Для редактирования файлов конфигурации и навигации по файловой системе удобно использовать программу **mc**, для ее загрузки необходимо набрать в командной строке:

[root@lis] $ **mc**
Интерфейс программы интуитивно понятен и не представляет трудностей при использовании.
Файлы конфигурации web-сервера **Apache** в данной системе находятся в каталоге: **/etc/httpd/conf/**
Для запуска, остановки, перезапуска web-сервера используются следующие скрипы (программы):
Остановка: **/etc/rc.d/init.d/httpd stop**.
Запуск: **/etc/rc.d/init.d/httpd start**.
Перезапуск: **/etc/rc.d/init.d/httpd restart**.
В ходе лабораторной работы необходимо произвести настройку web-сервера в соответствии с заданием, продемонстрировать его работоспособность преподавателю.
Для демонстрации работоспособности необходимо переписать с *ftp://ais.khstu.ru/incoming* средствами Linux на локальный компьютер каталоги testserver1 и testserver2, для чего использовать в программе ***mc*** пункт меню ***left (right) –> ftp link***. В данных каталогах содержатся тестовые сайты:

- проверка исполнения файла */cgi-bin/test1.cgi*
- проверка исполнения файла */cgi-bin/test2.pl*
- проверка исполнения файла */cgi-bin/test3*
- проверка SSI-включения файла */cgi-bin/test1.cgi в файл test4.html*
- проверка SSI-включения файла */cgi-bin/test1.cgi в файл test4.shtml*

Для создания виртуальных серверов по доменным именам используемые доменные имена необходимо прописать в файле *c:\winnt\system32\driver\etc\hosts* на том компьютере, на котором будет проводиться проверка.
При защите необходимо знать назначение используемых директив, уметь объяснить информацию, содержащуюся в log файлах.

Контрольные вопросы

1. Какие файлы содержат конфигурационную информацию web-сервера?
2. Какова последовательность установки web-сервера?
3. Как проверить работоспособность web-сервера?
4. Где хранятся log файлы?
5. Что такое виртуальный web-сервер?

Библиографический список

1. *Бэндл Дэвид.* Защита и безопасность в сетях Linux = Linux security toolkit: Пер. с англ. / Бэндл Дэвид. - СПб.: Питер, 2002. – 480 с.
2. *Блэк У.* Интернет: Протоколы безопасности = Internet Security Protocols.Protecting IP Traffic: Пер. с англ. / У. Блэк. – СПб.: Питер, 2001. – 288 с.
3. *Попов В. Б.* Основы компьютерных технологий: Учеб. пособие для вузов / В. Б. Попов. – М.: Финансы и статистика, 2002. - 704с.
4. *Уэйнгроу К.* UNIX. Администрирование. – М.: ДМК Пресс, 2002. – 416 с.
5. *Холден Г.* Apache Server в комментариях. – М.: DiaSoft, 2000. – 480 с.
6. *Рич Б.* Apache. Настольная книга администратора. – М.: DiaSoft, 2002. –378 с.
7. *Хокинс С.* Администрирование Web-сервера Apache и руководство по электронной коммерции. – М.: Вильямс, 2001. – 330 с.
8. *Ньюкомер Э.* Веб-сервисы. XML, WSDL, SOAP и UDDI. – СПб.: Питер, 2003. – 256 с.

Лабораторная работа 5 Установка, настройка и администрирование proxy-сервера

Цель работы: настройка и администрирование Proxy-сервера **squid** под операционной системой Linux.

Лабораторная работа выполняется в локальной сети на рабочей станции с операционной системой Linux 7 или более поздней. В лабораториях кафедры операционная система Linux работает на компьютерах под управлением программного пакета VMware. Этот пакет позволяет создавать так называемые «виртуальные машины» – мнимые компьютеры, не зависящие от выполняющейся в текущее время на данном компьютере операционной системы (ОС). Для запуска ОС Linux необходимо запустить VMware на рабочей станции, выбрать из списка требуемую операционную систему и нажать кнопку «Power ON».
После окончания загрузки для входа в систему необходимо использовать имя пользователя ***root.***

Порядок выполнения лабораторной работы

Подготовка и допуск к работе. К выполнению лабораторной работы допускаются студенты, которые подготовились к работе и имеют не более двух невыполненных предыдущих работ.
Перед работой студент должен:

- предъявить преподавателю полностью оформленный отчет о предыдущей работе;
- ответить на вопросы преподавателя.

Студенты, которые не выполнили одно из вышеперечисленных требований, к работе не допускаются.

Отчёт по работе должен содержать:

- текст задания;
- перечень всех использованных в лабораторной работе команд и инструкций;
- вывод по работе.

Установка и администрирование Proxy-сервера

Proxy-сервер, осуществляющий доступ в Internet, предоставляет следующие возможности:

- централизованный выход в Internet через один сервер в сети;
- локальное хранение часто просматриваемых документов для увеличения скорости загрузки страниц (один пользователь загрузил документ с удаленного сервера в Internet, а все остальные после этого берут этот документ с Proxy-сервера);
- регулирование пропускной способности канала в зависимости от его нагрузки;
- авторизованный доступ в Internet (пользователь может загружать документы из Internet только при наличии логина и пароля).

Установка Proxy-сервера. Прежде чем устанавливать Proxy-сервер s*quid*, надо убедиться в том, что:

- машина, на которой будет работать *Proxy*, может соединиться (по WWW, FTP, Telnet – неважно) с другими машинами в сети;
- машины из внутренней сети могут соединиться с машиной, на которую устанавливается *Proxy,* опять же неважно каким клиентом;

Если же связь изнутри к *Proxy*-машине есть и эта *Proxy*-машина может общаться с внешним миром, можно переходить к настройке *squid*. *Squid* надо устанавливать из *packages* или *ports,* тогда есть уверенность, что все его компоненты будут размещены по нужным директориям. После установки должно получиться примерно следующее:

- двоичные файлы должны находиться в каталоге */usr/locl/sbin*;
- в каталоге */usr/local/etc* должна появиться директория *squid,* в которой лежит *squid.conf* – это его конфигурационный файл, его надо будет редактировать;
- в каталоге */usr/local/etc/rc.d* появился файл *squid.sh* – это основной стартовый файл (дело в том, что система при старте просматривает этот каталог и все, что найдет там типа **.sh,* запустит автоматически). Но можно его запустить и вручную, просто написав *./squid.sh.* Если такого файла нет, то при перезагрузке системы *squid* не будет запускаться;
- в каталоге */usr/local* образовалась директория *squid*, в которой две поддиректории: *cache* – там будет его кэш и *logs* – логи. Там же должен быть файл (возможно, он появится после первого запуска Proxy-сервера) *squid.out* – это основной лог-файл, в котором и будут сообщения об ошибках, если *squid* почему-либо не сможет стартовать нормально.

Для начала *squid.conf* можно редактировать незначительно. Там стоят значения «по умолчанию» и они вполне приемлемы. Единственное, что необходимо сделать – определиться, сколько мегабайт диска следует выделить под кэш. По умолчанию – 100 Мб. Для изменения этого значения найдите в *squid.conf* строчку

cache_swap 100

и поставьте подходящее значение. Максимальный объем можно оценить исходя из пропускной способности канала Internet за одни сутки.

Скорее всего, понадобится еще одно исправление. Дело в том, что нельзя запускать Proxy-сервер от имени *root*. Поскольку при старте машины *squid.sh* будет исполняться от имени *root*, то надо сделать следующее. Надо найти в конфигурационном файле строчку

cache_effective_user nobody nogroup

и «раскомментировать» ее. Это будет означать, что в процессе работы *squid* будет иметь права «псевдоюзера» *nobody*.

На самом деле это не совсем правильно. Правильнее завести нового «псевдоюзера» *squid* (или еще как-нибудь - *www, cache ...*) и в конфиг-файле вписать именно его, а не *nobody*. После этого надо будет поправить владельца для директории */usr/local/squid*. Для этого выполните команду

chown -R nobody /usr/local/squid

Здесь **-R** означает, что меняется владелец не только директории, но и рекурсивно меняется владелец всего содержимого поддиректорий. Если Вы завели специального «псевдоюзера», то, естественно, в команде вместо *nobody* укажите его имя. Теперь осталось сформировать «внутреннюю структуру кэша». Кстати, если этого не сделать, то *squid* при запуске сам подскажет вам: «**запустите программу squid –z**». Естественно, это надо сделать. Только учтите, что */usr/local/sbin* обычно не прописан в PATH даже у *root*, поэтому лучше набрать полный путь

/usr/local/sbin/squid -z

Теперь можно попытаться запустить *squid*. Зайдите в */usr/local/etc/rc.d* и запустите *./squid.sh*. На консоли должно появиться сообщение от *squid*: "Ready to serve requests" («готов обслуживать запросы»).

Настройка Proxy-сервера. Общая настройка Proxy-сервера чаще всего не вызывает сложностей. Сложности обычно вызывают три обстоятельства: настройка *ACL* (*access control*

list – список прав доступа) и правил для них, настройка дополнительных программ вроде «баннерорезалок» и настройка ограничений использования канала.
Настройка ACL. Обратимся к соответствующему месту файла squid.conf и прокомментируем его.
ACL прописываются в виде строки ***acl имя_acl тип_acl параметры ACL или acl имя_acl тип_acl «файл»*** – при этом в файле сохраняется по одному значению на строку.
Итак, сначала типы списков.
acl aclname src ip-address/netmask – в этом *acl* описывается *ip*-адрес или сеть, принадлежащая клиентам *squid*. Например:
acl vasya src 192.168.1.1/255.255.255.255 – описывает единственную машину с адресом 192.168.1.1 и назначает ей ACL с именем *vasya*.
acl office src 192.168.1.1/255.255.255.0 – описывает диапазон машин с адресами 192.168.1.1-.254 и назначает этому ACL имя *office*. Если диапазон необходимо сузить, то необходимо либо изменить маску подсети, либо воспользоваться явным указанием: *acl vip_user src 192.168.1.1-192.168.1.5/255.255.255.0.* Здесь *squid* выбирает тот диапазон адресов, который окажется меньше либо по маске, либо по явному указанию.
acl aclname dst ip-address/netmask – этот тип ACL описывает уже сервер, страницы с которого будут запрашивать клиенты. Следует отметить, что в этом типе ACL задается не символьный адрес сервера, а *ip*.
acl aclname srcdomain.domain.ru – описывает клиентов, но уже не по *ip*-адресам, а по реверсным DNS. Это значит, что нет разницы, какие *ip*-адреса принадлежат клиентам, главное, чтобы они определялись *dns*. Соответственно под это правило попадут все клиенты, стоящие в домене *domain.ru*.
acl aclname dstdomain .domain.ru – описывает сервер. Сравнивается с запросом из URL. Под этот ACL попадут все серверы третьего уровня домена *domain.ru*.
acl aclname srcdom_regex [-i] xxx acl aclname dstdom_regex [-i] xxx – описания аналогичны предыдущим, но теперь для выяснения, подходит ли правило под запрос, используются *regex*-правила. Если символьный адрес не смог определиться из *ip*-адреса, к запросу будет применена строка *none*.
acl aclname time [day-abbrevs] [h1:m1-h2:m2] – ACL, описывающий время. Коды дней недели определяются так: *S* – Sunday – Воскресенье, *M* – Monday – Понедельник, *T* – Tuesday – Вторник, *W* – Wednesday – Среда, *H* – Thursday – Четверг, *F* – Friday – Пятница, *A* – Saturday – Суббота.
Ну а вместо *h1:m1* и *h2:m2* вставляется время. Запомните – *h1:m1* всегда должно быть меньше *h2:m2*.
Итак, *acl worktime time MTWHF 08:00-17:00* описывает рабочее время с понедельника по пятницу, с 8 утра до 5 вечера, *acl weekday time SA* описывает целиком субботу с воскресеньем, а *acl evening time 17:00-23:59* описывает время до полуночи. Если необходимо описать всю ночь, то приходится заводить два ACL: первый – с вечера до полуночи, а второй – с полуночи до утра.
acl aclname url_regex [-i] ^http:// – *regex*-правила, применяемые ко всему URL.
acl aclname urlpath_regex [-i] \.gif$ – аналогичные правила, применяемые к URL.
acl aclname port 80 70 21 – ACL, описывающий порты. Вместо простого перечисления можно указать диапазон, например, 1-1024.
acl aclname proto HTTP FTP – ACL, описывающий протокол, по которому клиент желает сделать запрос на сервер.
acl aclname method GET POST – метод, которым передаются данные клиента серверу.
acl aclname browser [-i] regexp – *regexp*-запрос на клиентский браузер. Вычисления основаны на заголовке *User-Agent*, который пересылает браузер.
acl aclname ident username – ACL описывает имя пользователя, от которого запущена программа на клиентской машине. Имя узнается с помощью *ident*-сервера.

acl aclname ident_regex [-i] pattern – то же самое, но основанное на *regex*-правилах.
acl aclname proxy_auth username acl aclname proxy_auth_regex [-i] pattern – ACL, описывающий имя пользователя. Это имя возвращает внешняя авторизующая программа.
acl aclname maxconn number – это правило сработает, если клиент сделает больше *number* запросов к кешу.
acl req_mime_type mime-type1 – правило, срабатывающее при *upload* файлов клиентом. Заметьте – *uplode*, а не скачивание.
Здесь представлены не все описания ACL, но большинство, необходимых в повседневной практике. Для более полного знакомства с описанием следует обратиться к исходному тексту файла *squid.conf* – там все описано, правда, по-английски.
Итак, создадим правила обычной сети:
acl all src 0.0.0.0/0.0.0.0 acl office src 192.168.1.0/255.255.255.0
all – правило, описывающее все машины, и *office* – описывающее все машины в подсети 192.168.1.0.
http_access allow office http_access deny all
Эти два правила описывают полный доступ машинам, описываемым *acl office,* и запрещает доступ машинам, описываемым *all*. В приведенном примере есть конфликт в описания прав доступа: машины, попадающие под правило *all* (а по этому правилу все запрещено) не могут использовать Proxy-сервер. Тут в дело вступает порядок просмотра ACL – они просматриваются в порядке объявления, и если сработало одно правило, то другие уже не просматриваются.
К примеру, если мы введем в дополнение ACL **acl vasya src 192.168.1.100/255.255.255.255** и расположим правила так:
http_access allow office http_access deny vasya http_access deny all ,
то машина с *ip*-адресом 192.168.1.100 по-прежнему будет иметь возможность соединяться через Proxy сервер;
а если так:
http_access deny vasya http_access allow office http_access deny all ,
то все будет в порядке. Остальные офисные машины не попадают под действие первого правила.
Если в списке нет ни одного правила, то запрос будет отвергнут. Если ни одно правило не сработало, то за основу берется последнее. Если, к примеру, мы заменим предпоследнее правило на *http_access allow all*, то нашим Proxy-сервером смогут пользоваться абсолютно все (кроме *vasya*), кто сможет соединиться с портом *squid*. Так что будьте внимательнее. Но авторы *squid* постарались: даже если последнее правило будет разрешающим для всех, то запрос будет отвергнут. Это поможет избежать дыр в Proxy-сервере.
На основе этих же списков-правил так же управляется и доступ к другим возможностям Proxy-сервера (см. файл *squid.conf*, где все расписано).
Правила – правилами, но, предположим, что в сети появились пользователи, которые честно подключаются к серверу и начинают выкачивать гигабайтами запрещенную информацию. При этом занесение этих сайтов в *deny*-список вызывает их возмущение.
На этот счет придумали много вещей, но самым эффективным остается сокращение канала для таких пользователей: доступ есть, но качается плохо, возразить им нечего – такая ситуация в Internet не редкость.
Итак, давайте разберемся с «траффик-шейпингом» – именно так это называется. В *squid* же это называется *delay-pool*. Заметим, что *squid* при сборке должен быть собран с опцией *--enable-delay-pools*.
Итак, сначала разберемся, какие есть пулы. Пулы делятся на три класса. Первый, и самый простой, это когда всему *acl* ограничивается трафик до определенной величины. Второй – когда отдельно ограничивается трафик для одной машины из подсети и для всей подсети. И

третий класс – когда ограничивается трафик для отдельных машин, для подсети класса *C* или меньше и для подсети класса *B*.
Итак, давайте обыграем ситуацию, когда в сети завелся «дискокачальщик».
delay_pools 1 – у нас всего один пул.
delay_class 1 1 – первый пул первого класса.
delay_access 1 allow vasya
delay_access 1 deny all
В первый пул попадают только машины, описываемые ACL *vasya*. Остальные работают, как им положено, ведь им доступ к первому пулу запрещен.
delay_parameters 1 800/64000
Вот и все. Теперь файлы и страницы объемом до 64 Кб будут скачиваться на максимальной скорости, а то, что больше этого – на скорости 800 б/с.
Или совсем уж радикальная мера:
delay_parameters 1 800/800 – и «злобный качальщик» все будет качать на скорости 800 б/с.
Но даже в не очень большой сети будут возникать ситуации, когда все хотят что-то качать, в итоге никому ничего не хватает.
Исправляем строчку с *delay_pools* на *delay_pools 2*. Теперь у нас будет два пула.
delay_class 2 2 – второй пул будет второго класса (совпадение номеров чисто случайно) – первый – это *vasya*.
delay_access 2 allow office
delay_access 2 deny all
Во второй пул попадают только машины с ACL *office*.
delay_parameters 2 64000/64000 4000/4000
В итоге вся подсеть, описываемая *office*, будет использовать канал не больше 512 Кбит/с (64 Кб/с), но каждый отдельный хост будет качать не более 4 Кб/с. Этим правилом очень легко разграничить по скорости разные подсети, использующие один канал.
К примеру, у нас есть две подсети, описываемые *office* и *office1*. При этом *office* не должна иметь никаких ограничений на канал (примем канал за 256 Кбит) в целом, но каждый из *office* не должен качать быстрее 6 Кб/с. А *office1* – это пользователи, которым всем и 5 Кб/с хватит.
Создаем два пула второго класса и прописываем для них ACL. Затем определяем этим пулам параметры.
delay_parameters 3 -1/-1 6000/6000 – это определение для *office* (ему отдан номер пула 3).
delay_parameters 4 5000/5000 -1/1 – а это для *office1*.
В итоге после применения этих правил получаем все, что заказано – первый офис грузит канал как хочет (-1/-1), но никто из сотрудников больше 6 Кб/с не получает. А второй офис грузит канал не больше 5 Кб/с, но в распределении этих 5 Кб/с между сотрудниками нет никаких правил.
Понятно, что в описание пулов можно заложить и другие параметры, например, время, место доступа и т. д. Остается еще одна маленькая вещь, которую нельзя оставить без внимания. И эта вещь – навязанная реклама через баннеры и другие объекты. Для того, чтобы такую рекламу не пропустить на броузер, каждый URL, который передается *squid*, первоначально передается *редиректору*. И тот либо возвращает прежний URL в случае, если все в порядке, либо возвращает тот, который, по его мнению, более правильный. А кто мешает нам перехватывать обращения к баннерам и счетчикам и вместо них подсовывать свою картинку? В итоге страницы можно заполнить прозрачными окошками.
Итак, в *squid.conf* прописываем строку
redirect_program /squid/bin/redirector
где */squid/bin/redirector* – путь до выполняемой программы, которая как раз и обеспечивает разбор URL. Ее можно написать на чем угодно, но наиболее предпочтительным является Perl

– этот язык как раз предназначен для подобного рода работ. Полная версия *редиректора* лежит на *http://linuxnews.ru/redirector*.
Описание директив squid. Описание директив содержится непосредственно в файле конфигурации *squid.conf* и в документации, прилагаемой к данной лабораторной работе.

Задание на лабораторную работу

Лабораторная работа выполняется по вариантам.

Вариант	Задание
Подгруппа 1, 5	Сконфигурировать Proxy-сервер на порт 3128; разрешить доступ с *ip* 10.10.146.150 с 10-00 до 15-00 ч; запретить доступ с *ip* 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к доменам **.khb.ru*
Подгруппа 2, 6	Сконфигурировать Proxy-сервер на порт 8080; разрешить доступ с *ip* 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к файлам **.gif*; запретить доступ к домену **.khb.ru* с *ip* 10.10.146.150
Подгруппа 3, 7	Сконфигурировать Proxy-сервер на порт 12345; разрешить доступ с *ip* 10.10.146.176; запретить доступ к файлам **.cgi*; запретить доступ к домену **.khb.ru* с *ip* 10.10.146.150 с 10-00 до 15-00 ч
Подгруппа 4, 8	Сконфигурировать Proxy-сервер на порт 1345; запретить доступ с *ip* 10.10.146.176; запретить метод *POST*; запретить доступ к домену **.khstu.ru* и к файлам **.gif* с *ip* 10.10.146.150 с 10-00 до 15-00 ч

При защите необходимо знать назначение используемых директив, уметь объяснить информацию, содержащуюся в log-файлах
Для редактирования файлов конфигурации и навигации по файловой системе удобно использовать программу ***mc***. Для ее загрузки необходимо набрать в командной строке
[root@lis] $ mc
Интерфейс программы ***mc*** интуитивно понятен и не представляет трудностей при использовании.

Содержание отчета

1. Содержимое конфигурационных файлов (без комментариев в тексте).
2. Отрывок из log-файлов, демонстрирующий обращения через прокси-сервер.

Контрольные вопросы

1. Назначение сервера прокси.
2. В каком файле содержится информация о конфигурации Proxy-сервера?
3. Какие протоколы кэшируются прокси?
4. Какие условия должны быть выполнены перед установкой Proxy-сервера?
5. Что такое список прав доступа?
6. Общий синтаксис ACL.

Библиографический список

1. *Мартин Майкл Дж.* Введение в сетевые технологии = Understanding the Network: Практическое руководство по организации сетей / Мартин Майкл Дж. – М.: ЛОРИ, 2002. – 660 с.
2. *Бэндл Дэвид.* Защита и безопасность в сетях Linux = Linux security toolkit: Пер. с англ. / Бэндл Дэвид. – СПб.: Питер, 2002. – 480 с. (Для профессионалов. Б-ка Linux).
3. *Мельников Д. А.* Информационные процессы в компьютерных сетях: Протоколы, стандарты, интерфейсы, модели... / Мельников Д. А. – М.: КУДИЦ-ОБРАЗ, 1999. – 256 с. (Б-ка профессионала).

Практическое занятие 6 Создание защиты компьютерной сети с использованием брандмауэра

Упражнение 1. Сетевое сканирование nmap

nmap — свободная утилита, предназначенная для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети (портов и соответствующих им служб). Изначально программа была реализована для систем UNIX, но сейчас доступны версии для множества операционных систем [5].

nmap использует различные методы сканирования, таких как UDP, TCP (connect), TCP SYN (полуоткрытое), FTP-proxy (прорыв через ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN- и NULL-сканирование. Nmap также поддерживает большой набор дополнительных возможностей, а именно: определение операционной системы удалённого хоста с использованием отпечатков стека TCP/IP, «невидимое» сканирование, динамическое вычисление времени задержки и повтор передачи пакетов, параллельное сканирование, определение неактивных хостов методом параллельного ping-опроса, сканирование с использованием ложных хостов, определение наличия пакетных фильтров, прямое (без использования portmapper) RPC-сканирование, сканирование с использованием IP-фрагментации, а также произвольное указание IP-адресов и номеров портов сканируемых сетей.

1. Создайте виртуальную машину, используя инструкции из упражнения 1 практического занятия №1. Используйте установочный образ ***RedHat Enterprise Linux 6***.
2. Ping-сканирование - наиболее распространенным и простым способом сканирования является простое ping-сканирование, которое заключается в отправке ICMP пакетов на исследуемый узел сети. Просканируйте заданный преподателем узел командой

ping 192.168.58.103

3. TCP Connect - метод сканирования, при котором сканирующая машина пытается установить соединение со сканируемой. Успешный результат говорит о том, что порт открыт, неудачный — о том, что он закрыт или фильтруется. Просканируйте заданный преподателем узел командой

nmap –sT 192.168.58.103

4. Увеличьте информативность вывода результатов сканирования, используя ключ -v.
5. TCP-SYN - режим полуоткрытого сканирования. При вызове nmap посылает SYN-пакет. Если в ответе присутствуют флаги SYN или ACK, считается, что порт открыт. Флаг RST говорит об обратном. Сканирование осуществляется только при наличии прав суперпользователя (root). Просканируйте заданный преподателем узел командой

nmap –sS 192.168.58.103

6. Межсетевой экран или другие защитные средства могут ожидать приходящие SYN-пакеты. Существует еще целая группа возможных способов сканирования, альтернативных TCP SYN. Это FIN, Xmas Tree и NULL-сканирования. Просканируйте заданый преподателем узел командами

nmap –sF 192.168.58.103
nmap –sX 192.168.58.103
nmap –sN 192.168.58.103

7. Сканирование протоколов IP - сканируемому узлу передаются IP пакеты без заголовков для каждого протокола сканируемого хоста. Если получено сообщение, говорящее о недоступности протокола, то этот протокол не поддерживается хостом. В противном случае — поддерживается. Просканируйте заданный преподателем узел командой

nmap –sO 192.168.58.103

8. ACK сканирование заключается в передаче ACK пакетов на сканируемый порт. Если в ответ приходит RST-пакет, порт классифицируется как нефильтруемый. Просканируйте заданный преподателем узел командой

nmap –sA 192.168.58.103

Упражнение 2. Изучение возможностей файервола Windows 7

Первая версия Windows Firewall позволяла очень много сделать в плане защиты от взлома, например настраиваться как с помощью Group Policy Object (GPO) так и из командной строки. Однако возможностей защиты явно не хватало в таких областях, как настройка правил доступа и фильтрация исходящего трафика. Версия Windows Firewall, которая поставляется в составе Windows Vista, имеет 2 существенных нововведения:

Microsoft включила расширенный интерфейс управления для того, чтобы дать администраторам очень подробные правила настройки рабочих станций. Более того, Брандмауэр может быть настроен из оснастки Групповая политика, что означает для корпоративных IT-специалистов более легкий путь навязывания политик организации, например, запрещение специфической активности – обмен мгновенными сообщениями или сети peer-to-peer.

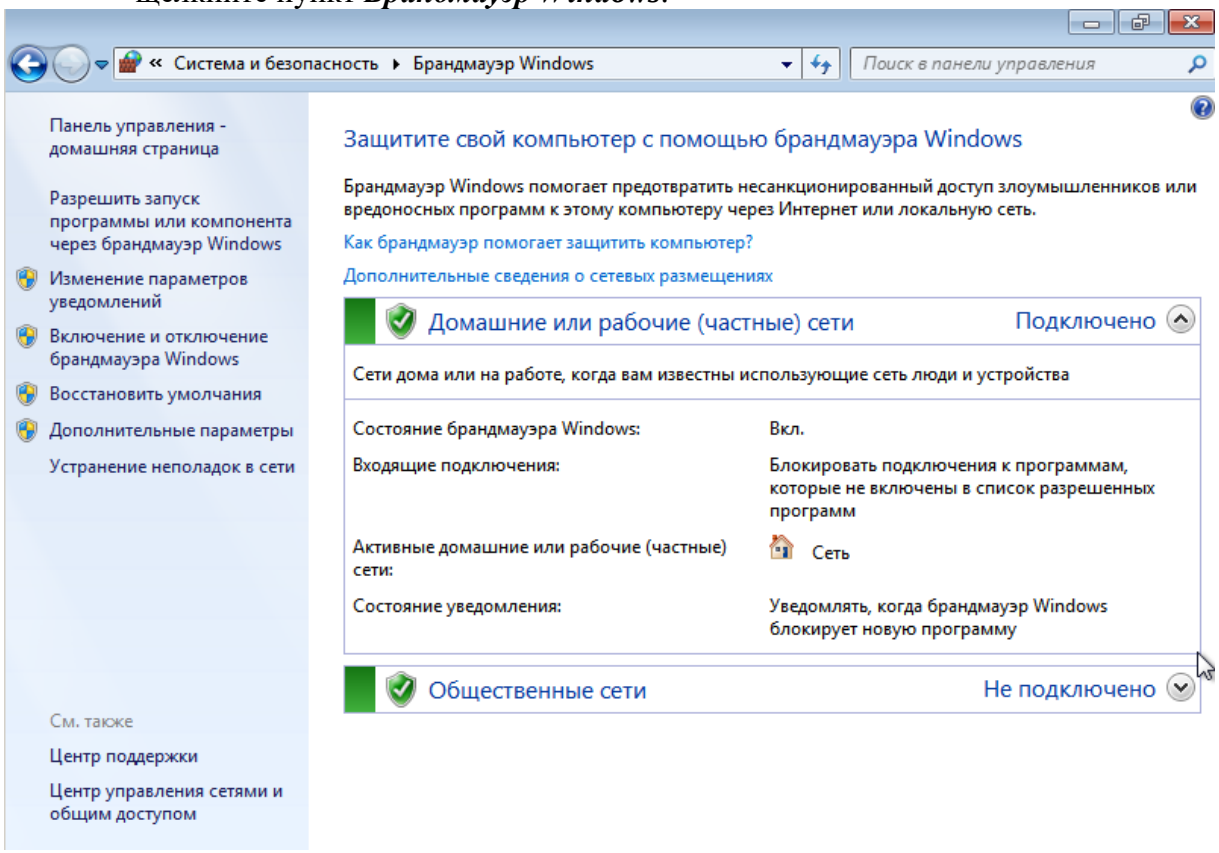
Управление МЭ сосредоточено в MMC-консоли, получившей название Windows Firewall with Advanced Security. Некоторые параметры брандмауэра можно по-прежнему настроить централизованно через Group Policy или же настроить брандмауэр локально при помощи командной утилиты Netsh. Как и другие оснастки, Windows Firewall with Advanced Security поддерживает удаленный доступ, который позволяет управлять настройкой брандмауэра на удаленной станции.

Брандмауэр Windows блокирует входящий трафик по умолчанию, так что необходимо сразу же настроить список Exceptions, если планируется работать с сетевым приложением. Exceptions — это то, что Microsoft называет правилами (rules), а более точно, списки ACL.

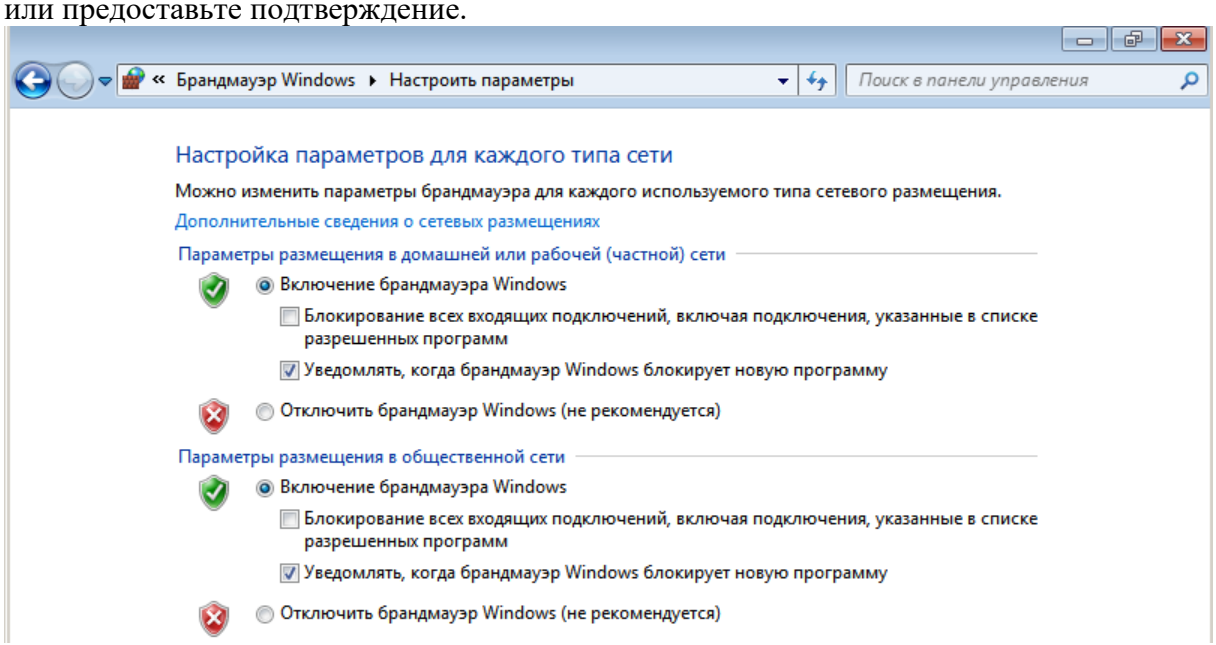
Во многих случаях хост-брандмауэры, написанные независимыми разработчиками, выдают предупреждения при попытке установить исходящее соединение и запрашивают разрешение у пользователя. На основе полученного ответа брандмауэр может даже создать правило реагирования на такие события в будущем. Брандмауэр Windows поступает иначе — весь исходящий трафик по умолчанию разрешен. Создание исключений для исходящего трафика — процедура несложная, но она требует использования новой оснастки. Большинство пользователей об этом даже и не вспомнит, но администраторы систем должны как следует разобраться во всех обязательных настройках Windows Firewall with Advanced Security.

Оснастка позволяет настроить все функции брандмауэра. В левой панели можно выбрать Inbound Exceptions, Outbound Exceptions, Computer Connection Security или Firewall Monitoring, дважды щелкнуть по выбранному элементу — узлу структуры — и увидеть дополнительные настройки в центральной панели. На правой панели располагается список всех доступных действий для выбранного узла. Такое построение консоли делает процесс настройки брандмауэра интуитивно понятной процедурой; например, если нужно включить или отключить правило, достаточно щелкнуть правой кнопкой мыши, вызвав контекстное меню, или же на правой панели выбрать требуемое действие. Большинство действий вступает в силу немедленно, что упрощает отладку при настройке брандмауэра. Чтобы просмотреть и задать свойства брандмауэра, необходимо открыть контекстное меню Windows Firewall with Advanced Security в левой панели и выбрать Properties.

1. Упражнение выполняется на ранее установленной виртуальной машине с операционной системой Windows 7.
2. Откройте компонент ***Брандмауэр Windows***. Для этого нажмите кнопку ***Пуск*** и выберите пункт ***Панель управления***. В поле поиска введите брандмауэр и затем щелкните пункт ***Брандмауэр Windows***.

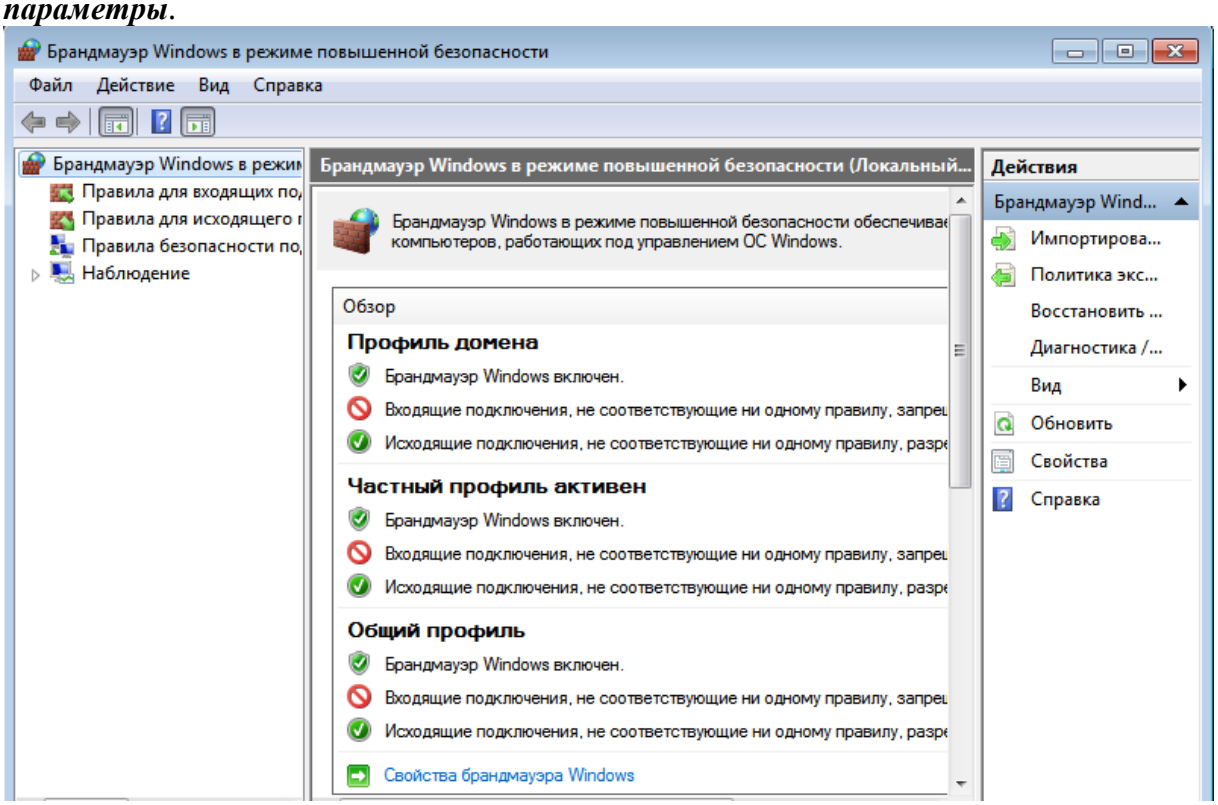


В левой области выберите ***Включение и отключение брандмауэра Windows***. Если отображается запрос на ввод пароля администратора или его подтверждения, укажите пароль или предоставьте подтверждение.



Блокирование всех входящих подключений, включая подключения, указанные в списке разрешенных программ. Этот параметр блокирует все неожидаемые попытки подключения к компьютеру. Этот параметр служит для максимальной защиты компьютера, например, при подключении к общедоступной сети в отеле или в аэропорте или в периоды распространения

через Интернет особо опасных вирусов-червей. При использовании этого параметра вы не будете уведомлены о блокировке программ брандмауэром Windows, и все программы из списка разрешенных программ будут проигнорированы.
Включение брандмауэра Windows. Этот вариант выбран по умолчанию. Если брандмауэр Windows включен, то установка связи большинства программ брандмауэром блокируется. Если следует разрешить программе устанавливать связь через брандмауэр, можно добавить ее в список разрешенных программ.
Убедитесь, что брандмауэр Windows включен, и перейдите во вкладку ***Дополнительные параметры***.



Создайте дополнительное правило для ***Брандмауэра Windows***. Выберите перечень правил ***Правила для входящих\исходящих подключений*** и нажмите кнопку ***Создать правило***.
Создайте дополнительное правило для входящих подключений для порта 80/TCP. Обясните назначение порта.
Проведите сканирование АРМ, защищённого настроенным вами межсетевым экраном. Сделайте выводы о защищённости.

Упражнение 3. Изучение возможностей межсетевого экрана iptables.
iptables — утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) netfilter для ядер Linux версий 2.4, 2.6, 3.x, 4.x . Для использования утилиты iptables требуются привилегии суперпользователя (root).Иногда под словом iptables имеется в виду и сам межсетевой экран netfilter.
Все пакеты пропускаются через определенные для них последовательности цепочек. При прохождении пакетом цепочки, к нему последовательно применяются все правила этой цепочки в порядке их следования. Под применением правила понимается: во-первых, проверка пакета на соответствие критерию, и во-вторых, если пакет этому критерию соответствует, применение к нему указанного действия. Под действием может подразумеваться как элементарная операция (встроенное действие, например, ACCEPT, MARK), так и переход в одну из пользовательских цепочек. В свою очередь, действия могут быть как терминальными, то есть прекращающими обработку пакета в рамках данной базовой цепочки (например, ACCEPT, REJECT), так и нетерминальными, то есть не

прерывающими процесса обработки пакета (MARK, TOS). Если пакет прошел через всю базовую цепочку и к нему так и не было применено ни одного терминального действия, к нему применяется действие по умолчанию для данной цепочки (обязательно терминальное).

1. Загрузите виртуальную машину с ОС Linux.

Проверьте наличие в составе ОС межсетевого экрана ***iptables***. При необходимости доустановите/переустановите компоненты системы с учётом следующих требований:
на виртуальной машине должны использоваться 2 сетевых адаптера, один из которых используется для связи с сетью учебной аудитории, а второй – для связи с виртуальной сетью ***VMware vSphere***;
во внутренней сети ***VMware vSphere*** должно входить минимум 2 виртуальных машины.
Используя справочные материалы, настройте следующую конфигурацию МЭ ***iptables***:
обеспечьте связь через виртуальную машину с МЭ сети класса и виртуальной сети;
добавьте правило, запрещающее прохождение ICMP трафика между сетями.

2. Проверьте настройки МЭ, используя компьютеры в сети аудитории и ***VMware vSphere.***

Лабораторная работа 6 Установка, настройка маршрутизатора сети

Цель работы

Изучить принципы маршрутизации в IP-сетях, получить практические навыки настройки программных маршрутизаторов на базе операционной системы (ОС) Linux с применением маршрутизации интерфейсов и маршрутизации виртуальных локальных сетей (VLAN).

Теоретические сведения

Маршрутизация включает в себя следующие частные задачи:

1. Обмен информацией о топологии сети. Реализуется протоколами маршрутизации.
2. Определение оптимальных маршрутов и построение таблиц маршрутизации.
3. Продвижение пакета маршрутизаторами на основании таблиц маршрутизации.

В данной лабораторной работе рассматриваются вопросы продвижения пакетов. Информация об оптимальных маршрутах представляется в маршрутизаторе в виде таблицы маршрутизации. В случае адресации без масок таблица маршрутизации имеет вид:

Таблица 1.

Формат таблицы маршрутизации без масок

Адрес назначения	IP след. маршр.	Интерф.	Метрика

В таблице:

- Адрес назначения – это IP-адрес сети или конкретного узла.

- IP следующего маршрутизатора – адрес следующего транзитного маршрутизатора на пути к адресу назначения.
- Интерфейс – это идентификатор сетевого интерфейса маршрутизатора, через который нужно передавать данные следующему маршрутизатору.
- Метрика – абстрактная характеристика качества маршрута. В качестве метрики может выступать количество транзитных узлов («хопов»), пропускная способность и т.п. Меньшее значение метрики всегда соответствует лучшему маршруту.

При поиске маршрута для продвижения IP-пакета предпочтение отдается специфическим маршрутам (то есть маршрутам до конкретного узла, а не сети), даже в том случае, если метрика этого маршрута хуже.

При использовании масок алгоритм работы маршрутизатора при продвижении пакета несколько усложняется. В таблицу маршрутизации добавляется колонка с маской, соответствующей адресу назначения. Рассмотрим пример:

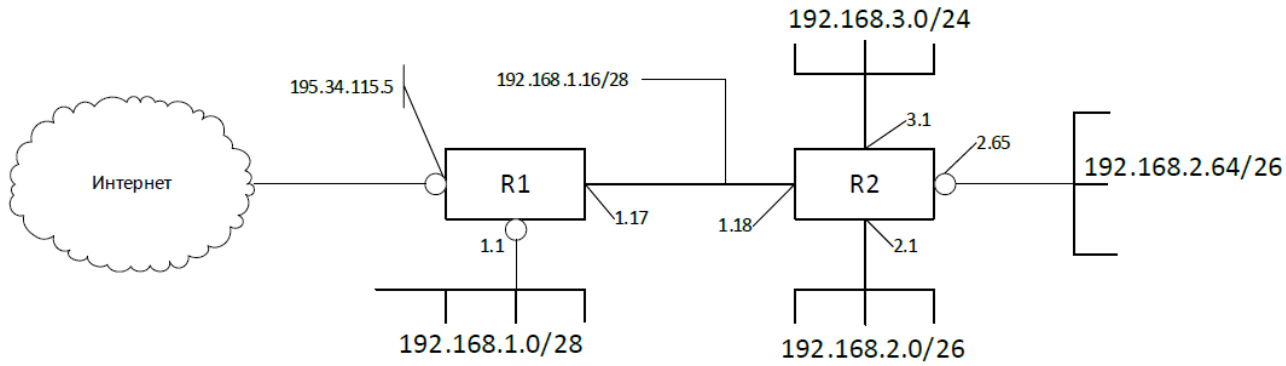


Рис. 1. Пример структуры сети

Для данного примера в таблице маршрутизации R2 могут быть следующие записи:

Таблица 2.

Пример таблицы маршрутизации с масками

Адрес назначения	Маска	Адрес след. маршр.	Интерфейс	Метрика
192.168.1.16	255.255.255.240	-	192.168.1.18	0
192.168.3.0	255.255.255.0	-	192.168.3.1	0
192.168.2.0	255.255.255.192	-	192.168.2.1	0
192.168.2.64	255.255.255.192	-	192.168.2.65	0
192.168.1.0	255.255.255.240	192.168.1.17	192.168.1.18	1
Default	0.0.0.0	192.168.1.17	192.168.1.18	16

Продвижение пакета основывается на следующем алгоритме:

1. Маршрутизатор извлекает из пакета IP-адрес назначения.
2. Поиск специфического маршрута, то есть записи, в которой адрес назначения равен целевому IP-адресу. Если запись найдена, то используется этот маршрут, иначе на шаг 3.
3. Поиск неспецифического маршрута. Включает следующие шаги:
   - для каждой записи таблицы маршрутизации выполняется операция IPd & M, где IPd – целевой адрес из пакета, M - маска из записи таблицы маршрутизации;
   - если результат равен адресу назначения IP, запись отмечается подходящей.
4. Из всех подходящих маршрутов выбирается наиболее специфический. Такой маршрут содержится в записи с наиболее длинной маской.
5. Среди подходящих маршрутов, обладающих одинаковой "специфичностью", выбирается маршрут с меньшей метрикой.

Задание

Настроить взаимодействие двух IP-сетей между собой и с внешней сетью средствами программного маршрутизатора на базе ОС Linux (рис. **2**). Настроить простейшие правила фильтрации трафика средствами ОС Linux.

Ситуация 1. Сети изолированы друг от друга физически, т.е. построены на различных коммутаторах, не связанных друг с другом непосредственно.

Ситуация 2. Изоляция сетей обеспечивается за счет применения технологии виртуальных локальных сетей.

Проверить настройку маршрутизации и фильтров на примере взаимодействия рабочих станций PC-1 и PC-2, принадлежащих различным сетям.

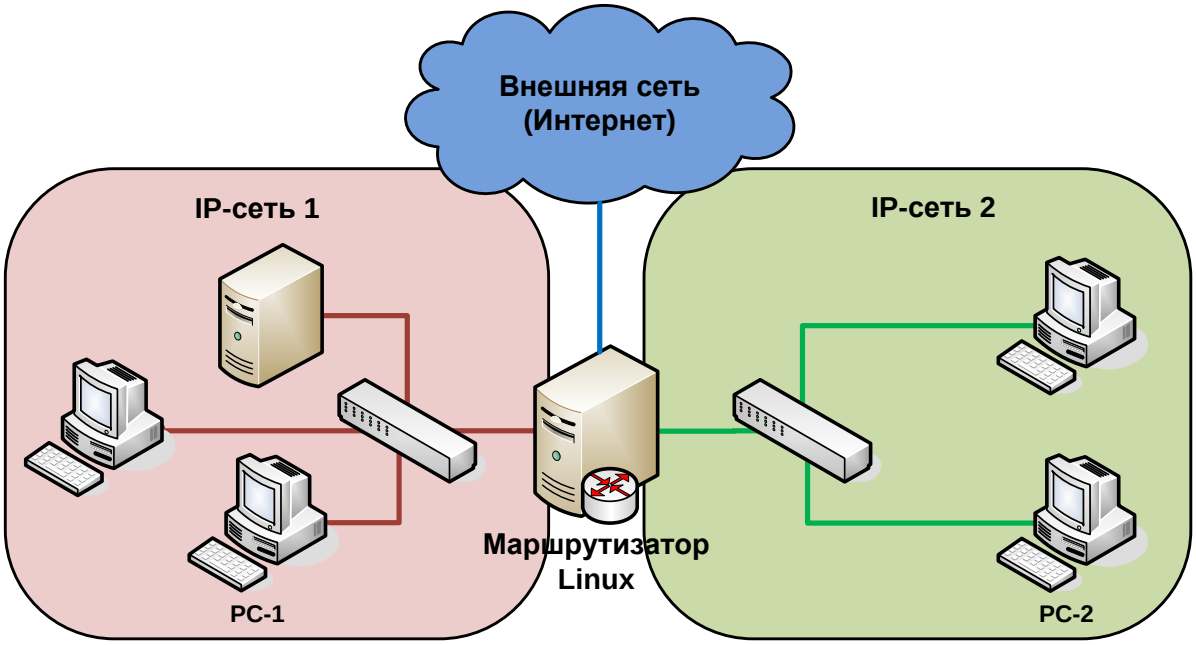


Рис. **2**. Принципиальная схема взаимодействия сетей для лабораторной работы №1

Схема ЛВС

В приведённых примерах схем ЛВС для лабораторной работы указана адресация для компьютеров PC151 (ПК-1), PC152 (ПК-2), PC154 (ПК-4) лаборатории. При использовании других компьютеров IP-адреса будут другими (см. прил. 1).

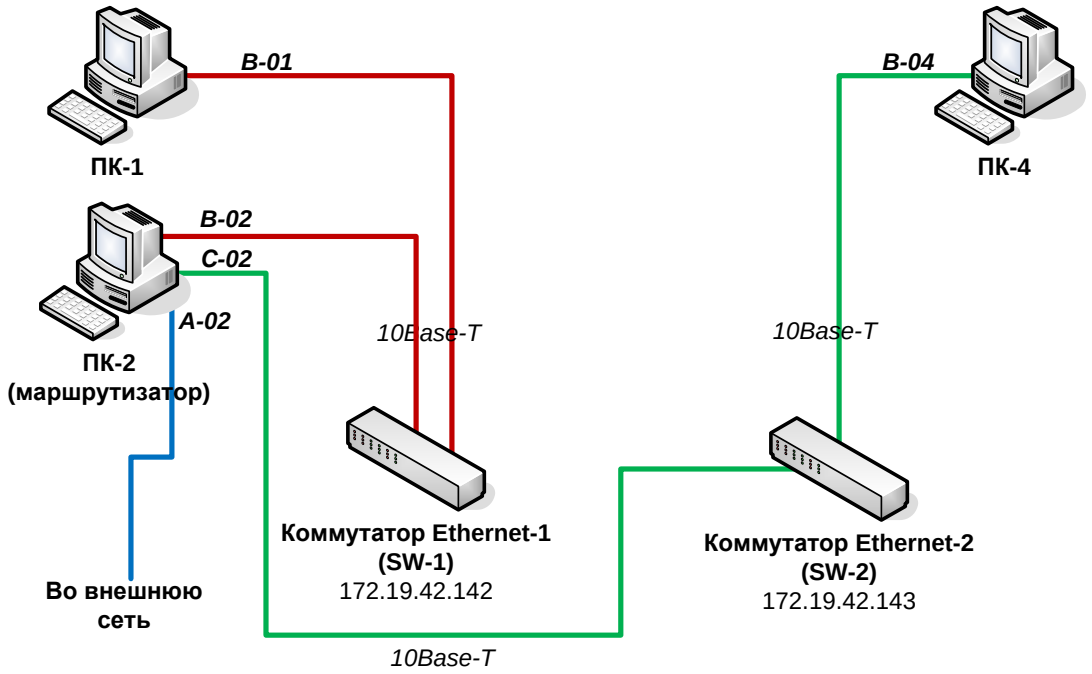


Рис. 3. Физическая схема ЛВС для лабораторной работы №1 (ситуация 1)

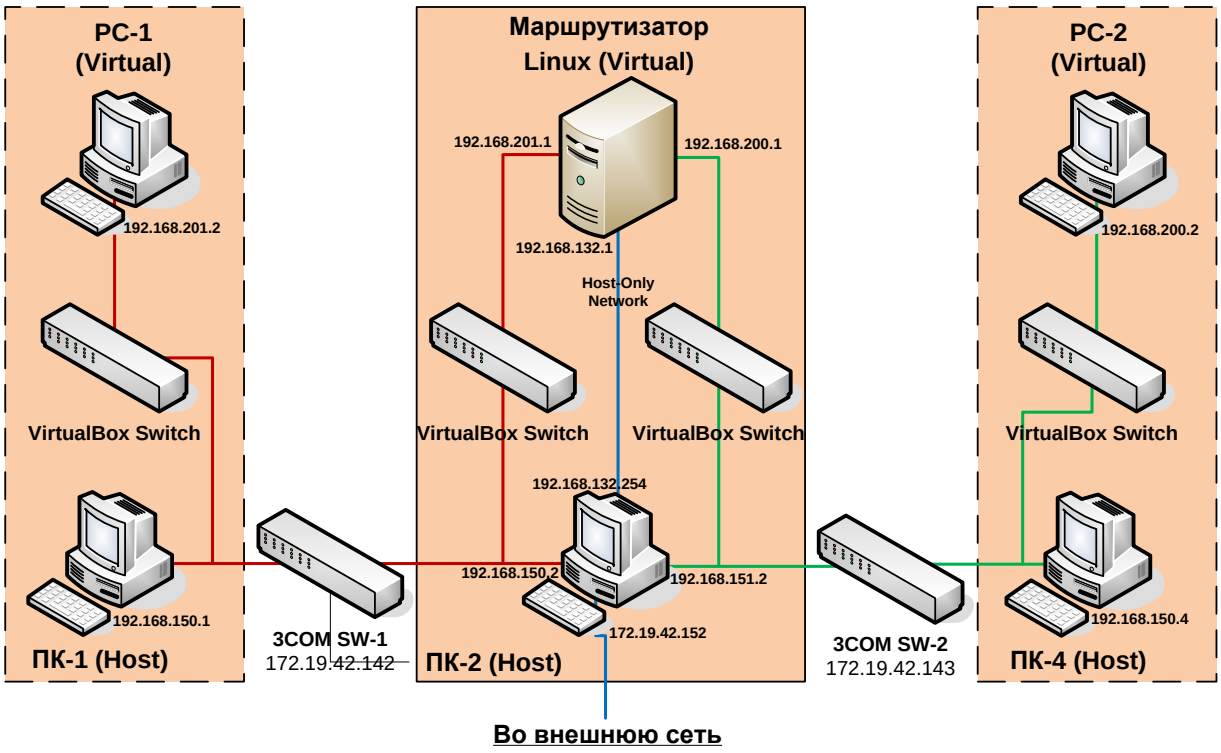


Рис. 4. Логическая схема ЛВС для лабораторной работы №1 (ситуация 1). IP-сеть 1 имеет адрес 192.168.201.0/24, IP-сеть 2 имеет адрес 192.168.200.0/24, внешняя сеть подключена через 192.168.132.0/24

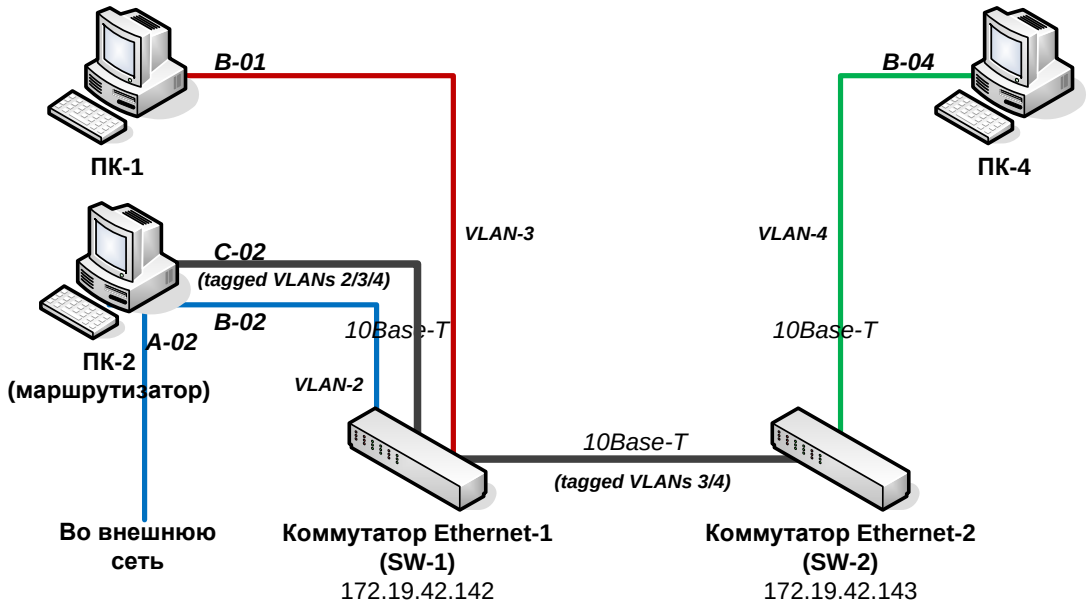


Рис. 5. Физическая схема ЛВС для лабораторной работы №1 (ситуация 2)

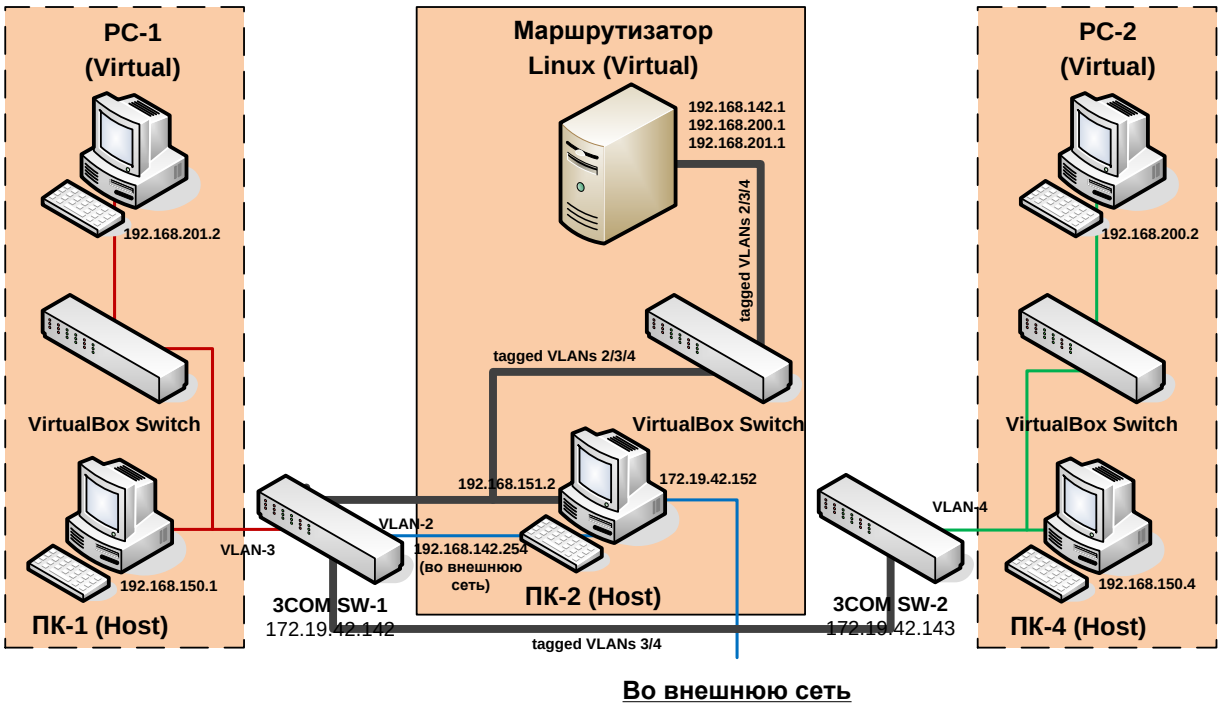


Рис. 6. Логическая схема ЛВС для лабораторной работы №1 (ситуация 2). IP-сеть 1 имеет адрес 192.168.201.0/24, IP-сеть 2 имеет адрес 192.168.200.0/24, внешняя сеть подключена через 192.168.142.0/24

Порядок выполнения работы

Перед выполнением лабораторной работы необходимо:

1. Подготовить схемы сети с указанием ip-адресов в соответствии с вариантом.
2. Подготовить виртуальные машины PC-1, PC-2, на которых может быть установлена любая сетевая операционная система, например Microsoft Windows XP или Linux.
3. Подготовить виртуальную машину, которая будет выполнять роль маршрутизатора. Рекомендуется использовать дистрибутив Debian GNU/Linux. Для данной виртуальной машины использовать сетевые адаптеры ***«PCnet»***, обеспечивающие передачу тэгированных кадров VLAN.

   В установленной ОС Linux должна быть включена поддержка VLAN. В Debian необходимо установить соответствующий пакет командой

   ```
   apt-get install vlan
   ```

   Рекомендуется также установить графическую версию текстового редактора vi. В Debian соответствующий пакет устанавливается командой

   ```
   apt-get install vi-gnome
   ```

   Только для Debian! В случае, если образ ОС Debian Linux копируется/клонируется, необходимо отключить контроль MAC-адресов интерфейсов в файле «/etc/udev/rules.d/75-persistent-net-generator.rules» (изменять файл можно только с правами администратора). Для этого внести в приведенный ниже раздел строку с маской сетевых адаптеров VirtualBox и перезагрузить операционную систему.

   ```
   # ignore interfaces with locally administered or null MAC addresses
   # and VMWare virtual interfaces
   ENV{MATCHADDR}=="?[2367abef]:*",  ENV{MATCHADDR}=""
   ```

```
ENV{MATCHADDR}=="00:00:00:00:00:00",    ENV{MATCHADDR}=""
ENV{MATCHADDR}=="00:0c:29:*|00:50:56:*", ENV{MATCHADDR}=""
ENV{MATCHADDR}=="08:00:27:*",           ENV{MATCHADDR}=""
```

4. Подготовить коммутаторы 3COM Switch 1100 (SW-1, SW-2) для моделирования ситуаций 1, 2 задания (при подключении к коммутаторам использовать имя «manager», пароль – «superuser»). Для этого настроить:
   - порты 1-2 коммутаторов SW-1, SW-2 для работы в VLAN-2 без тегирования;
   - порты 3, 4 коммутатора SW-1 для работы в VLAN-3 без тегирования;
   - порты 3, 4 коммутатора SW-2 для работы в VLAN-4 без тегирования;
   - порты 5, 6 коммутаторов SW-1, SW-2 для работы в VLAN-2/3/4 с тегированием.

Выполнение лабораторной работы включает следующие этапы:

1. Подключение и запуск рабочих станций PC-1, PC-2

Цель данного этапа состоит в подготовке моделей IP-сетей, взаимодействие между которыми будет настраиваться в лабораторной работе.

1. Подключить образ виртуальной машины в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 1 сетевой адаптер, для которого указать тип подключения «Сетевой мост» и сетевой адаптер «Realtek 8029» или «3COM» (в зависимости от компьютера).
2. Запустить виртуальную машину.
3. Настроить IP-адрес сетевого интерфейса виртуальной машины в соответствии с вариантом.

4. Подключить компьютер через соответствующий разъем патч-панели «В-0х» на порт коммутатора 1 или 2 (для 1-й ситуации из задания).
5. Повторить шаги 1-5 для 2-й виртуальной машины (на другом компьютере).

2. Подключение и настройка маршрутизации. Ситуация 1

Цель данного этапа состоит в настройке маршрутизации средствами ОС Linux для ситуации, когда на маршрутизаторе установлено 3 сетевых интерфейса. Фильтры на данном этапе не настраиваются. Выход во внешнюю сеть обеспечивается через VirtualBox Host-Only Network (адрес сети 192.168.13x.0/24) и маршрутизируется хост-компьютером.

1. Подключить хост-компьютеры в сеть в соответствии с физической схемой для ситуации 1. Использовать порты 1, 2 коммутаторов.
2. Подключить образ виртуальной машины под управлением ОС Linux в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 3 сетевых адаптера и настроить следующим образом:
   - адаптер 1: тип подключения «Сетевой мост» и сетевой адаптер «Realtek 8029» или «3COM» (в зависимости от компьютера);
   - адаптер 2: тип подключения «Сетевой мост» и сетевой адаптер «D-Link DGE-528»;
   - адаптер 3: тип подключения «Виртуальный адаптер хоста» и сетевой адаптер «VirtualBox Host-Only Ethernet Adapter».
3. Запустить ОС Linux на виртуальной машине, запустить консоль с правами администратора.
4. Выключить сетевые интерфейсы командой
   ```
   ifdown --all
   ```
5. Настроить статические адреса сетевых интерфейсов маршрутизатора. Для этого в файле /etc/network/interfaces для каждого сетевого интерфейса (eth0, eth1, eth2) задать ip-адрес, маску подсети и шлюз:

```
auto eth0
iface eth0 inet static
     address x.x.x.x
     netmask x.x.x.x
     gateway x.x.x.x
```

6. Включить сетевые интерфейсы с заданными настройками:
   `ifup --all`
7. Проверить с помощью команды ifconfig, что требуемые настройки сетевых интерфейсов установлены.
8. Включить перенаправление пакетов с использованием команды:
   `echo 1 > /proc/sys/net/ipv4/ip_forward`
9. С использованием утилиты ping проверить возможность сетевого взаимодействия виртуальных рабочих станций PC-1, PC-2 с сервером и между собой (проверять по ip-адресам!). Проверить возможность доступа с сервера Linux к серверу кафедры Asuserv (для доступа к Asuserv с PC-1, PC-2 потребуется выполнить настройку NAT, см. п. 4).

3. Подключение и настройка маршрутизации. Ситуация 2

Цель данного этапа состоит в настройке маршрутизации средствами ОС Linux для ситуации, когда на маршрутизаторе установлен 1 сетевой интерфейс, а сегменты сети разделены за счет использования VLAN. Фильтры на данном этапе не настраиваются. Выход во внешнюю сеть обеспечивается через сеть 192.168.14x.0/24 и маршрутизируется хост-компьютером.

1. Подключить хост-компьютеры в сеть в соответствии с физической схемой для ситуации 2:
   - ПК-1 (PC-1) включить в VLAN-3;
   - ПК-4 (PC-2) включить в VLAN-4;
   - ПК-2 (Router Linux): сетевой интерфейс 1 («Realtek 8029» или «3COM») через разъем B-0x включить в VLAN-2; сетевой интерфейс 2 (адаптер «D-Link DGE-528») через разъем C-0x

включить в порт коммутатора, настроенный для передачи тегированных кадров 802.1Q (VLAN-2/3/4);

- коммутаторы соединить друг с другом через порты, настроенные для передачи тегированных кадров 802.1Q (VLAN-2/3/4).

2. Подключить образ виртуальной машины под управлением ОС Linux в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 1-й виртуальный сетевой адаптер и настроить его на работу в режиме «Сетевой мост» через физический адаптер «D-Link DGE-528». Другие виртуальные сетевые адаптеры выключить.
3. Запустить ОС Linux на виртуальной машине, запустить консоль с правами администратора.
4. Создать виртуальные интерфейсы для каждой VLAN с использованием команды:

```
//добавление виртуального интерфейса для VLAN-2 на физический интерфейс eth0
vconfig add eth0 2
```

5. Настроить способ формирования имен виртуальных интерфейсов в виде eth0.x, где x – идентификатор VLAN:

```
vconfig set_name_type DEV_PLUS_VID_NO_PAD
```

6. Выключить сетевые интерфейсы командой

```
ifdown --all
```

7. Настроить статические адреса виртуальных интерфейсов маршрутизатора. Для этого в файле /etc/network/interfaces для каждого виртуального интерфейса (eth0.x) задать ip-адрес, маску подсети и шлюз. При этом на сам физический интерфейс адрес не назначать:

```
auto eth0
iface eth0 inet static
     address 0.0.0.0
     netmask 0.0.0.0
auto eth0.2
```

```
iface eth0.2 inet static
     address x.x.x.x
     netmask x.x.x.x
```

8. Включить сетевые интерфейсы с заданными настройками:
   `ifup -all`
9. Проверить с помощью команды ifconfig, что требуемые настройки сетевых интерфейсов установлены.
10. Включить перенаправление пакетов с использованием команды:
    `echo 1 > /proc/sys/net/ipv4/ip_forward`
11. С использованием утилиты ping проверить возможность сетевого взаимодействия виртуальных рабочих станций PC-1, PC-2 с сервером и между собой (проверять по ip-адресам!). Проверить возможность доступа с сервера Linux к серверу кафедры Asuserv (для доступа к Asuserv с PC-1, PC-2 потребуется выполнить настройку NAT, см. п. 4).

4. Настройка правил фильтрации и NAT

Цель данного этапа состоит в настройке простейших правил фильтрации на маршрутизаторе средствами ОС Linux для разграничения доступа между IP-сетями на уровне адресов и сетевых служб (по номерам портов). Средствами стандартного межсетевого экрана Linux настраивается также режим NAT для выхода во внешнюю сеть.

В примерах команд настройки правил фильтрации, приведённых ниже, предполагается, что через интерфейсы eth0, eth1 подключены сегменты локальной сети, а через eth2 организован выход во внешнюю сеть.

1. Настроить трансляцию адресов для интерфейса, обеспечивающего выход во внешнюю сеть (необходимо для корректной маршрутизации во внешней сети):
   `iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE`
2. Проверить возможность доступа с виртуальных рабочих станций к внешнему серверу по адресу 172.19.42.10.

3. Очистить все цепочки таблицы фильтрации и установить правила по умолчанию для цепочек INPUT, FORWARD, обеспечивающие удаление явно не обрабатываемых пакетов (для всех вариантов):

```
iptables -F
iptables -P INPUT DROP  //отбрасывание  всех  входящих
пакетов,
                //предназначенных для самого маршрутизатора
iptables -P FORWARD DROP //отбрасывание  всех  пакетов,
                //требующих  перенаправления  на  другой
                интерфейс
```

4. Разрешить прохождение ICMP-трафика между сетями для обеспечения работы утилит ping и traceroute (для всех вариантов):

```
//для проверки связи с самим маршрутизатором
iptables -A INPUT -i eth0 -p icmp -j ACCEPT
iptables -A INPUT -i eth1 -p icmp -j ACCEPT
//для проверки связи между подсетями
iptables -A FORWARD -i eth0 -p icmp -j ACCEPT
iptables -A FORWARD -i eth1 -p icmp -j ACCEPT
```

5. Настроить правила доступа к ресурсам сети в соответствии с вариантом, например:

```
//разрешаем подключаться к Asuserv на порт TCP-110 (протокол
//доступа к почтовому ящику POP3) из сети, подключенной на
//интерфейс eth0
iptables -A FORWARD -i eth0 -d 172.19.42.10 -p tcp --dport
110 -j ACCEPT
//разрешаем прохождение трафика в обратном направлении
//для установленных соединений
iptables -A FORWARD -m state --state ESTABLISHED, RELATED -j
ACCEPT
//разрешаем подключаться клиентам из сети 10.1.1.0/24 к
//рабочей станции 10.1.2.1 по любому протоколу
iptables -A FORWARD -s 10.1.1.0/24 -d 10.1.2.1 -p ANY -j
ACCEPT
```

6. Проверить взаимодействие узлов сети в соответствии с заданием, предусмотренным вариантом. Проверить невозможность других видов взаимодействия между сетями. Проверить невозможность доступа к маршрутизатору из внешней сети.

5. Сохранение сценария настройки маршрутизатора

Цель данного этапа состоит в создании файла сценария, определяющего сетевые настройки ОС Linux для работы в качестве маршрутизатора с правилами доступа, определенными вариантом. Сценарий должен обеспечивать конфигурацию экранирующего маршрутизатора, заданную вариантом, после перезагрузки операционной системы.

Содержание отчета

1. Титульный лист.
2. Цель работы, задание.
3. Схемы ЛВС с указанием IP-адресов по варианту.
4. Маршрутизация интерфейсов.
   4.1. Содержание файла «etc/network/interfaces» для ситуации 1.
   4.2. Результаты проверки достижимости из сети 1 для сети 2 и внешней сети.
5. Маршрутизация VLAN.
   5.1. Содержание файла «etc/network/interfaces» для ситуации 2.
   5.2. Результаты проверки достижимости из сети 1 для сети 2 и внешней сети.
6. Правила фильтрации.
   6.1. Результаты проверки доступности сетевых служб, предусмотренных вариантом.
   6.2. Результаты проверки невозможности сетевого доступа, не предусмотренного вариантом.
7. Сценарий запуска (файл сценария для п.5).

Контрольные вопросы

1. В чем заключается основной принцип работы устройств: концентратор, хаб, повторитель?
2. Укажите недостатки структурирования сети на основе концентраторов?
3. Используя схему сети, приведите пример передачи кадров от узла отправителя к узлу получателю?
4. Каким из указаных в проекте устройств необходимо наличие физических адресов (MAC)?
5. Какую логическую топологию выстраивает сетевой концентратор? Какую физическую топологию сети используют при применении концентраторов?
6. В каком режиме разделения общей среды передачи данных функционирует концентратор?
7. Опишите сценарий возникновения и обработки коллизии в сети передачи данных, основанной на применении концентраторов?
8. Перечислите наиболее распространненные типы физических сетевых разъемов и номенклатуру соединительных кабелей, используемых на концентраторах?
9. Укажите назначение Uplink-порта на Ethernet-концентраторе и назначение тумблера Normal/Uplink?

Практическое занятие 7 Практическое изучение WIFI роутера

Цель: Изучить основные способы подключения устройств к беспроводной сети

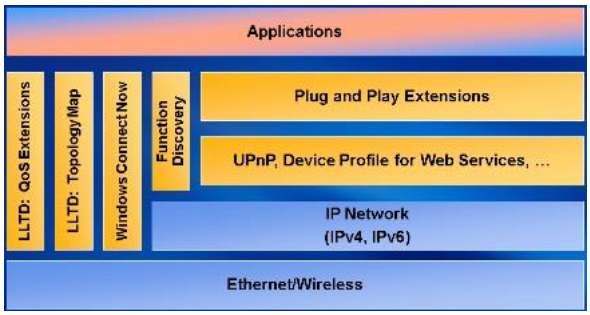
Краткие теоретические сведения

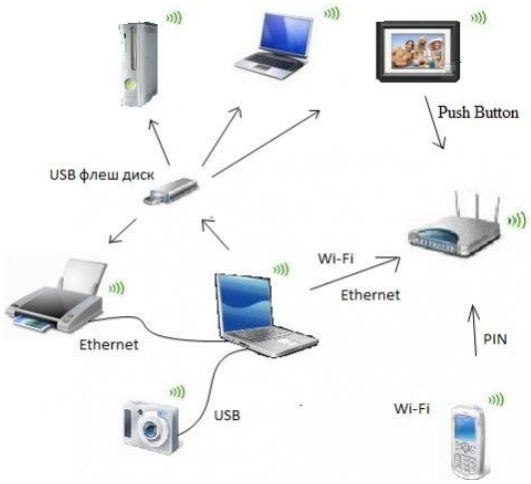
Использование различных устройств с WiFi стремительно врывается в нашу жизнь, сейчас WiFi оснащена не только сложная техника вроде ноутбуков и коммуникаторов, но и даже такие простые гаджеты, как фоторамка. Настраивать их для подключения к беспроводным сетям становится непростым делом, поэтому не удивительно появление технологий, позволяющих значительно упростить процедуру настройки.

Создание новой беспроводной сети начинается непосредственно с конфигурации точки доступа (беспроводного маршрутизатора) подключения к ней компьютеров и другого беспроводного оборудования.

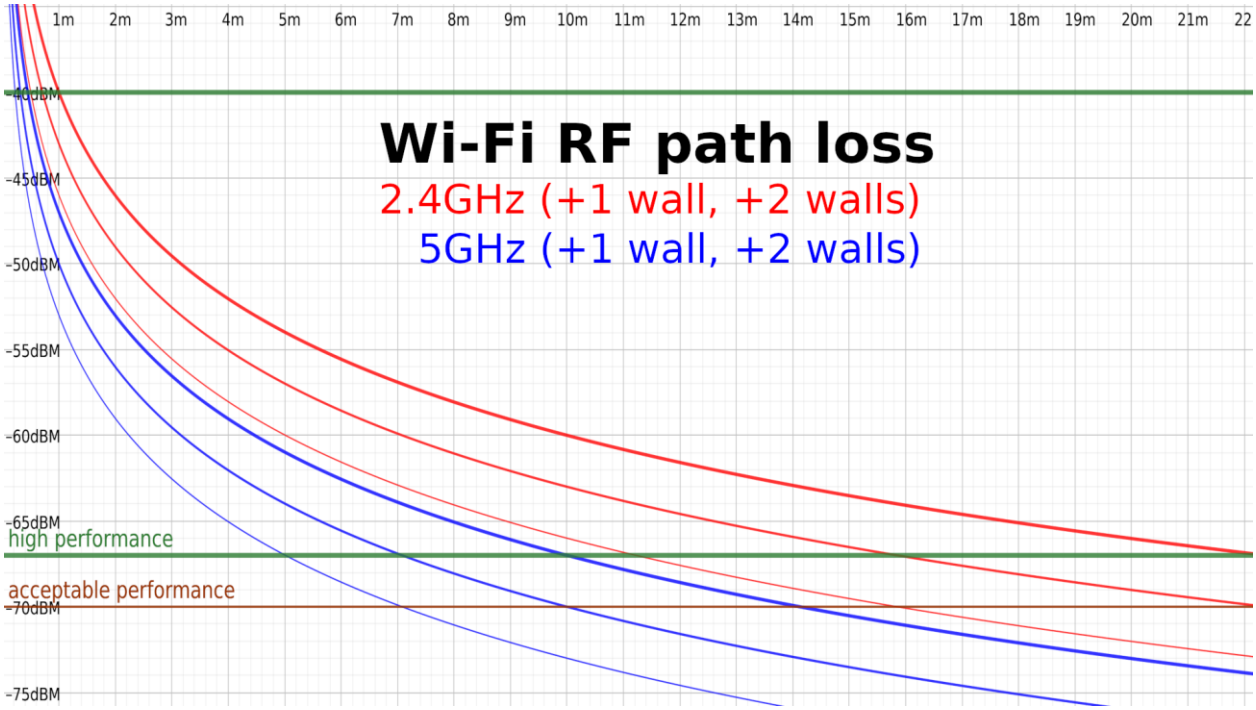
Традиционный способ настройки с точки зрения обычного пользователя выглядит очень сложным: нужно произвести непростые действия с подключением к точке доступа для первой настройки, нужно создать вручную имя беспроводной сети, указать сложный и трудно воспроизводимый ключ безопасности. И весь этот процесс настройки требует, чтобы пользователь имели базовые знания о WiFi. Ему будет куда проще просто нажать на кнопку или ввести ПИН, чтобы все само настроилось и подключилось. **Wireless Protected Setup** (**WiFi Protected Setup - WPS**) как раз придумана, чтобы быстро, максимально просто и безопасно настраивать сетевые устройства и компьютеры. К примеру, один из способов подключения игровой консоли с WiFi к беспроводной сети: достаточно нажать на кнопки WPS на беспроводном маршрутизаторе и на игровой консоли.

WiFi Protected Setup сейчас является стандартом для простого и безопасного создания беспроводной сети. В терминологии Microsoft это **Windows Connect Now** (**WCN**).





Среда конфигурации беспроводных устройств может быть разной: **Ethernet**, **WiFi**, **USB кабель** или **USB флеш** диск. Конфигурация беспроводного устройства может осуществляться через ПИН-код (PIN - Personal Identification Number), через нажатия конфигурационных кнопок (PBC - Push Button Configuration,
На качекство связи сильное влияние оказывет расстояние и наличие препятсвий между роутером и абонентским устройством



Каждое удвоение расстояния приводит к падению сигнала на 6 дБм, и мы чётко видим это, изучая толстую красную кривую для 2,4 ГГц: на 1 м сигнал -40 дБм, на 2 м это -46 дБм, на 4 м это -52 дБм.

Стены и иные препятствия – включая, но не ограничиваясь человеческие тела, шкафы, мебель, бытовую технику – ещё больше ослабят сигнал. Простое практическое правило — -3 дБм для каждой стены или другого значительного препятствия. Более тонкие линии того же цвета на графике показывают падение сигнала на тех же расстояниях при добавлении одной-двух стен (или других препятствий).

В идеале хочется иметь уровень сигнала не менее -67 дБм, однако не нужно беспокоиться о том, чтобы повышать его сильно выше этой отметки – обычно разницы в скорости между мощным -40 дБм и хилым -65 дБм нет, пусть они и находятся далеко друг от друга на графике. На работу WiFi влияет гораздо больше факторов, чем просто

мощность сигнала; как только вы превысите минимум, уже неважно, насколько именно вы его превысили.

На самом деле, слишком мощный сигнал может оказаться такой же проблемой, как и слишком хилый — многие пользователи на форумах жалуются на низкую скорость, пока какой-нибудь смышлёный человек не спросит их: вы что, разместили устройство прямо рядом с точкой доступа? Отодвиньте его на метр-два и попробуйте снова. И, конечно, проблема исчезает.

Правило 1: не больше двух комнат и двух стен

Наше первое правило для размещения точки доступа (ТД) — не больше двух комнат и двух стен между ТД и устройствами. Правило довольно расплывчатое, поскольку комнаты бывают разного размера и формы, а у разных домов разный состав стен – но это неплохая точка отсчёта, и она хорошо послужит вам в домах типичного размера и квартирах с достаточно современными межкомнатными стенами из гипсокартона.

Типичный размер, по крайней мере, в большей части США, означает спальни длиной 3-4 метра и гостиные длиной 5-6 метров по одной из стен. Если взять девять метров как среднюю дистанцию, покрывающую «две комнаты», и добавим две внутренних стены, по -3 дБм на каждую, наша кривая потерь радиоволн показывает, что сигналы на 2,4 ГГц будут прекрасно себя чувствовать с показателем -65 дБм. С 5 ГГц ситуация не такая хорошая – если нам потребуются все 9 метров и 2 стены, то мы опустимся до -72 дБм. Этого достаточно для установления связи, но и только. В реальной жизни устройство с сигналом на -72 дБм на 5 ГГц увидит примерно ту же пропускную способность, что и устройство на -65 дБм на 2,4 ГГц – однако формально более медленная связь на 2,4 ГГц окажется более стабильной и покажет меньшие задержки.

Конечно, всё это при условии, что единственными нашими проблемами будут расстояние и ослабление сигнала. Пользователи в сельской местности и в домах с большими участками уже наверняка заметили эту разницу и уяснили себе практическое правило «2,4 ГГц – это круто, а вот 5 ГГц – это полный отстой». У городских жителей или владельцев домов, стоящих на участке размером с почтовую марку, имеется совершенно другой опыт, который мы учтём во 2-м правиле.

Правило 2: слишком большая мощность передачи – это плохо

Плюсом сигнала на 2,4 ГГц служат дальнобойность и эффективное проникновение сквозь препятствия. Минусом сигнала на 2,4 ГГц служат… дальнобойность и эффективное проникновение сквозь препятствия.

Если два WiFi устройства на расстоянии «слышимости» друг от друга передают на одной и той же частоте одновременно, у них ничего не выходит: у устройств, для которых они передают сигнал, нет возможности разобраться в этом и понять, какой из сигналов предназначается для них. Вопреки распространённому мнению, тут совершенно неважно, находится ли устройство в вашей сети или нет – название и пароль WiFi не имеют никакого значения.

Чтобы по большей части избежать такой проблемы, любое WiFi устройство перед передачей должно сначала прослушать эфир – и если любое другое устройство уже передаёт на этом диапазоне частот, то наше должно заткнуться и подождать конца передачи. Это не устраняет проблему полностью; если два устройства решать передавать одновременно, они «столкнутся» – и каждому нужно будет выбрать случайный промежуток времени, которое они проведут в ожидании перед тем, как попытаться снова что-то передавать. Устройство, выбравшее меньшее случайное число, начинает первым –

если они не выберут одинаковое случайное число, или какое-то другое устройство не заметит передышку в эфире и не решит передавать сигнал, опередив обоих.

Это называется «затором», и для большинства современных пользователей WiFi это такая же большая проблема, как и ослабление сигнала. Чем больше у вас устройств, тем более загружена сеть. Каждое из ваших устройств может столкнуться с другим, и каждому приходится уважать правила пользования эфиром.

Если ваш роутер или ТД поддерживают такой вариант, то уменьшение мощности исходящего сигнала может наоборот, улучшить быстродействие и роуминг – особенно если у вас меш-набор или другая похожая схема. Сети 5 ГГц обычно не нужно так ослаблять, поскольку сигнал в том спектре и так достаточно быстро ослабляется, однако для 2 ГГц такой вариант может творить чудеса.

Последнее замечание для любителей «дальнобойных» ТД – такая ТД может и правда выдать сигнал сильнее обычной, и добить на большее расстояние. Однако она не может заставить ваш телефон или ноутбук усилить сигнал в ответ. При таком дисбалансе отдельные части веб-страницы могут загружаться быстро, однако в целом соединение будет казаться нестабильным, поскольку ваш ноутбук или телефон будет сначала с трудом загружать десятки или сотни отдельных запросов по HTTP/HTTPS, необходимых для загрузки каждой их веб-страниц.

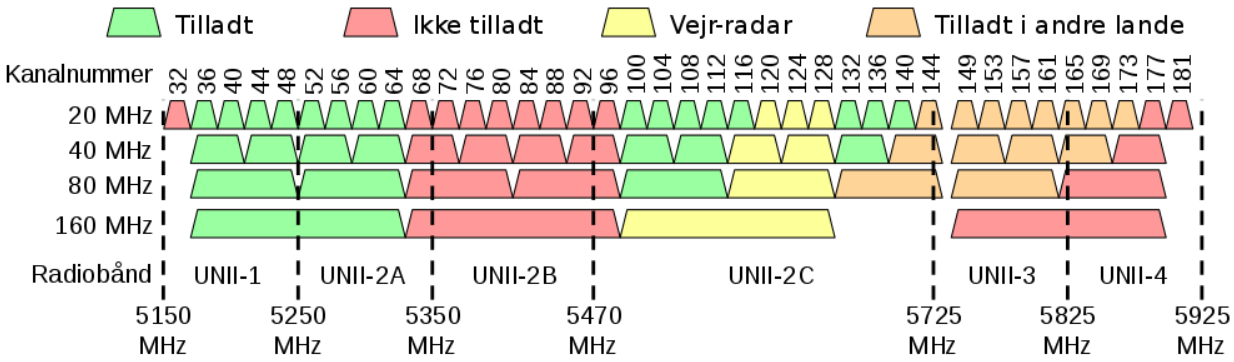
Правило 3: используйте спектр с умом

Во втором правиле мы упомянули, что все устройства на одном канале соревнуются за эфирное время, вне зависимости от того, к какой сети они принадлежат. У большинства людей отношения с соседями не настолько хорошие, чтобы можно было убедить их понизить мощность передачи – даже если их роутер поддерживает такую функцию – но вы можете понять, какие каналы используют соседние сети, и избегать их.

С 5 ГГц такой проблемы обычно не возникает, но на 2,4 ГГц это может довольно сильно влиять. Поэтому мы рекомендуем большинству людей избегать стандарта 2,4 ГГц. А где избегать его не получается, используйте приложение типа inSSIDer, чтобы периодически изучать своё радиоволновое окружение, и пытаться избегать использования самого загруженного спектра в районе вашего дома.

Однако это, к сожалению, может быть сложнее, чем кажется на первый взгляд. Неважно, сколько SSID вы увидите на определённом канале – важно, сколько эфирного времени они реально используют, а это нельзя подсчитать ни исходя из количества SSID, ни исходя из чистой мощности сигнала у видимых SSID. InSSIDer позволяет вам сделать ещё один шаг и изучить реальную утилизацию эфирного времени в каждом канале.

В сетях 5 ГГц загрузка каналов представляет собой гораздо меньшую проблему, поскольку уменьшение дальности действия и проницаемости сигнала означает наличие меньшего количества устройств, с которыми приходится соревноваться. Часто можно услышать заявления о том, что у этого стандарта больше каналов для работы, но на практике это не так, если вы не занимаетесь настройкой WiFi на территории вашего предприятия, где нет конкурирующих сетей. Домашние роутеры на 5 ГГц обычно настраивают на ширину канала 40 или 80 МГц, что означает, что непересекающихся каналов реально всего два – нижний, состоящий из каналов 36-64 шириной 5 МГц, и верхний, на каналах 149-165.



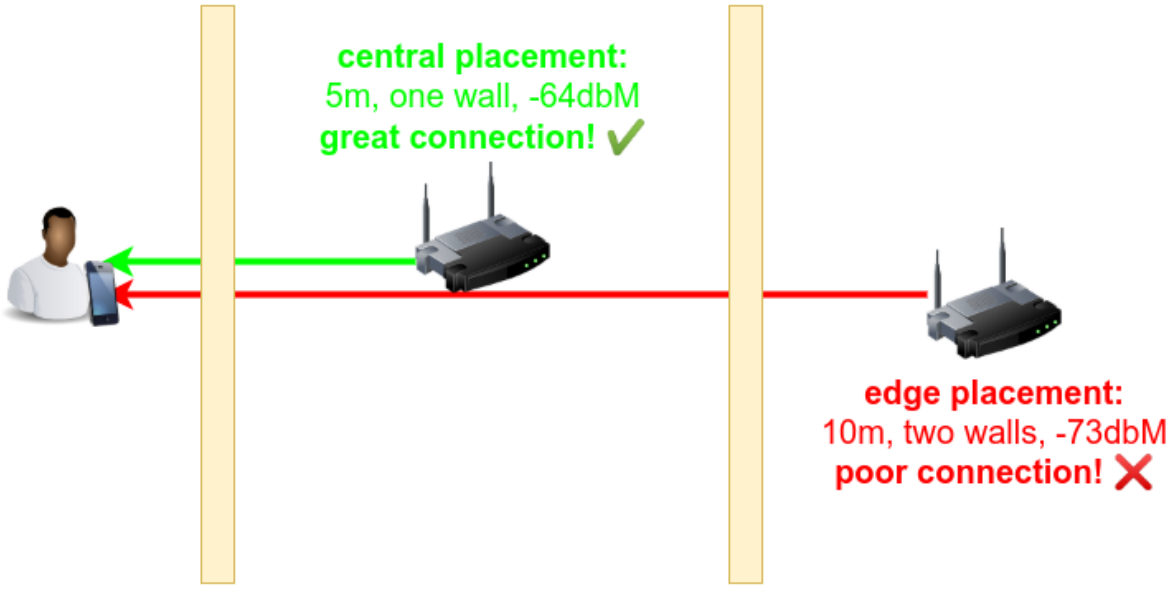
Каждая сеть 5 ГГц шириной 40 МГц занимает чуть больше 8 реальных каналов шириной 5 МГц. Каждый пенёк тут символизирует четыре канала шириной 5 МГц.

В комментариях наверняка стоит ожидать дискуссии по поводу данных утверждений. Технически, можно уместить четыре сети шириной 40 МГц или две сети шириной 80 МГц на нижней части полосы 5 ГГц. На практике же потребительское оборудование работает через пень-колоду с накладывающимися каналами (к примеру, с полосой 80 МГц центрированной на канале 48 или 52), из-за чего такой эффективности спектра в реальных домашних условиях достичь сложно или практически невозможно.

Между двумя стандартными потребительскими полосами (в США) есть ещё два канала с динамической частотой DFS (Dynamic Frequency Spectrum), однако их нужно делить с такими устройствами, как коммерческие и военные радары. Многие потребительские устройства отказываются даже пытаться использовать DFS. И даже если у вас есть роутер или ТД, согласные на использование DFS, они должны подчиняться строжайшим требованиям, чтобы не давать помех никаким радарам. Пользователи «в глуши» могут прекрасно использовать DFS – однако у них и проблем с загрузкой каналов, скорее всего, не будет.

Если вы живёте рядом с аэропортом, военной базой или портом, DFS вам, скорее всего, не подойдёт – а если вы живёте за пределами США, разрешённые у вас частоты могут отличаться от того, что описано тут (как DFS, так и остальные), в зависимости от местных законов.

Правило 4: лучший вариант – центральное размещение

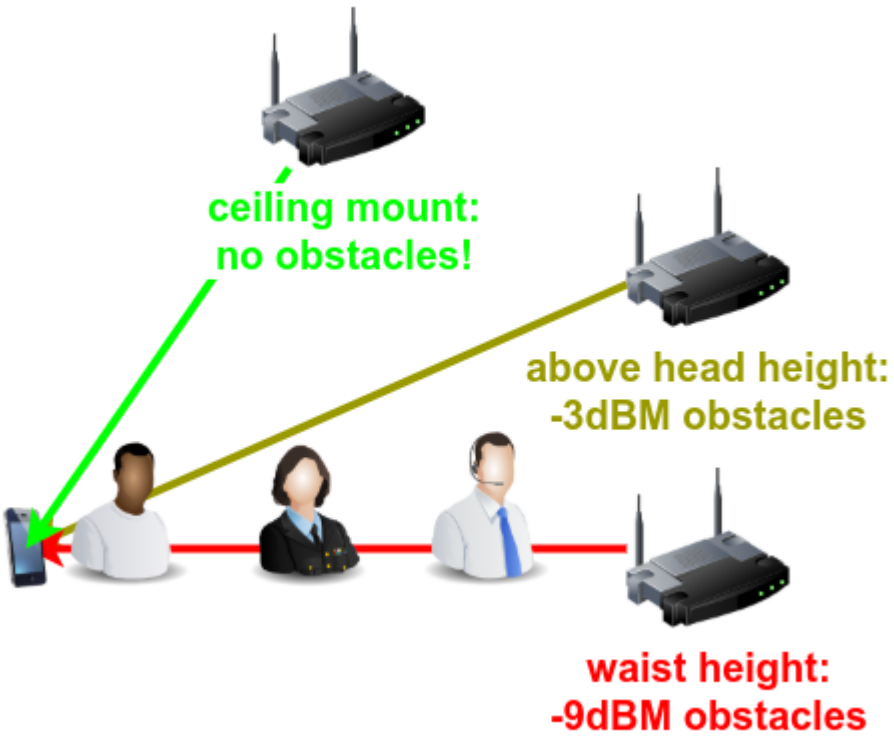


Разница между «роутером с краю дома» и «ТД посередине» может оказаться критической

Возвращаясь к ослаблению сигнала, отметим, что идеальное место для расположения ТД WiFi – это центр пространства, которое ему нужно покрывать. Если ваше жилище имеет длину по одной из сторон 30 м, то роутеру, расположенному посередине, нужно будет покрыть только 15 м в каждую сторону, а роутеру с краю (где установщики от провайдера любят заканчивать коаксиальный кабель или линию DSL) придётся покрывать 30 м.

Это же справедливо и для меньших помещений с большим количеством ТД. Помните, сигналы WiFi быстро затухают. Шести метров – длины достаточно большой гостиной – может хватить для того, чтобы сигнал на 5 ГГц, ослабнув, опустился ниже оптимального уровня, если добавить туда пару препятствий типа мебели или людей. Что приводит нас к следующему правилу…

Правило 5: высота – выше человеческого роста



Технически наилучшим расположением будет место у потолка – но если это слишком, то поместите ТД хотя бы наверху книжных полок.

Чем выше вы сможете закрепить ТД, тем лучше. Человеческое тело ослабляет сигнал примерно на столько же, на сколько и внутренняя стена – это одна из причин, по которой WiFi в вашем доме значительно ухудшается, когда на вечеринку пришло много друзей.

Разместив ТД – или роутер – выше человеческого роста, можно избежать необходимости передавать радиоволны сквозь все эти надоедливые и ослабляющие сигнал мешки с мясом. Также сигнал избегает большей части мебели и бытовой техники – диванов, столов, духовок и шкафов.

Самым идеальным вариантом будет размещение ТД на потолке в геометрическом центре комнаты. Если это невозможно, не беспокойтесь – почти так же хорошо будет поставить её наверх шкафа, особенно, если вам нужно, чтобы эта ТД обслуживала как ту комнату, где она стоит, так и комнату с другой стороны стены.

Правило 6: делите расстояния пополам

Допустим, некоторые ваши устройства расположены слишком далеко от ближайшей точки доступа для того, чтобы получить хороший сигнал. Вам повезло купить расширяемую систему, или у вас осталась одна ТД из меш-кита. Где её поместить?

Мы наблюдали замешательство людей в подобной ситуации, размышлявших о том, стоит ли поместить дополнительную ТД поближе к первой (с которой она берёт данные) или поближе к самым дальним устройствам (к которым она должна передавать данные). Ответ обычно такой: ни то, ни другое. Размещайте вашу ТД прямо посередине между ближайшей ТД и самым дальним клиентом, которого она должна обслуживать.

Суть в том, что вы пытаетесь сохранить эфирное время, организуя наилучшее соединение из возможных между дальними устройствами и новой ТД, и между новой ТД и ближайшей к ней. Обычно не стоит отдавать предпочтение одной из сторон. Однако не забывайте правило 1: две стены, две комнаты. Если нельзя разбить расстояние между самыми дальними клиентами и основной ТД, не нарушая первого правила, тогда размещайте новую ТД так далеко, как это позволяет первое правило.

Если вам это кажется слишком простым и логичным, не волнуйтесь: есть ещё один момент «только если не», который необходимо учитывать. У некоторых меш-наборов, например, Netgear's Orbi RBK-50/RBK-53 или Plume's Superpods, связь между ТД имеет очень высокую пропускную способность и работает по схеме 4x4. Поскольку это соединение работает гораздо быстрее 2x2 или 3x3, доступных клиентам, возможно, стоит уменьшить качество сигнала связи между этими ТД, так, чтобы их пропускная способность была ближе к той, которую могут позволить себе лучшие из ваших клиентов.

Если ваш меш-набор предлагает очень быстрое соединение между ТД, и вам никак не удаётся добавить к схеме дополнительных ТД, то вам, возможно, лучше будет поместить последнюю ТД ближе к клиентам, чем к предыдущей ТД. Однако тут придётся поэкспериментировать и изучить результаты.

Прикольная штука — WiFi, не правда ли?

Правило 7: обходите препятствия

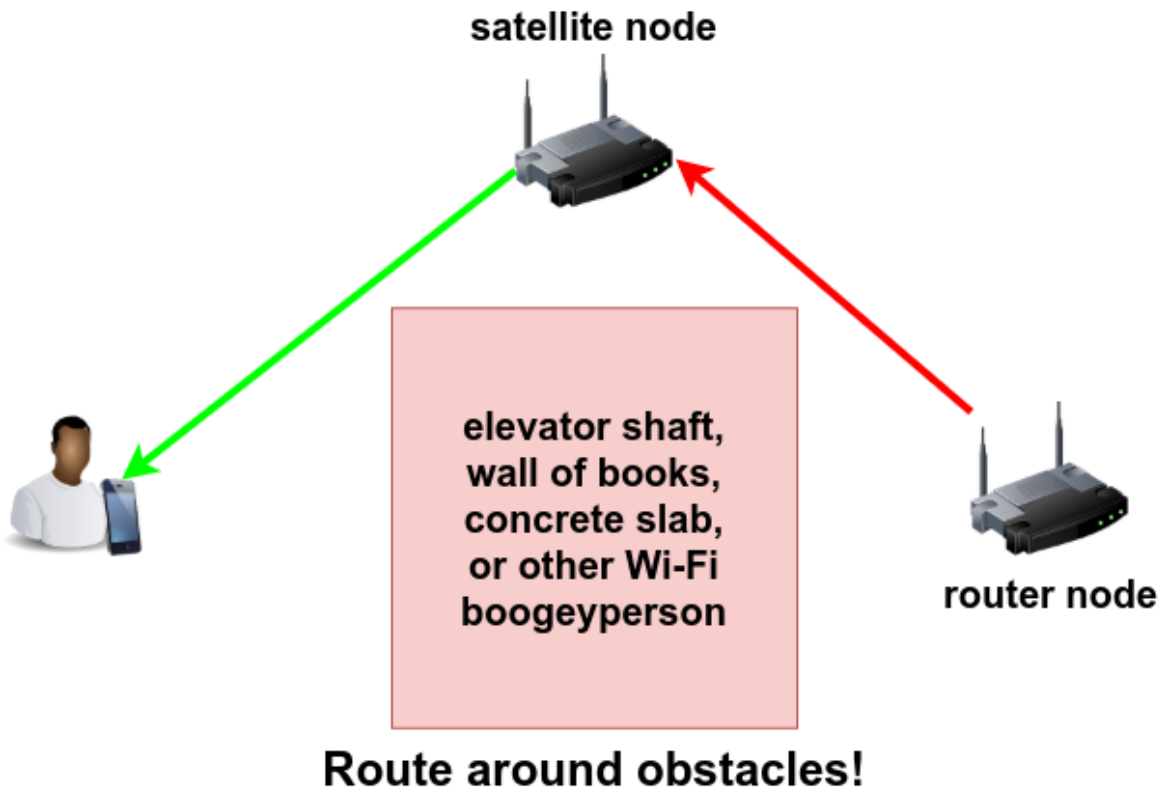
Плотно набитый книжный шкаф – серьёзное препятствие для радиоволн. Он стоит пары обычных стен даже при перпендикулярном проникновении. А уж в длину его пересекать вообще бесполезно.

Если вам досталось особо сложное помещение, в нём могут оказаться такие места, куда сигнал просто не сможет пройти. В нашем испытательном доме была бетонная плита и несколько метров плотной земли, закрывавшие линию видимости между роутером и

подвалом. Мы встречали небольшие предприятия, точно так же обеспокоенные тем, что в одной части помещения WiFi работал хорошо, а в другой его не было – и в итоге оказывалось, что на пути сигнала стоит, например, книжный шкаф, забитый книгами, и расположенный вдоль коридора, из-за чего на пути сигнала оказывались несколько метров ослабляющей его переработанной древесины.

В каждом из случаев решением будет создание обходного пути вокруг препятствия при помощи нескольких точек доступа. Если у вас есть меш-набор WiFi, используйте его так, чтобы сигнал обходил препятствия. С одной стороны препятствия поместите ТД на линии прямой видимости с основной, причём так, чтобы её было видно с другой стороны препятствия, и сигналу не нужно было идти насквозь.

С достаточным количеством ТД и тщательным их размещением вы, возможно, сможете справиться даже со стенами, сделанными из дранки и металлической сетки, как строили в США в начале XX века. Мы видели, как люди успешно размещали ТД в прямой видимости друг друга через дверные проёмы и коридоры, когда для проникновения сквозь стены проще было бы использовать перфоратор.



Если слишком большое количество препятствий не даёт вам обойти их сбоку, сверху или снизу – смотрите правило 8.

Правило 8: всё дело в связи между точками доступа

Большинство потребителей выбирают чистые меш-наборы WiFi, поскольку это удобно – не нужно вести провода, просто подключаете кучу точек доступа, и пусть они там осуществляют свою магию между собой самостоятельно, без шума и пыли.

Звучит удобно, но на самом деле это самое плохое решение. Помните, мы говорили о правилах 2 и 3? Эти проблемы есть и здесь. Если вашему устройству нужно общаться с одной ТД, которой нужно передавать данные в другую ТД, то вы уже занимаете чуть более, чем в два раза больше эфирного времени.

Ладно, на самом деле не так всё плохо – вы удваиваете использование эфирного времени, если ваш клиент находится там же, где и вспомогательная ТД. А поскольку вы последовали правилу 6 – поделили расстояния пополам – это значит, что качество связи у основной ТД с клиентом гораздо лучше той, которую организовал был клиент, подключаясь к основной ТД напрямую. Так что даже в самом худшем случае – когда вспомогательная ТД беседует с клиентом на том же канале, на котором она беседует с основной ТД – у них получится передавать данные, потребляя меньше эфирного времени, чем если бы один клиент работал с гораздо более длинным и менее качественным соединением.

Однако гораздо лучше будет полностью избежать этой проблемы, если ваши ТД будут общаться друг с другом на другой частоте. Двухполосные ТД могут делать это, общаясь с клиентами в диапазоне 2,4 ГГц, а между собой – на 5 ГГц, или наоборот. В реальном мире упрямые клиенты (и пользователи) часто хотят соединяться не так оптимально, в итоге получается, что клиенты есть и на 2,4 ГГц, и на 5 ГГц, поэтому «чистого» канала для внутренней связи не остаётся.

Особо умные наборы, такие, как Eero, могут избегать такой ситуации благодаря динамической маршрутизации внутренней связи, минимизируя заторы путём передачи в диапазоне, отличной от того, в котором они ведут приём, даже когда диапазоны меняются. Самые продвинутые трёхполосные наборы типа Orbi RBK-50/53 или Plume Superpods могут избежать такой проблемы, используя второй передатчик на 5 ГГц. Это позволяет им соединяться с клиентами либо по 2,4 ГГц, либо по 5 ГГц, оставляя себе незанятый диапазон на 5 ГГц. У Orbi передатчик для внутренней связи фиксированный и выделенный. Plume принимает решения по использованию частот в зависимости от того, какой вариант его облачный оптимизатор считает наилучшим в конкретном окружении).

Лучший вариант – вообще не использовать WiFi для внутренней связи. Если можно проложить Ethernet-кабель, надо так и сделать. Он не только быстрее WiFi, он ещё и не страдает от проблем с загруженностью каналов. При высокой загрузке сети дешёвые проводные ТД типа Ubiquiti UAP-AC-Lites или TP-Link EAP-225v3s уделывают всухую даже самые дорогие меш-наборы, если последние ограничены внутренней связью по WiFi. Проводная внутренняя связь также решает проблему непрозрачных для радиоволн препятствий – если сквозь него нельзя пробить сигнал или обойти его, то протянутый сквозь него кабель творит чудеса!

Пользователям, которым не удалось реализовать ни меш-наборы с WiFi, ни протянуть кабели Ethernet, стоит рассмотреть современное оборудование для передачи сигналов по линиям электропередач. Результаты могут быть совершенно разные, и зависеть от качестве проводки в доме и даже от типа подсоединённых бытовых приборов, но в большинстве случаев достаточно надёжным будет оборудование серий AV2 (AV1000 и выше) или g.hn, задержки передачи будут достаточно низкими, сравнимыми с Ethernet. Пропускная способность жёстко ограничена – в реальном мире стоит ожидать не более 40-80 Мб/с для домашних условий. Если вы в интернете занимаетесь только играми или просмотром веб-страниц, тогда передача данных по электропроводке может стать гораздо лучшим решением, чем WiFi.

Пойдя по этому пути, обязательно читайте инструкцию и принимайте меры для шифрования связи. В первый раз мы при проверке такого оборудования случайно построили мост с соседом, и перенастроили его роутер – он был почти такой же модели, что и наш, а пароль на нём стоял по умолчанию. «Здравствуйте, я взломал ваш роутер, прошу прощения» – плохой способ познакомиться, не рекомендуем.

Правило 9: обычно проблемы не в пропускной способности, а в задержках

У пропускной способности хорошо то, что это один красивый яркий номер, который легко получить, соединившись с сайтом для проверки скорости или используя инструмент типа iperf3 для связи с локальным сервером.

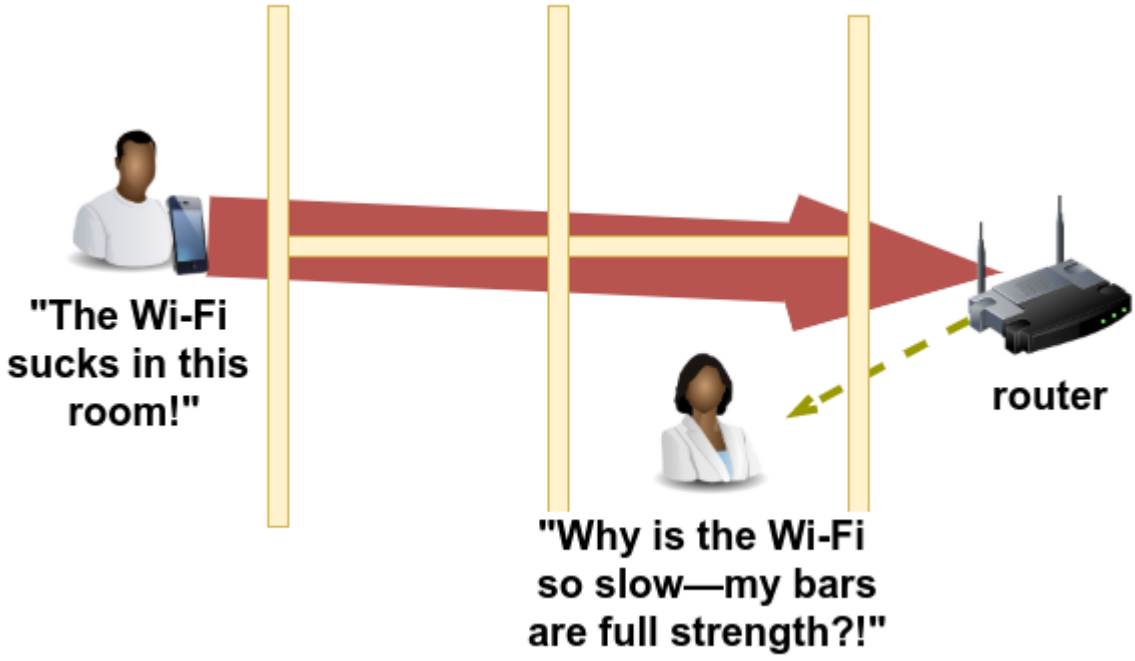
Плохо у пропускной способности то, что это ужасный способ измерять как впечатление пользователя от сети, так и то, как сеть WiFi ведёт себя под реальной нагрузкой. Большинство людей расстраивает их WiFi-сеть либо при просмотре веб-страниц, либо в играх – а не тогда, когда они скачивают большой файл. В обоих случаях проблема не в том, «сколько мегабит в секунду может выдержать эта труба» – а в том, «сколько миллисекунд уходит на завершение конкретного действия».

И хотя можно увидеть ухудшение качества работы загруженной сети по уменьшающимся цифрам «скорости» скачивания, это более сложный, запутанный, и не связанный с реальностью способ, по сравнению с изучением задержек приложений. Задержки являются функцией как от просто скорости, так и от эффективности обработки сетью трафика и эфирного времени.

При проверке WiFi-сетей наша любимая метрика – это задержка в приложениях, которую мы симулируем загрузкой достаточно сложной веб-страницы. Что ещё важнее, нужно измерять загрузку страницы параллельно со всей остальной активностью в сети. Помните описание заторов в правилах 2 и 3 – «очень быстрая» сеть с одним активным устройством может превратиться в кошмарного тормоза со многими устройствами, или, во многих случаях, с одним плохо подключённым устройством, что приводит нас к последнему правилу.

Вывод 9-го правила такой, что рекламируемая скорость, идущая после букв AC в модели, — это фигня. Нужно доверять тщательным, технически компетентным обзорщикам, а не рейтингу скорости производителя на коробке.

Правило 10: скорость вашей сети WiFi ограничена скоростью самого медленного из подключённых устройств



Одно устройство с хреновым подключением может убить качество связи для всей сети и всех подсоединённых устройств

К сожалению, один человек, пытающийся посмотреть ролик с YouTube в «спальне с хреновым приёмом», мучается не только сам – его проблемы настигают и других. Телефону, находящемуся в одной комнате с ТД, нужно всего лишь около 2,5% имеющегося эфирного времени для потоковой передачи ролика в качестве 1080P на скорости 5 Мб/с. Но телефон «в плохой спальне», мучающийся из-за буферизации и медленной связи, может забрать себе 100% эфирного времени сети, и не суметь при этом посмотреть то же самое видео.

Конечно, потоковое видео очень сильно занимает входящий канал, и роутеры или ТД обычно отказываются вести передачу 100% времени. ТД, которой нужно передать большое количество данных, обычно оставит немного эфирного времени для других устройств и запроса собственных данных, а потом оно разобьёт время на скачивание между близлежащим устройством и «плохой спальней», чтобы попытаться выполнить оба запроса. Но это всё равно увеличивает время ожидания окна от этих устройств на сотни миллисекунд, и им всё равно приходится соревноваться друг с другом при открытии этого окна.

Ситуация ухудшается, если пользователь в «плохой спальне» пытается загрузить видео, отправить емейл или запостить большую фотку в соцсеть. Роутер пытается оставить часть эфирного времени другим устройствам – однако на телефон пользователя эти ограничения не действуют, и он с радостью сожрёт всё доступное эфирное время. Что хуже, телефон не представляет, сколько данных запросили другие пользователи в те краткие моменты, когда у них было окно для запросов. Роутер знает, сколько данных нужно доставить каждому из клиентов, поэтому он может размещать время для скачивания данных сообразно – но всё, что знает телефон, это то, что ему нужно закачать свои данные, поэтому пока он этим занимается, страдают все остальные. Поэтому даже если из всей этой мудрости вам нужно оставить только одно правило, пусть это будет правило 10.

1. Контрольные вопросы

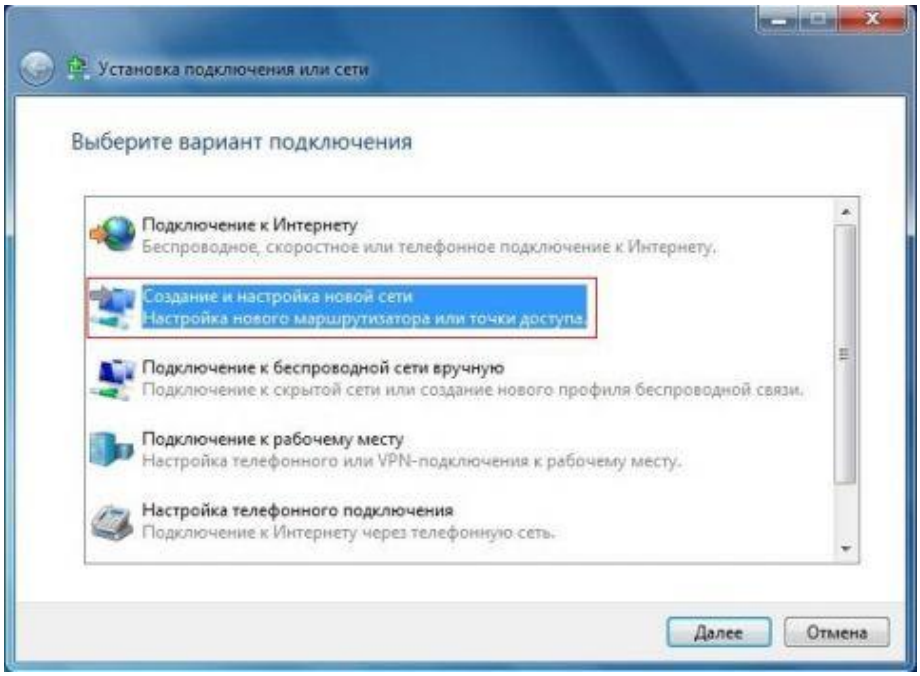
1. Перечислите набор стандартов IEEE для организации беспроводной связи WiFi?
2. Укажите рабочие частотные диапазоны стандартов 802.11a, 802.11b, 802.11g и 802.11n?
3. Объясните назначение и функции следующего оборудования: беспроводная сетевая
4. карта (WIC), точка доступа (AP), беспроводной маршрутизатор (WR)?
5. В чем заключается различие между топологиями беспроводной сети Ad-hoc и Hotspot?
6. Укажите способы аутентификации беспроводного клиента? Приведите примеры практического использования указанных способов?
7. Приведите способы защиты доступа к информационной структуре беспроводной сети?
8. Дайте определение понятиям: SSID, WPA, WEP, WLAN, OFDM, DSSS?
9. Укажите методы первоначального конфигурирования класса беспроводного оборудования?
10. Объясните необходимость назначения IP-адреса беспроводной точке доступа?

Лабораторная работа 7 Установка и настройка и конфигурирование WIFI роутера

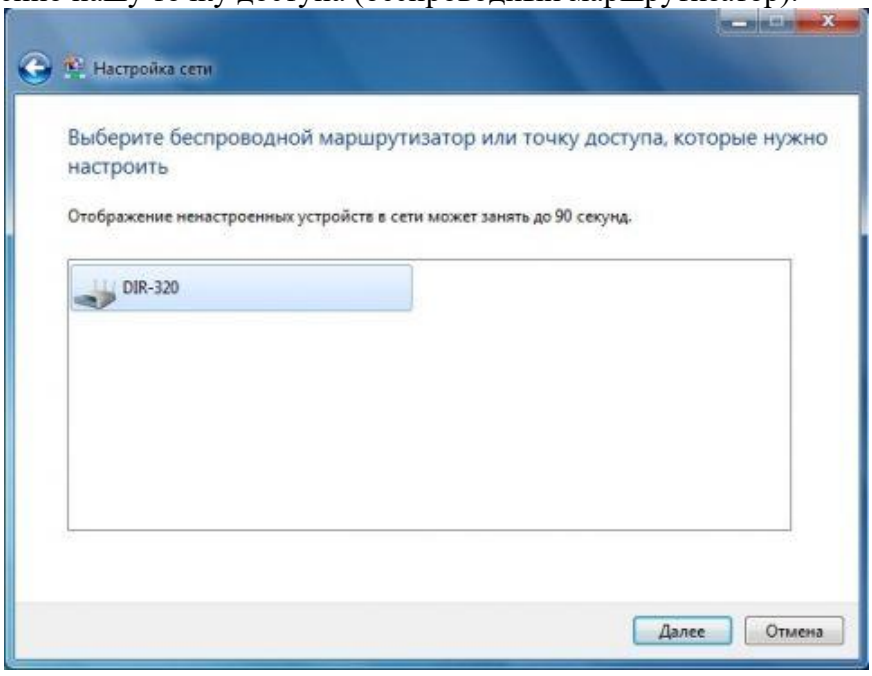
2. Порядок выполнения работы:

1.1 Настройка беспроводного маршрутизатора

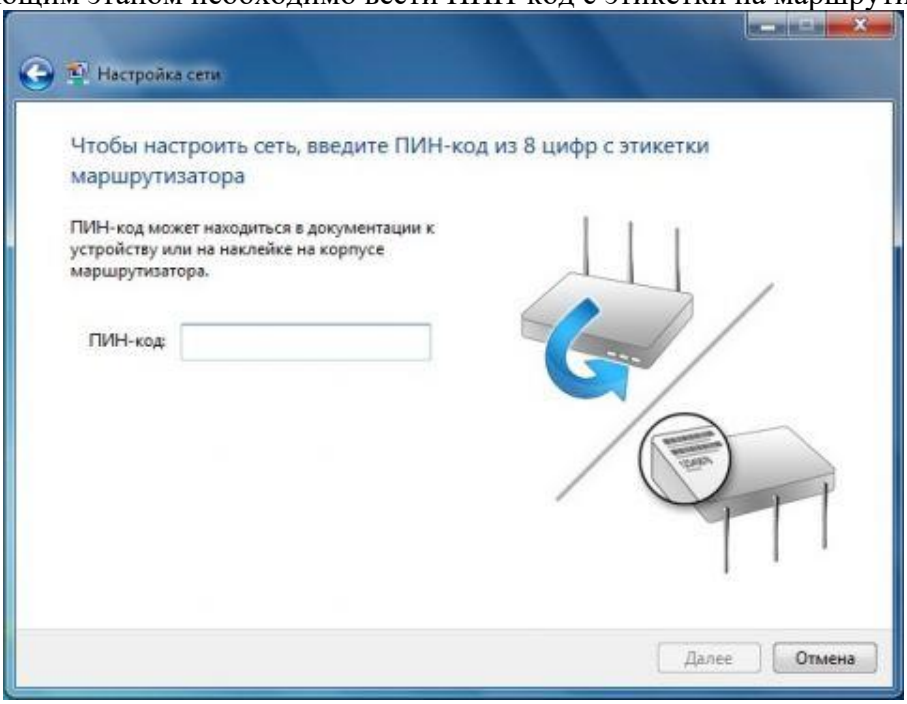
1.1.1 Устройство распаковано и подключено к электросети. Настраивать его можно через Ethernet, используя при этом патчкорд (который входит в комплект поставки) или через WiFi, но от этого никак не зависит сам процесс настройки. На ноутбуке или десктопе нужно зайти в **Панель Управления** – **Центр управления сетями и общим доступом** – **Настройка нового подключения или сети**, где выбрать **Создание и настройка новой сети**.



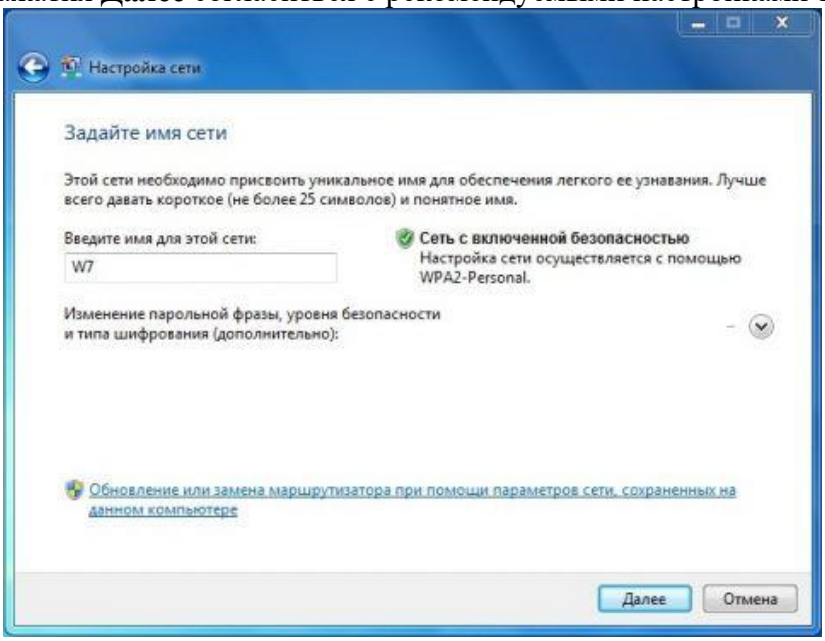
1.1.2 В списке устройств будут видны беспроводные устройства с поддержкой WCN. Выбираем именно нашу точку доступа (беспроводный маршрутизатор).



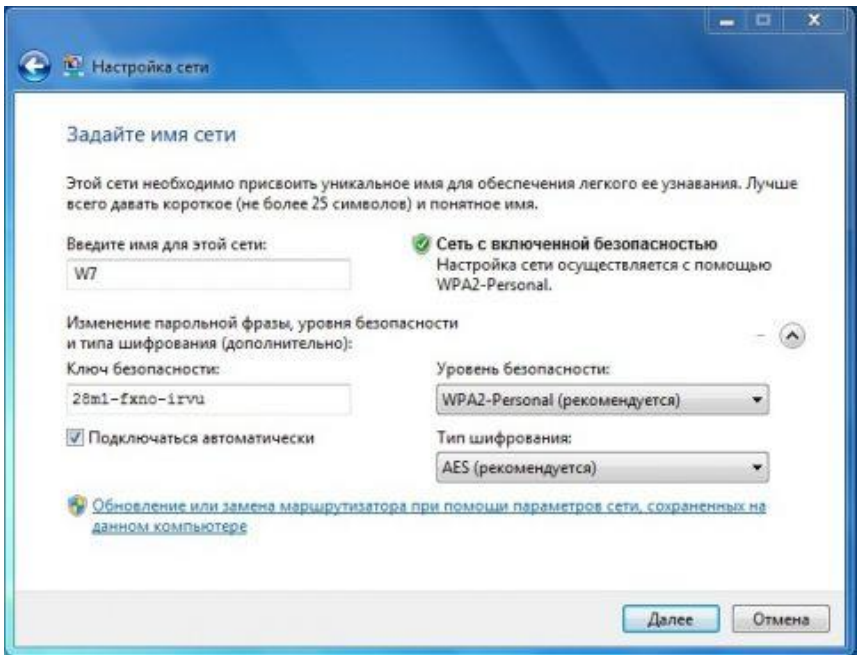
1.1.3 Следующим этапом необходимо вести ПИН-код с этикетки на маршрутизаторе.



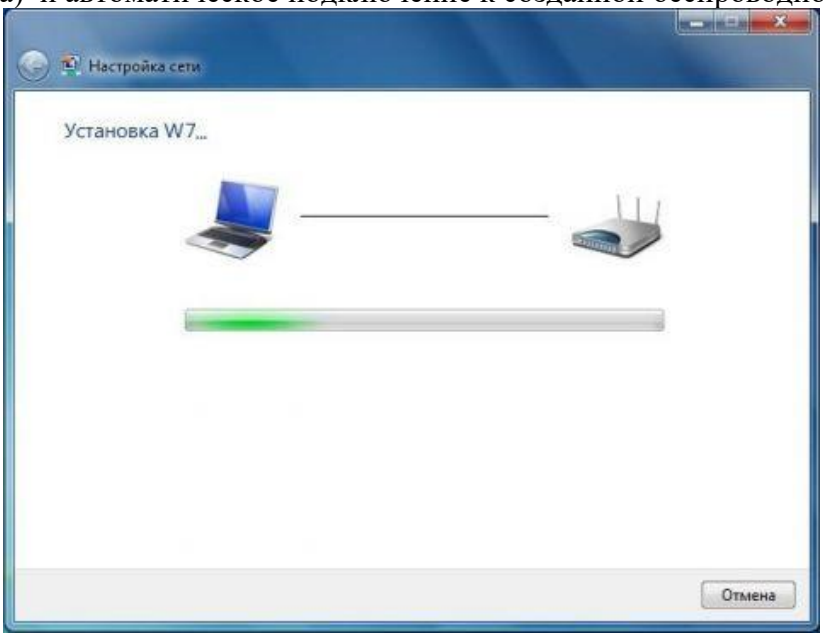
1.1.4 и после нажатия **Далее** согласиться с рекомендуемыми настройками точки доступа

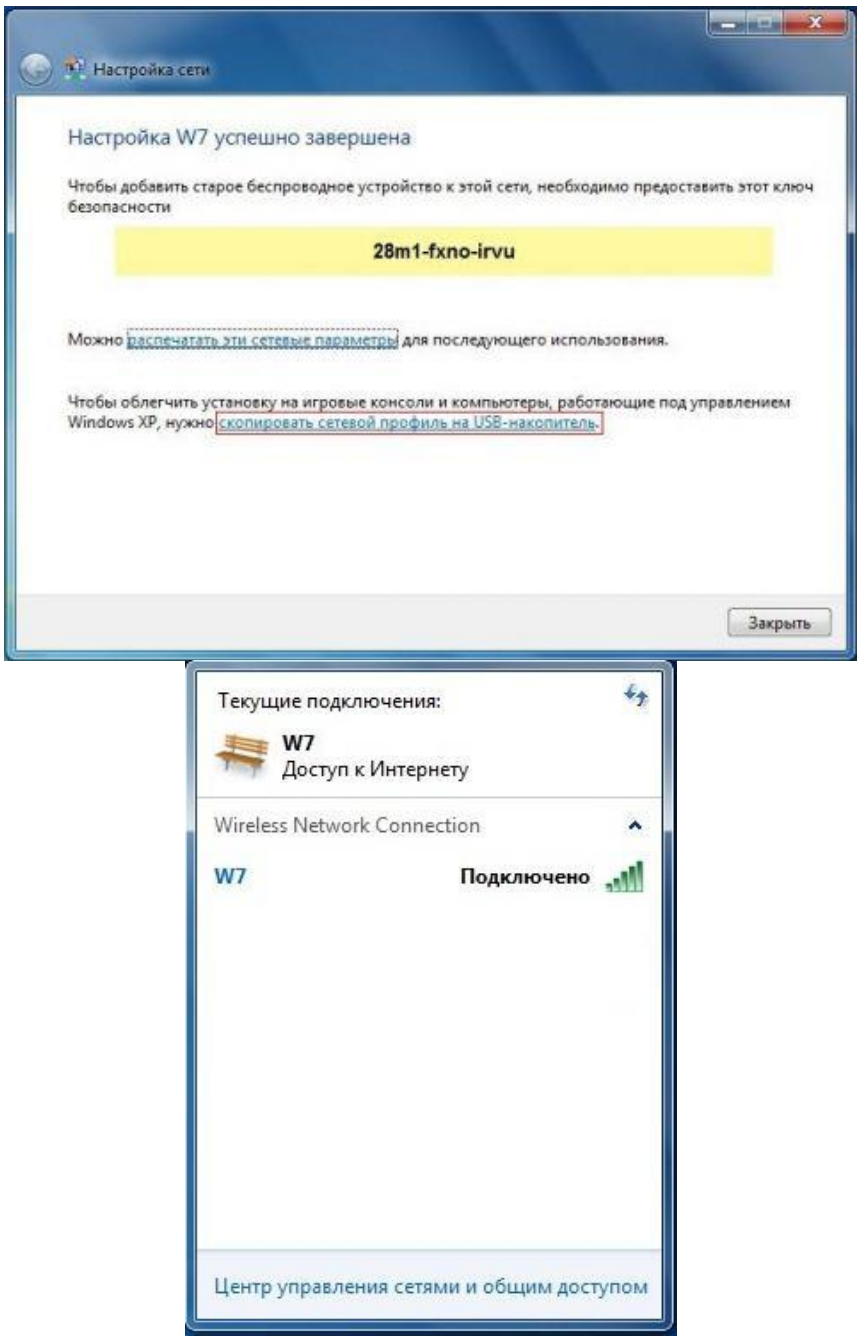


1.1.5 Или задать свои, есть в этом есть необходимость: имя беспроводной сети, пароль для доступа к сети, уровень безопасности и тип шифрования.



1.1.6 После нажатия кнопки **Далее** произойдет настройка точки доступа (беспроводного маршрутизатора) и автоматическое подключение к созданной беспроводной сети.

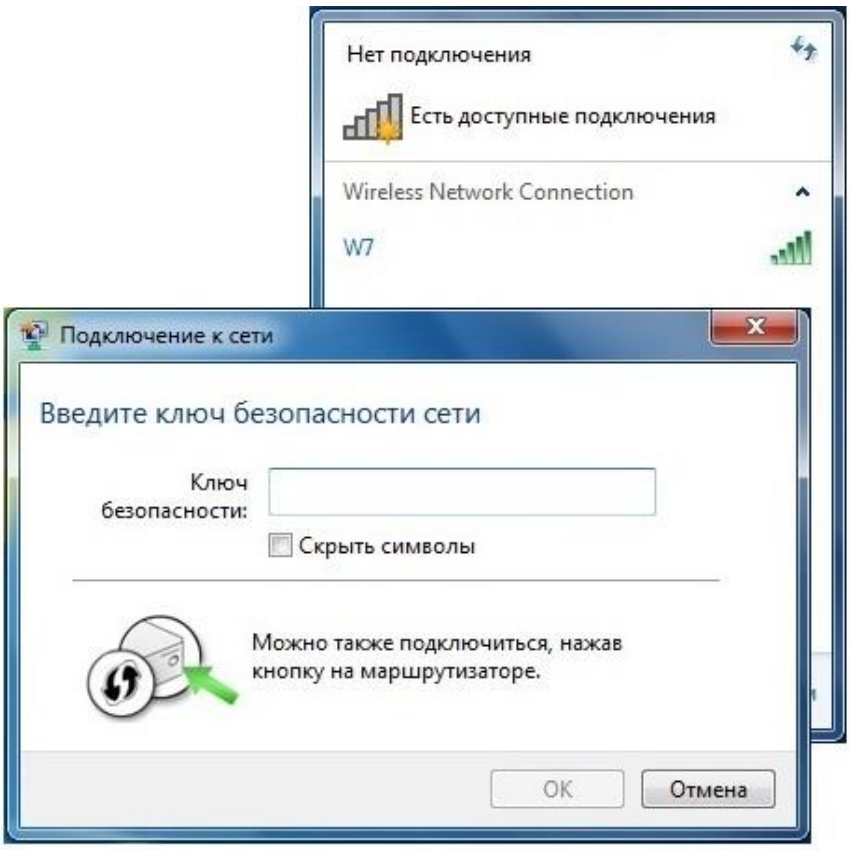




1.1.7 По завершении настройки можно распечатать подробную инструкцию для подключения остальных компьютеров к точке доступа (беспроводному маршрутизатору), а также подготовить флешку с настройками для импорта сетевого профиля на другие беспроводные устройства. Если в данный момент в этом нет необходимости, то это можно сделать позже, в свойствах беспроводной сети.

1.2 Подключиться к точке доступа через Push Button

1.2.1 При подключении к нашей беспроводной сети с компьютера под управлением Windows 7, можно не вводить ключ безопасности, а нажать кнопку WCN на маршрутизаторе. Подключение к беспроводной сети произойдет автоматически.

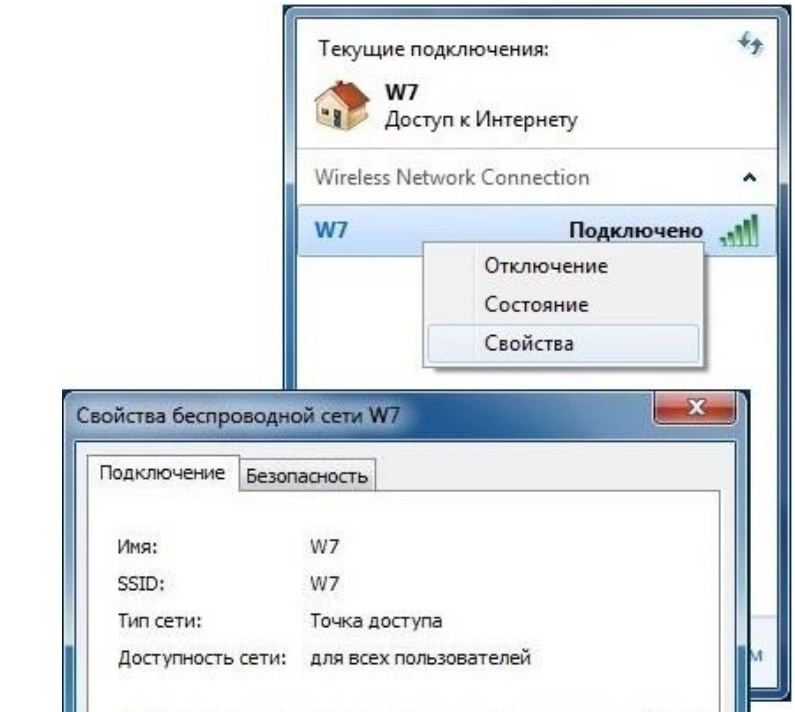


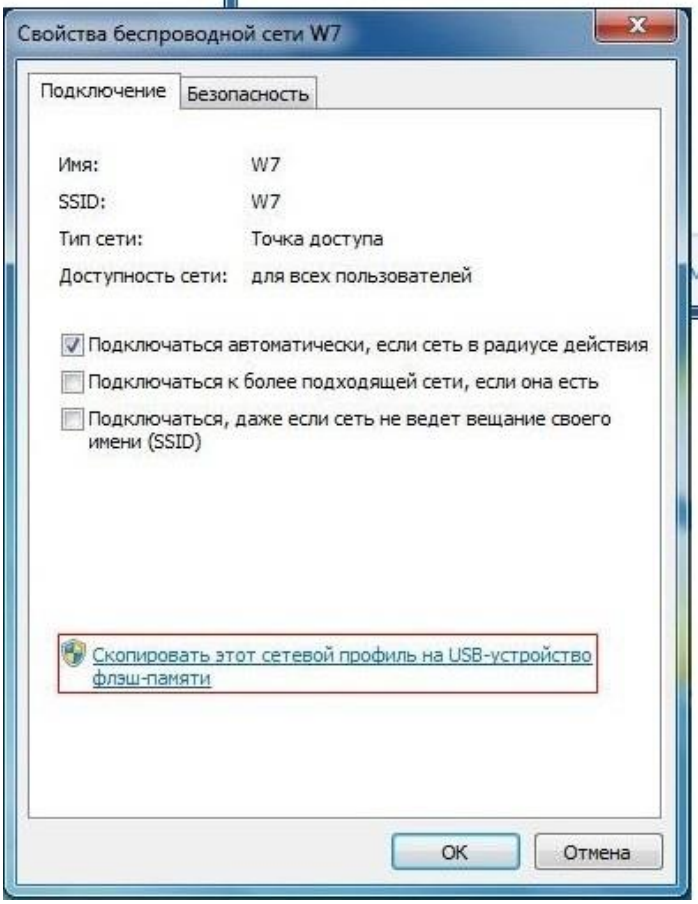
1.2.2 На беспроводных устройствах, поддерживающих метод PBC, достаточно нажать кнопку WPS на маршрутизаторе, а потом на беспроводном устройстве, после чего произойдет подключения устройства к беспроводной сети.

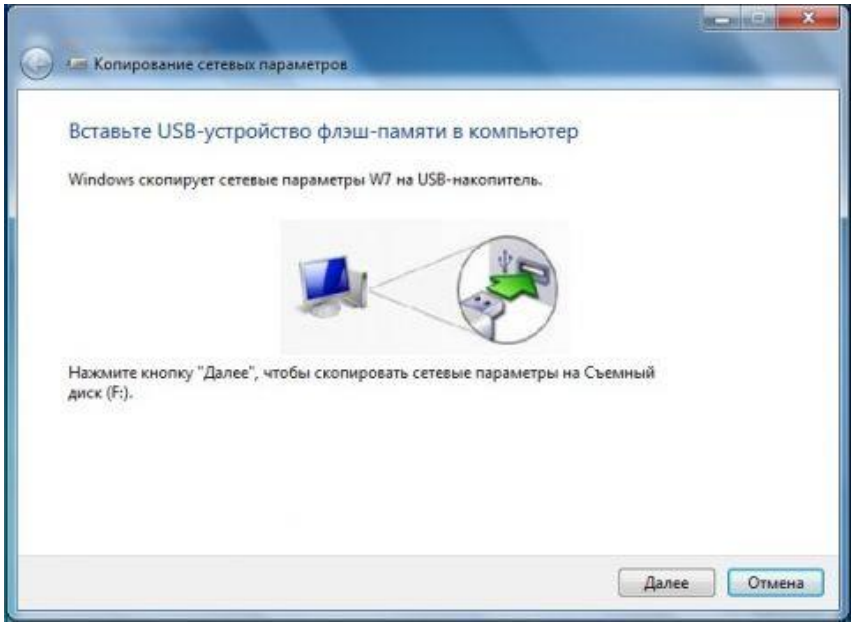
1.2.3 На компьютерах, работающих под управлением более старых операционных систем Windows, а также на беспроводных устройствах, не поддерживающих метод Push Button, необходимо воспользоваться импортом профиля сетевого подключения к беспроводной сети.

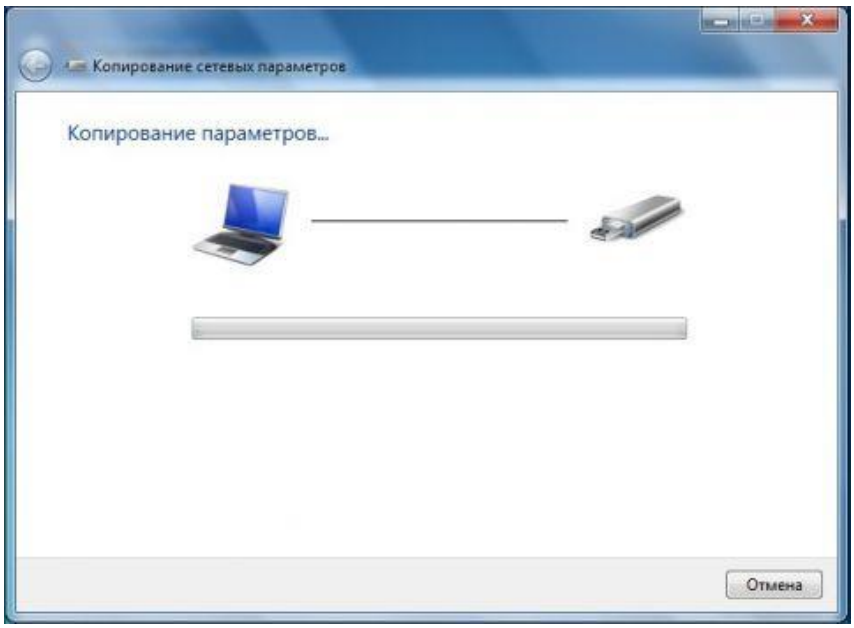
1.3 Подключение к точке доступа через импорт профиля сетевого подключения

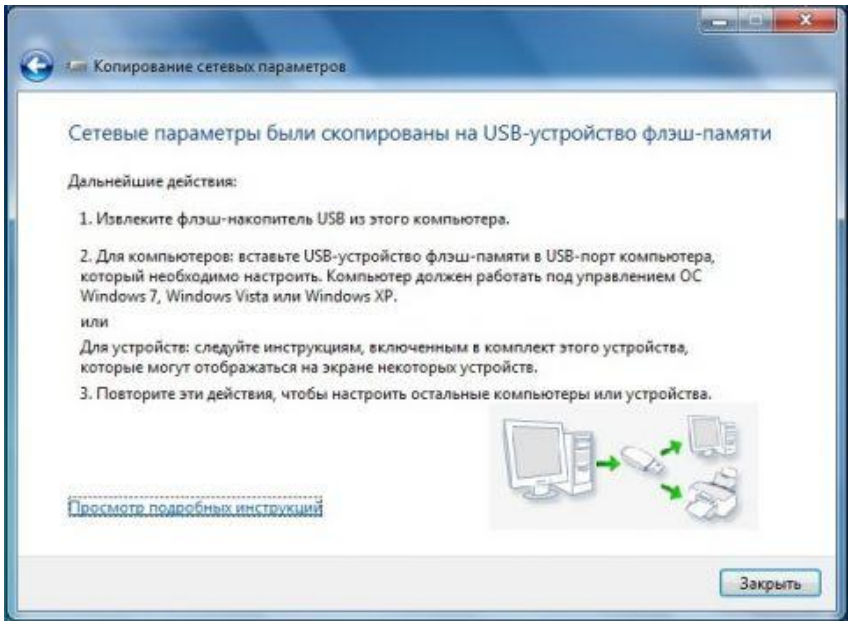
1.3.1 Если USB флеш диск с настройками сетевого профиля не был создан по завершении настройки точки доступа (беспроводного маршрутизатора), то нужно его создать. Для этого необходимо подключить USB флеш диск, в **центре соединений** вызвать **свойства** беспроводной сети и выбрать **Скопировать этот сетевой профиль на USB Устройство флеш-памяти.** Также можно открыть свойства беспроводной сети через **Панель Управления** – **Центр управления сетями и общим доступом** – **Управление беспроводными сетями**.


Текущие подключения:
W7
Доступ к Интернету
Wireless Network Connection
W7
Подключено
Отключение
Состояние
Свойства


Свойства беспроводной сети W7
Подключение
Безопасность
Имя: W7
SSID: W7
Тип сети: Точка доступа
Доступность сети: для всех пользователей
Подключаться автоматически, если сеть в радиусе действия
Подключаться к более подходящей сети, если она есть
Подключаться, даже если сеть не ведет вещание своего имени (SSID)
Скопировать этот сетевой профиль на USB-устройство флэш-памяти
OK
Отмена


Копирование сетевых параметров
Вставьте USB-устройство флэш-памяти в компьютер
Windows скопирует сетевые параметры W7 на USB-накопитель.
Нажмите кнопку "Далее", чтобы скопировать сетевые параметры на Съемный диск (F:).
Далее
Отмена

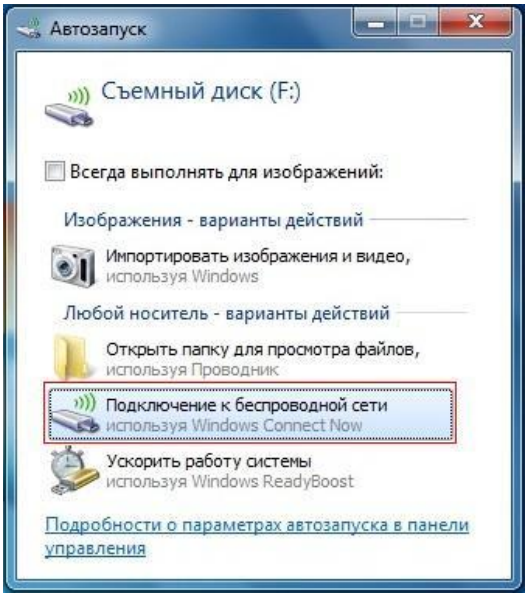




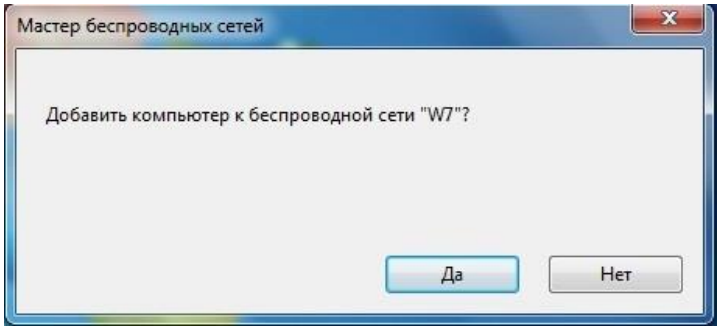
1.3.2 По завершении работы мастера USB флеш диск может использоваться для подключения различных беспроводных устройств, а также компьютеров, оснащенных беспроводным адаптером и работающих под управлением Windows XP/Vista/Windows 7.

1.3.3 Для устройств с беспроводным адаптером, таких как фоторамки, принтеры, игровые консоли, необходимо подключить к ним USB флеш диск с сетевым профилем и согласиться с импортом настроек. По окончании настройки устройство автоматически подключится к беспроводной сети. Аналогичные действия и для операционных систем Windows: подключить USB флеш диск с сетевым профилем,

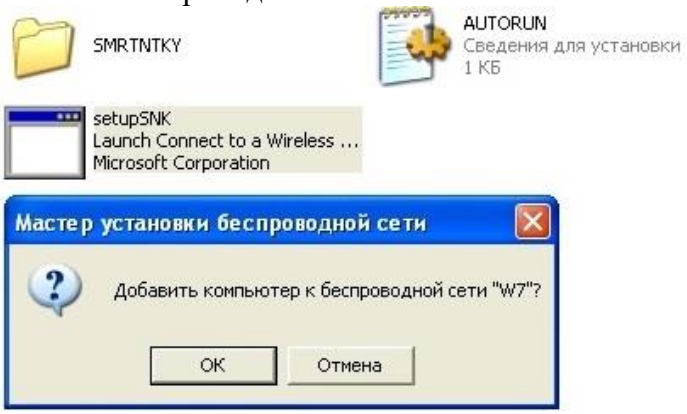
1.3.4 в окне **Автозапуска** выбрать **Подключение к беспроводной сети используя Windows Connect Now**,



1.3.5 согласиться с импортом настроек.



1.3.6 Если на компьютере с Windows отключена функция автозапуска, необходимо открыть USB флеш диск, запустить файл **SetupSNK.Exe** и согласиться с импортом настроек для подключения к беспроводной сети.



1.3.7 Windows Connect Now значительно упрощает настройку беспроводной сети и теперь вам не потребуется запоминать учетные данные сети и тратить много времени для подключения новых ПК.

2. **Содержание отчета**:

7.1 Наименование и цель работы
7.2 Выполненное задание
7.3 Ответы на контрольные вопросы
7.4 Вывод о проделанной работе

Контрольные вопросы

1. Какие процедуры и функции уровней эталонной сетевой модели ISO/OSI реализованы в маршрутизаторе?
2. Укажите, в чем заключается основное отличие принципов работы коммутаторов от маршрутизаторов? Приведите пример, основываясь на схеме проекта.
3. Укажите длину IP-адреса протокола IPv4 и протокола IPv6?
4. Объясните назначение следующих адресов:
    - 169.254.0.0/16;
    - 127.0.0.0/8;
    - 255.255.255.255;
    - 224.0.0.0;
    - 240.0.0.0;
5. Объясните необходимость наличия маски подсети, при указании IP-адреса устройства?
6. Перечислите служебные адреса подсети IPv4? Опишите границы частных диапазонов адресов IPv4?
7. Дайте определение понятиям: классовая маршрутизация, VLSM, CIDR?
8. Вычислите адреса и маски сетей по имеющимся IP-адресам:
    - 10.10.12.7 255.255.255.240;
    - 192.168.1.4 255.255.128.1;
    - 172.31.100.15 255.252.0.0;
9. Приведите недостатки и достоинства метода статической маршрутизации в сети ЛВС?
10. Какие среды конфигурации беспроводных устройств вы знаете?
11. Какая технология беспроводного подключения в настоящее время считается наиболее простой и безопасной?

Практическое занятие 8 Администрирование беспроводной сети

Основной целью администрирования является приведение сети в соответствие с целями и задачами, для которых она предназначена. Администрирование заключается в контроле за работой сетевого оборудования и управление функционированием сети в целом. Администрирование выполняет администратор сети – специалист, отвечающий за нормальное функционирование и использование ресурсов сети. Если более детально, то администрирование информационных систем включает следующие цели:

- Установка и настройка сети. Поддержка её дальнейшей работоспособности.
- Мониторинг. Планирование системы.
- Установка и конфигурация аппаратных устройств.
- Установка программного обеспечения.
- Архивирование (резервное копирование) информации.
- Создание и управление пользователями.
- Установка и контроль защиты.

Вот выписка должностных обязанностей администратора сети:

- Устанавливает на серверы и рабочие станции сетевое программное обеспечение.
- Конфигурирует систему на сервере.
- Обеспечивает интегрирование программного обеспечения на файл-серверах, серверах систем управления базами данных и на рабочих станциях.
- Поддерживает рабочее состояние программного обеспечения сервера.
- Регистрирует пользователей, назначает идентификаторы и пароли.
- Обучает пользователей работе в сети, ведению архивов; отвечает на вопросы пользователей, связанные с работой в сети; составляет инструкции по работе с сетевым программным обеспечением и доводит их до сведения пользователей.
- Контролирует использование сетевых ресурсов.
- Организует доступ к локальной и глобальной сетям.
- Устанавливает ограничения для пользователей по:
- использованию рабочей станции или сервера;
- времени;
- степени использования ресурсов.
- Обеспечивает своевременное копирование и резервирование данных.
- Обращается к техническому персоналу при выявлении неисправностей сетевого оборудования.
- Участвует в восстановлении работоспособности системы при сбоях и выходе из строя сетевого оборудования.
- Выявляет ошибки пользователей и сетевого программного обеспечения и восстанавливает работоспособность системы.
- Проводит мониторинг сети, разрабатывает предложения по развитию инфраструктуры сети.
- Обеспечивает:
    - сетевую безопасность (защиту от несанкционированного доступа к информации, просмотра или изменения системных файлов и данных);
    - безопасность межсетевого взаимодействия.

- Готовит предложения по модернизации и приобретению сетевого оборудования.
- Осуществляет контроль за монтажом оборудования специалистами сторонних организаций.
- Сообщает своему непосредственному руководителю о случаях злоупотребления сетью и принятых мерах.
- Ведет журнал системной информации, иную техническую документацию.

Любая беспроводная сеть состоит как минимум из двух базовых компонентов – точки беспроводного доступа и клиента беспроводной сети (режим ad-hoc, при котором клиенты беспроводной сети общаются друг с другом напрямую без участия точки доступа, мы рассматривать не будем). Стандартами беспроводных сетей 802.11a/b/g предусматривается несколько механизмов обеспечения безопасности, к которым относятся различные механизмы аутентификации пользователей и реализация шифрования при передаче данных.

Протокол WEP

Все современные беспроводные устройства (точки доступа, беспроводные адаптеры и маршрутизаторы) поддерживают протокол безопасности WEP (Wired Equivalent Privacy), который был изначально заложен в спецификацию беспроводных сетей IEEE 802.11. Данный протокол является своего рода аналогом проводной безопасности (во всяком случае, расшифровывается он именно так), однако реально никакого эквивалентного проводным сетям уровня безопасности он, конечно же, не предоставляет.

Протокол WEP позволяет шифровать поток передаваемых данных на основе алгоритма RC 4 с ключом размером 64 или 128 бит.

Данные ключи имеют так называемую статическую составляющую длиной от 40 до 104 бит и дополнительную динамическую составляющую размером 24 бита, называемую вектором инициализации (Initialization Vector, IV).

На простейшем уровне процедура WEP-шифрования выглядит следующим образом: первоначально передаваемые в пакете данные проверяются на целостность (алгоритм CRC-32), после чего контрольная сумма (integrity check value, ICV) добавляется в служебное поле заголовка пакета. Далее генерируется 24-битный вектор инициализации, (IV) и к нему добавляется статический (40-или 104-битный) секретный ключ. Полученный таким образом 64-или 128-битный ключ и является исходным ключом для генерации псевдослучайного числа, использующегося для шифрования данных. Далее данные смешиваются (шифруются) с помощью логической операции XOR с псевдослучайной ключевой последовательностью, а вектор инициализации добавляется в служебное поле кадра. Вот, собственно, и всё.

Протокол безопасности WEP предусматривает два способа аутентификации пользователей: Open System (открытая) и Shared Key (общая). При использовании открытой аутентификации никакой аутентификации, собственно, и не существует, то есть любой пользователь может получить доступ в беспроводную сеть. Однако даже при использовании открытой системы допускается использование WEP-шифрования данных.

Протокол WPA

Как будет показано чуть позже, протокол WEP имеет ряд серьёзных недостатков и не является для взломщиков труднопреодолимым препятствием. Поэтому в 2003 году был представлен следующий стандарт безопасности — WPA (Wi-Fi Protected Access). Главной особенностью этого стандарта является технология динамической генерации ключей шифрования данных, построенная на базе протокола TKIP (Temporal Key Integrity Protocol), представляющего собой дальнейшее развитие алгоритма шифрования RC 4. По протоколу TKIP сетевые устройства работают с 48-битовым вектором инициализации (в отличие от 24-битового вектора WEP) и реализуют правила изменения последовательности его битов, что исключает повторное использование ключей. В

протоколе TKIP предусмотрена генерация нового 128-битного ключа для каждого передаваемого пакета. Кроме того, контрольные криптографические суммы в WPA рассчитываются по новому методу под названием MIC (Message Integrity Code). В каждый кадр здесь помещается специальный восьмибайтный код целостности сообщения, проверка которого позволяет отражать атаки с применением подложных пакетов. В итоге получается, что каждый передаваемый по сети пакет данных имеет собственный уникальный ключ, а каждое устройство беспроводной сети наделяется динамически изменяемым ключом.

Кроме того, протокол WPA поддерживает шифрование по стандарту AES (Advanced Encryption Standard), то есть по усовершенствованному стандарту шифрования, который отличается более стойким криптоалгоритмом, чем это реализовано в протоколах WEP и TKIP.

При развёртывании беспроводных сетей в домашних условиях или небольших офисах обычно используется вариант протокола безопасности WPA на основе общих ключей – WPA-PSK (Pre Shared Key). В дальнейшем мы будем рассматривать только вариант WPA-PSK, не касаясь вариантов протокола WPA, ориентированных на корпоративные сети, где авторизация пользователей проводится на отдельном RADIUS-сервере.

При использовании WPA-PSK в настройках точки доступа и профилях беспроводного соединения клиентов указывается пароль длиной от 8 до 63 символов.

Фильтрация MAC-адресов

Фильтрация MAC-адресов, которая поддерживается всеми современными точками доступа и беспроводными маршрутизаторами, хотя и не является составной частью стандарта 802.11, тем нее менее, как считается, позволяет повысить уровень безопасности беспроводной сети. Для реализации данной функции в настройках точки доступа создаётся таблица MAC-адресов беспроводных адаптеров клиентов, авторизованных для работы в данной сети.

Режим скрытого идентификатора сети SSID

Ещё одна мера предосторожности, которую часто используют в беспроводных сетях – это режим скрытого идентификатора сети. Каждой беспроводной сети назначается свой уникальный идентификатор (SSID), который представляет собой название сети. Когда пользователь пытается войти в сеть, то драйвер беспроводного адаптера прежде всего сканирует эфир на наличие в ней беспроводных сетей. При использовании режима скрытого идентификатора (как правило, этот режим называется Hide SSID) сеть не отображается в списке доступных, и подключиться к ней можно только в том случае, если, во-первых, точно известен её SSID, и, во-вторых, заранее создан профиль подключения к этой сети.

Взлом беспроводной сети с протоколом WEP

Чтобы у читателя не сложилось впечатления, что перечисленных средств защиты вполне достаточно, дабы не опасаться непрошенных гостей, поспешим его разочаровать. И начнём мы с инструкции по взлому беспроводных сетей стандарта 802.11 b / g на базе протокола безопасности WEP.

Собственно, утилит, специально разработанных для взлома таких сетей и доступных в Интернете, предостаточно. Правда, есть одно «но». Почти все они «заточены» под Linux-системы. Собственно, с точки зрения продвинутого пользователя – это не только не помеха, но и наоборот. А вот обычными пользователями операционная система Linux используется редко, поэтому мы решили ограничиться рассмотрением утилит, поддерживаемых системой Windows XP.

Итак, для взлома сети нам, кроме ноутбука с беспроводным адаптером, потребуется утилита aircrack 2.4, которую можно найти в свободном доступе в Интернете.

Данная утилита поставляется сразу в двух вариантах: под Linux и под Windows, поэтому нас будут интересовать только те файлы, которые размещены в директории aircrack-2.4\win 32.

В этой директории имеется три небольших утилиты (исполняемых файлов): airodump.exe, aircrack.exe и airdecap.exe.
Первая утилита предназначена для перехвата сетевых пакетов, вторая – для их анализа и получения пароля доступа и третья – для расшифровки перехваченных сетевых файлов.
Конечно же, не всё так просто, как может показаться. Дело в том, что все подобные программы «заточены» под конкретные модели чипов, на базе которых построены сетевые адаптеры. То есть не факт, что выбранный произвольно беспроводной адаптер окажется совместим с программой aircrack-2.4. Более того, даже при использовании совместимого адаптера (список совместимых адаптеров, а точнее – чипов беспроводных адаптеров, можно найти в документации к программе) придётся повозиться с драйверами, заменив стандартный драйвер от производителя сетевого адаптера на специализированный драйвер под конкретный чип. К примеру, в ходе тестирования мы выяснили, что стандартный беспроводной адаптер Intel PRO Wireless 2200 BG, который является составной частью многих ноутбуков на базе технологии Intel Centrino, просто не совместим с данной программой при использовании ОС Windows XP (правда, он поддерживается при использовании Linux-версии программы). В итоге мы остановили свой выбор на беспроводном PCMCIA-адаптере Gigabyte GN-WMAG на базе чипа Atheros. При этом сам беспроводной адаптер устанавливался как Atheros Wireless Network Adapter с драйвером 3.0.1.12.
Сама процедура взлома беспроводной сети достаточно проста. Начинаем с запуска утилиты airodump.exe, которая представляет собой сетевой сниффер для перехвата пакетов. При запуске программы (рис. 1) откроется диалоговое окно, в котором потребуется указать беспроводной сетевой адаптер (Network interface index number), тип чипа сетевого адаптера (Network interface type (o/a)), номер канала беспроводной связи (Channel (s): 1 to 14, 0= all) (если номер канал неизвестен, то можно сканировать все каналы). Также задаётся имя выходного файла, в котором хранятся перехваченные пакеты (Output filename prefix) и указывается, требуется ли захватывать все пакеты целиком (cap-файлы) или же только часть пактов с векторами инициализации (ivs-файлы) (Only write WEP IVs (y/n)). При использовании WEP-шифрования для подбора секретного ключа вполне достаточно сформировать только ivs-файл. По умолчанию ivs-или с ap-файлы создаются в той же директории, что и сама программа airodump.

После настройки всех опций утилиты airodump откроется информационное окно, в котором отображается информация об обнаруженных точках беспроводного доступа, информация о клиентах сети и статистика перехваченных пакетов (рис. 2).

Если точек доступа несколько, статистика будет выдаваться по каждой из них.
Первым делом, запишите MAC-адрес точки доступа, SSID беспроводной сети и MAC-адрес одного из подключённых к ней клиентов (если их несколько). Ну а дальше нужно подождать, пока не будет перехвачено достаточное количество пакетов.
Количество пакетов, которые нужно перехватить для успешного взлома сети, зависит от длины WEP-ключа (64 или 128 бит) ну и, конечно же, от удачи. Если в сети используется 64-битный WEP-ключ, то для успешного взлома вполне достаточно захватить пол миллиона пакетов, а во многих случаях – даже меньше. Время, которое для этого потребуется, зависит от интенсивности трафика между клиентом и точкой доступа, но, как правило, составляет не более нескольких минут. В случае же использования 128-битного ключа для гарантированного взлома потребуется перехватить порядка двух миллионов пакетов. Для останова процесса захвата пакетов (работы утилиты) используется комбинация клавиш Ctrl+C.

После того, как выходной ivs-файл сформирован, можно приступать к его анализу. В принципе, это можно делать и параллельно вместе с перехватами пакетов, но для простоты мы рассмотрим последовательное выполнение процедур перехвата и анализа.
Для анализа сформированного ivs-файла потребуется утилита aircrack.exe, которая запускается из командной строки. В нашем случае мы использовали следующие параметры запуска:
aircrack.exe –b 00:13:46:1C:A4:5F –n 64 –i 1 out.ivs
В данном случае «-b 00:13:46:1C:A4:5F» – это указание MAC-адреса точки доступа, «-n 64» – указание длины используемого ключа шифрования, «-i 1» – индекс ключа, а «out.ivs» – это файл, который подвергается анализу.
Полный перечень параметров запуска утилиты можно посмотреть, набрав в командной строке команду aircrack.exe без параметров.
В принципе, поскольку такая информация, как индекс ключа и длина ключа шифрования, как правило, заранее неизвестна, обычно используется следующий упрощённый вариант запуска команды: aircrack.exe out.ivs.
Результат анализа ivs-файла показан на рис. 4. Вряд ли строка KEY FOUND! Нуждается в комментариях. И обратите внимание, что секретный ключ был вычислен всего за 3 секунды.

Мы проводили множество экспериментов с использованием и 128-битного ключа, и с различными параметрами запуска команды aircrack.exe, но во всех случаях время, за которое вычислялся секретный ключ, не превосходило 7 секунд.
Вот так просто и быстро проводится вскрытие беспроводных сетей с WEP-шифрованием, и говорить о «безопасности» сетей в данной случае вообще неуместно. Ну, действительно, можно ли говорить о том, чего на самом деле нет!
В самом начале мы упомянули, что во всех точках доступа имеются ещё такие возможности, как использование режима скрытого идентификатора сети и фильтрации по MAC-адресам, которые призваны повысить безопасность беспроводной сети. Но не будьте оптимистами – это не спасает.
На самом деле, не таким уж и невидимым является идентификатор сети даже при активации этого режима на точке доступа. К примеру, уже упомянутая нами утилита airodump всё равно покажет вам SSID сети, который впоследствии можно использовать для создания профиля подключения к сети (причём несанкционированного подключения).
Ну а если говорить о такой наивной мере безопасности, как фильтрация по MAC-адресам, то здесь вообще всё очень просто. Существует достаточно много разнообразных утилит и под Linux, и под Windows, которые позволяют подменять MAC-адрес сетевого интерфейса. К примеру, для несанкционированного доступа в сеть мы подменяли MAC-адрес беспроводного адаптера с помощью утилиты SMAC 1.2 (рис. 5). Естественно, что в качестве нового MAC-адреса используется MAC-адрес авторизованного в сети клиента, который определяется всё той же утилитой airodump.

Итак, преодолеть всю систему безопасности беспроводной сети на базе WEP-шифрования не представляет никакого труда. Возможно, многие скажут, что это малоактуально, поскольку WEP-протокол давно умер и его просто не используют. Ведь на смену ему пришёл более стойкий протокол WPA. Однако не будем торопиться с выводами. Отчасти это действительно так, но только отчасти. Дело в том, что в некоторых случаях для увеличения радиуса действия беспроводной сети разворачиваются так называемые распределённые беспроводные сети (WDS) на базе нескольких точек доступа. Но самое интересное заключается в том, что эти самые распределённые сети не поддерживают WPA-протокола, и единственной допустимой мерой безопасности в данном случае

является применение WEP-шифрования. Ну а взламываются эти WDS-сети абсолютно так же, как и сети на базе одной точки доступа.
Теперь посмотрим, как обстоят дела с сетями на базе WPA-шифрования.

Взлом беспроводной сети с протоколом WPA

Собственно, сама процедура взлома сетей с протоколом WPA мало чем отличается от уже рассмотренной процедуры взлома сетей с WEP-протоколом.
На первом этапе используется всё тот же сниффер airodump. Однако есть два важных аспекта, которые необходимо учитывать. Во-первых, в качестве выходного файла необходимо использовать именно cap-файл, а не ivs-файл. Для этого в настройке утилиты airodump на последней вопрос «Only write WEP IVs (y/n)» отвечаем «нет».
Во-вторых, в cap-файл необходимо захватить саму процедуру инициализации клиента в сети, то есть придётся посидеть в «засаде» с запущенной программой airodump. Если используется Linux-система, то можно провести атаку, которая заставит произвести процедуру переинициализации клиентов сети, а вот под Windows такая программка не предусмотрена.
После того, как в cap-файл захвачена процедура инициализации клиента сети, можно остановить программу airodump и приступить к процессу расшифровки. Накапливать перехваченные пакеты в данном случае нет необходимости, поскольку для вычисления секретного ключа используется только пакеты, передаваемые между точкой доступа и клиентом в ходе инициализации.
Для анализа полученной информации используется все та же утилита aircrack, но с несколько иными параметрами запуска. Кроме того, в директорию с программой aircrack придётся установить ещё один важный элемент – словарь. Такие специализированные словари можно найти в Интернете, например, по ссылке http://ftp.se.kde.org/pub/security/tools/net/Openwall/wordlists/.
После этого запускаем из командной строки программу aircrack (рис 6), указывая в качестве выходного файла cap-файл (например, out. cap) и название словаря (параметр – w all, где all – название словаря).

Программа перебора ключей из словаря даёт очень интенсивную нагрузку на процессор, так что если для этого используется маломощный ПК, то на эту процедуру потребуется много времени. Если же для этого используется мощный многопроцессорный сервер или ПК на базе двухъядерного процессора, то в качестве опции можно указать количество используемых процессоров. К примеру, в нашем случае использовался новейший двухядерный процессор Intel Pentium Extreme Edition Processor 955 с поддержкой технологии Hyper-Threading (четыре логических ядра процессора), поэтому в параметрах запуска программы мы использовали опцию «–p 4», что позволило утилизировать все четыре логических ядра процессора, причём каждое ядро утилизируется на 100%. В результате после почти полутора часов работы программы секретный ключ был найден! (рис. 7.)

Это, конечно, не несколько секунд, как в случае с WEP-шифрованием, но тоже неплохой результат, который прекрасно демонстрирует, что и WPA-PSK защита не является абсолютно надёжной. Причём результат взлома секретного ключа никак не связан с тем, какой алгоритм шифрования (TKIP или AES) используется в сети.

Настройка соединения (провайдер dhcp):
Шаг 1. Необходимо на вкладку «Дополнительные настройки»-> «WAN»:
Нужно установить тип подключения WAN: «Динамический IP».

Шаг 2. На этой же вкладке, можно сразу задать адреса DNS:

Если мы не знаем - необходимо оставить «автоматическое» подключение к серверу DNS (как на рисунке).

Шаг 3 (опционально). Если провайдер сети Интернет - осуществляет привязку к MAC-адресу, роутер ASUS RT N12 C1 (как и более «старый») - позволит вам «подменить» его. Снизу вкладки, находится поле MAC-адреса:

Сюда, заносится требуемое вам значение адреса (последние 5 байт, формат HEX). А посмотреть значение аппаратного адреса карты ПК, можно на «Состоянии» соединения («Поддержка» -> «Подробности»).

По завершении этих настроек нажимаем: «Применить». Настройка роутера Асус RT N12 на соединение «DHCP» - завершена. Оно появится сразу же (через 10-15 секунд после нажатия на «Применить»).

Настройка соединения (провайдер pppoe):

Шаг 1. На вкладке «Дополнительные настройки» -> «WAN», установиТЬ «тип соединения»: PPPoE.

Шаг 2. Подразумевается, что происходит настройка соединения для «динамически» выделяемого IP-адреса. Поэтому, галочки «получить IP WAN автоматически», и «подключиться к DNS-серверу автоматически» - оставляем, как есть. На этом этапе, нужно только заполнить поля имени пользователя и пароля (см. договор).

Шаг 3 (опционально). Мы можем аналогично настройке для случая «DHCP», вписать «клонируемый» адрес «MAC» (в последнее поле на вкладке).

По завершении этих действий, нужно нажать «Применить». Соединение - будет создано.

Мы здесь рассмотрели, как настроить роутер RT N12 на соединение «DHCP» и «PPPoE»-типа (с «динамически» выделяемым IP-адресом). Предусмотрены и другие «типы» соединений, в том числе, если роутер - «клиент VPN», то настраивать нужно L2TP или PPTP (выбирается в зависимости от условий провайдера). А если соединение - создано, и подключено, осталось только «запомнить» настройки.

Запоминание настроек: Для выполнения этого действия, заходим на вкладку «Администрирование» -> «Восстановить… загрузить настройки»:

Именно кнопка «Сохранить», позволяет «запомнить» все сделанные изменения в постоянной памяти роутера. Тогда как, нажатие «Восстановить» - сбросит роутер Асус RT N12 к значениям «по умолчанию», причем - немедленно.

Если мы будем смотреть Интернет-телевидение через компьютеры нашей сети, то в самом роутере, нужно выполнить пару настроек. Установить две галочки (вкладка «Дополнительные настройки» -> «ЛВС» -> «Маршрут»).

Нажать «Применить». Теперь, Multicast-пакеты (многоадресные пакеты), будут «пропускаться» в локальную сеть. Пакеты Multicast, используются в цифровом телевидении.

Дополнительно, можно искусственно ограничить максимальное значение Multicast-трафика в беспроводной сети Wi-Fi. Необходимо зайти на вкладку «Дополнительные настройки» -> «Локальная сеть» -> «Профессионально», установите требуемое значение

Также, ASUS строго рекомендует сменить админский пароль. Это можно сделать в пункте «Администрирование». Там же можно обновить прошивку роутера RT-N12 C1 или экспортировать настройки устройства или восстановить заводские.

Мы также можем выполнить обновление микропрограммы с сайта ASUS в случае, когда предыдущая программная оболочка становится неактуальной.

При необходимости в программе возможно восстановление заводских настроек, функция сохранения и отправления настроек:

1. Контрольные вопросы

2. Укажите отличительные особенности в принципах работы концентратора и коммутатора? Приведите пример, основываясь на схеме проекта.
3. Каким из указанных в проекте устройств необходимо наличие физических адресов (MAC)?
4. Перечислите режимы коммутации?
5. Приведите разновидности коммутаторов?
6. Объясните, в чем заключается преимущество агрегирования коммутаторов?
7. Что представляет собой логическое объединение коммутаторов в стек?
8. Укажите методы физического подключения для управления современными коммутаторами? Перечислите основные сетевые протоколы управления активным оборудованием компьютерных сетей передачи данных?
9. Выделите отличительные особенности ассиметричной и симметричной коммутации?
10. Вычислите пропускную способность внутренней шины коммутации, если коммутатор работает в неблокирующем режиме и имеет 8 FastEthernet-портов?

Лабораторная работа 8 Расширенная диагностика беспроводной WIFI сети

Медленная скорость загрузки, отсутствие доступа к Сети в определенных уголках квартиры — в домашней сети всегда что-то может пойти не так. Потенциальных причин проблем с Wi-Fi множество: как конфигурация, так и расположение роутера.

Кроме того, у маршрутизатора могут быть внутренние основания нарушать соединение с интернет-провайдером или перегружать только те страницы, информацию с которых вы бы хотели скачать.

Для каждой такой проблемы необходимо систематично выявить свой подход к решению, начав с компьютера и закончив посещенной веб-страницей. Воспользуйтесь для этого веб-интерфейсом своего роутера, командной строкой Windows и рекомендуемыми утилитами.

Кроме того, понадобится установить последний драйвер для Wi-Fi-адаптера вашего компьютера. Чтобы узнать разработчика и модель, нажмите комбинацию клавиш «Win+Pause», вызовите Диспетчер устройств и откройте параметр «Сетевые адаптеры».

Проверяем компьютер
WiFi_1Если совсем не получается установить Wi-Fi-соединение, в первую очередь проверьте, не отключили ли вы случайно Wi-Fi-адаптер компьютера. Для этого вызовите «Параметры» и перейдите к пункту «Сеть и Интернет | Wi-Fi | Настройка параметров адаптера».

Устанавливаем соединение
WiFi_2Если «Беспроводная сеть» отмечена красным крестиком, необходимо задействовать ее либо переключателем на устройстве, либо функциональной клавишей или же отсоединить и снова подключить USB-адаптер. В случае, когда сеть обозначена серым цветом, правой кнопкой мыши вызовите контекстное меню и выберите команду «Подключение», чтобы включить ее со стороны программного обеспечения

Тестируем роутер
WiFi_3Чтобы исключить проблемы, связанные с компьютером, убедитесь, что на вашем роутере активен Wi-Fi. Для этого проверьте, что горит соответствующая лампочка. Если нет, активируйте беспроводное соединение подходящей клавишей на маршрутизаторе или с компьютера через веб-интерфейс. Для этого подключите роутер к ПК сетевым кабелем.

Проверяем Wi-Fi-подключение
WiFi_4Беспроводное соединение установлено, но загрузка тянется мучительно долго? В этом случае следует проверить качество связи между компьютером и маршрутизатором. Для этого с помощью комбинацией клавиш «Win+R» откройте окно «Выполнить», введите команду «cmd» и нажмите клавишу «Enter». В командной строке командой «ipconfig» запустите отображение всех сетевых интерфейсов.

Посылаем ping-запрос
WiFi_5Зачастую список интерфейсов бывает очень длинным. Найти нужный порт, впрочем, легко: рядом с записью «Основной шлюз» указан IP-адрес — он принадлежит

вашему маршрутизатору. В роутерах FritzBox, к примеру, это «192.168.178.1». Введите команду «ping -t 192.168.178.1». После этого Windows станет посылать запрос к роутеру каждую секунду.

Анализируем время ответа
WiFi_6Появление сообщения «Ответ от 192.168.178.1; число байтов=32…» означает, что связь между компьютером и роутером налажена. Прервите отправку команды ping комбинацией клавиш «Ctrl+C» и узнайте точное время ответа. Если значение превышает 20 миллисекунд, соединение необходимо оптимизировать.

Размещаем устройства правильно
WiFi_7Долгие ping-ответы и разрывы соединения являются индикаторами того что, хоть сигнал и проходит, но он очень слабый. Часто для его усиления достаточно поставить маршрутизатор повыше. Постарайтесь не подключать USB-адаптер в порт на задней стороне корпуса. Если разместить его на письменном столе с помощью удлиняющего кабеля, это дополнительно улучшит сигнал.

Меняем радиоканал
WiFi_8Утилита для анализа под названием InSSider (metageek.com) покажет, насколько хорошо и через какой канал функционируют Wi-Fi в вашем окружении. Видите больше 10 беспроводных сетей с той же силой, что и ваша? Подберите один из каналов 1, 5, 9 или 13, в чьем частотном диапазоне меньше всего сетей. Задайте этот канал в маршрутизаторе.

Диагностика Wi-Fi сетей выполняется специализированными утилитами. Среди них есть как платные, так и бесплатные версии, наиболее популярные представлены ниже:

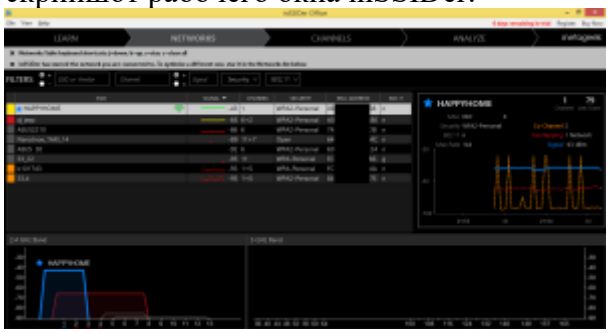
- **WirelessNetView**;
- **Free Wi-Fi Scanner**;
- **InSSIDer.**

Остановимся на утилите **inSSIDer**, так как это мощный инструмент для диагностики беспроводных сетей. Данная программа поможет вам измерить уровень сигнала и оценить производительность вашего Wi-Fi оборудования в различных местах. Вы наглядно сможете посмотреть и проверить, как стены, лестницы, двери, да и в целом планировка и материалы вашего помещения влияют на зону покрытия обслуживаемой беспроводной сети. При этом версия Home — совершенно бесплатна.
В настоящее время в любом современном городе каждый дом или офис буквально кишит обилием Wi-Fi сетей. При этом, когда несколько беспроводных сетей перекрывают канал на котором работают (т.е. несколько **AP** в здании раздают Wi-Fi сеть по одному каналу) возникает ситуация замедления работы всех сетей Wi-Fi на этом канале.
Сканер **inSSIDer** поможет обнаружить наилучший канал для вашего Wi-Fi.
Особенности inSSIDer
К основным особенности данного ПО относят:

- **inSSIDer** использует ваше текущее программное обеспечение беспроводной карты и Wi-Fi подключения
- работает с Microsoft Windows Vista, 7 и 8.1 (32 и 64 бит)
- отслеживает силу принимаемых сигналов в dBm с течением времени
- доступна сортировка по MAC-адресу, SSID, номеру канала, RSSI, и времени

Диагностика WiFi сети при помощи inSSIDer
Установка утилиты не вызывает никаких затруднений. Стоит отметить — если вы используете несколько беспроводных адаптеров, то в в меню **Сетевое подключение** выберите нужный беспроводной адаптер — при его помощи будет выполняться сканирование. Далее программа автоматически проведет сканирование

беспроводных сетей и выведет на экран информацию об эфире. Ниже представлен скриншот рабочего окна inSSIDer:



Рабочее окно программы inSSIDer

Рассмотрим более подробно представленную информацию:

SSID – имя беспроводной сети.

Channel – номер канала, на котором работает беспроводная сеть. Рекомендуется использовать беспроводной канал, на котором работает наименьшее количество других сетей.

RSSI – уровень мощности принимаемого сигнала. Чем выше число RSSI, или чем оно менее отрицательное, тем мощнее сигнал. Старайтесь не делить номер канала (Channel) с точками доступа, которые приближаются к вашей сети по уровню сигнала.

Security – тип безопасности. В некоторых версиях утилиты тип безопасности **WPA2-TKIP** обозначается как **RSNA**, а **WPA2-AES** как**CCMP**.

Max Rate – максимальная скорость работы устройства на физическом уровне (максимальная теоретическая скорость), предоставляемая точкой доступа.

Vendor – производитель точки доступа.

В России разрешены к использованию 13 беспроводных каналов, три из которых являются непересекающимися (это каналы 1, 6 и 11).

Если беспроводной адаптер, установленный на компьютере/ноутбуке/планшетном ПК/смартфоне, предназначен для использования в США, на нем можно будет использовать только каналы с 1 по 11. Поэтому, если установить номер канала 12 или 13 (а также если один из них был выбран алгоритмом автоматического выбора канала), беспроводной клиент не увидит точку доступа. В этом случае необходимо вручную установить номер канала из диапазона с 1 по 11.

Контрольные вопросы

Перечислите преимущества и области использования центральных сетевых контроллеров беспроводных точек доступа?

1. Перечислите ряд распространенных протоколов динамической маршрутизации для локальных и глобальных сетей?
2. Объясните различие терминов «сегментация» и «фрагментация», относительно структуризации данных в эталонной сетевой модели ISO/OSI?
3. Дайте определение понятию «транк» и изобразите подключение, описывающее данный термин?
4. Укажите отличительные черты функционирования сетевых мостов и коммутаторов компьютерных сетей передачи данных?
5. Объясните структуру объединения удаленных сетевых узлов, основанную на принципе микросегментации подключений в компьютерных сетях передачи данных?
6. Опишите основные отличия между способами доступа к разделяемой среде передачи данных по принципу CSMA/CD и CSMA/CA?

7. Укажите основные реализации компьютерных сетей, использующие методы доступа по принципу CSMA/CD и CSMA/CA?
8. Объясните назначение маркировки MDI/MDI-X на портах Ethernet-концентратора?
9. Чем отличаются симметричные алгоритмы шифрования от асимметричных?

Рекомендуемая литература				
1. Основная литература				
Код	**Авторы, составители**	**Заглавие**	**Издательство, год**	**Кол.**
	А.В. Кузин	Компьютерные сети: Учебное пособие /. - 3-е изд., перераб. и доп. -	М.: Форум: ИНФРА-М, 2011. - 192 с.: ил.;	Э1
	Поляк-Брагинский А.В.	Локальная сеть под Linux: Практическое руководство	СПб:БХВ-Петербург, 2010. - 234 с.	Э2
	Шапошников И.В.	Web-сервисы Microsoft .NET: Пособие	СПб:БХВ-Петербург, 2014. - 336 с.	Э3
	Назаров С. В.	Администрирование локальных сетей Windows NT/2000/.NET [Электронный ресурс] : Учеб. пособие	М.: Финансы и статистика, 2003. - 480 е.: ил. -	Э4
2 Дополнительная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л2.1	Будилов В.А.	Интернет-программирование на Java: Пособие /	СПб:БХВ-Петербург, 2014. - 698 с	Э5
Л2.2	В.П. Агальцов	Базы данных. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных: Учебник	М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 272 с.: ил.;	Э6
Л2.3	Н.Н. Заботина	Проектирование информационных систем: Учебное пособие /. -	М.: НИЦ Инфра-М, 2013. - 331 с.	Э7
3 Методическое обеспечение для самостоятельной работы обучающихся				
Код	Авторы, составители	Заглавие	**Издательство, год**	**Кол.**
Л3.1	П.Б. Храмцов [и др.].	Основы Web-технологий учебное пособие	Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017.— 375 с.—	Э8
Л3.2	Семенов Ю.А.	Протоколы и алгоритмы маршрутизации в Интернет	М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 998 с.—	Э9
Л3.3	А. Бражук	Сетевые средства Linux	М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013	Э10
6.2 Электронные образовательные ресурсы				
Э1	http://znanium.com/bookread2.php?book=249563			
Э2	http://znanium.com/bookread2.php?book=350476			

Э3	http://znanium.com/bookread2.php?book=939953
Э4	http://znanium.com/bookread2.php?book=369385
Э5	http://znanium.com/bookread2.php?book=940239
Э6	http://znanium.com/bookread2.php?book=372740
Э7	http://znanium.com/bookread2.php?book=371912
Э8	http://www.intuit.ru/studies/curriculums/912/info
Э9	http://www.intuit.ru/studies/courses/1123/200/info
Э10	http://www.intuit.ru/studies/courses/681/537/info