Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Д. М. Романенко

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

**Тексты лекций для студентов
специальности 1-40 01 02-03
«Информационные системы и технологии
(издательско-полиграфический комплекс)»**

Минск 2014

УДК 004.7(075.8)(0..34.2)
ББК 73я73
Р31

Рассмотрены и рекомендованы редакционно-издательским советом
Белорусского государственного технологического университета

Рецензенты:
доцент кафедры программного обеспечения
информационных технологий БГУИР, кандидат технических наук
А. Т. Пешков;
кандидат технических наук, заведующий кафедрой полиграфического оборудования и систем обработки информации Белорусского государственного технологического университета, доцент
М. С. Шмаков.

Романенко, Д. М.

Р 31 Администрирование информационных систем : тексты лекций для студентов специальности 1-40 01 02-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» / Д. М. Романенко. – Минск : БГТУ, 2014. – 121 с.

В текстах лекций дано понятие информационной системы, приведены основные задачи сетевого администрирования. Описаны особенности построения и использования RAID-массивов. Рассмотрены правила IP (статической, динамической) и символьной (DNS, Net Bios) адресации, приведены примеры планирования пространства доменных имен, раскрыты вопросы планирования и управления Active Directory, обеспечения безопасности. Подробно изложены протоколы маршрутизации, удаленного доступа и VPN, описаны основы удаленного администрирования на основе протоколов Telnet и SSH.

УДК 004.7(075.8)(0..34.2)
ББК 73я73

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

В настоящее время вычислительная техника является мощным средством ускорения научно-технического прогресса и находит все большее применение в различных отраслях человеческой деятельности. Это обстоятельство вызывает необходимость освоения вычислительной техники будущим инженером-программистом в объеме, позволяющем использовать ее на должном уровне при решении конкретных практических задач.

Целью дисциплины «Администрирование информационных систем» является обучение студентов общим методам создания, настройки и администрирования сетей, а также персональных компьютеров.

Задачей дисциплины является изучение студентами сетевых операционных систем на базе платформ Windows, а также методов управления информационными системами.

В рамках курса предполагается изучение базовых понятий, целей, задач сетевого администрирования, моделей межсетевого взаимодействия OSI/ISO, а также стека протоколов и моделиTCP/IP,методов настройки различных типов адресации в IP-сетях. Предполагается рассмотрение эффективных решений задач управления пользователями и ресурсами сети, а также приобретения необходимых навыков, освоение основных приемов и инструментов мониторинга компьютерной сети, овладение базовыми средствами обеспечения безопасности сети. В процессе изучения курса происходит воспитание творческого подхода к решению проблем, возникающих в процессе профессиональной деятельности специалиста.

Курс состоит из лекционной части и лабораторного практикума. В лекциях рассматриваются основные теоретические положения, необходимые для успешного освоения практических навыков и умений. Также приводится библиографический список дополнительной литературы по тематике курса.

Для успешного освоения материала желательно наличие базовых знаний по основам компьютерных сетей, хотя все необходимые сведения приводятся в лекциях или при описании лабораторных работ.

ТЕМА 1. ЗАДАЧИ И ЦЕЛИ СЕТЕВОГО АДМИНИСТРИРОВАНИЯ. ПОНЯТИЕ СЕТЕВЫХ ПРОТОКОЛОВ И СЛУЖБ

План

1. Цели и задачи администрирования информационных систем.

2. Модели межсетевого взаимодействия.

Данная тема рассчитана на одну лекцию.

1.1. Цели и задачи администрирования информационных систем

Информационная система – это взаимосвязанная совокупность информационных, технических, программных, математических, организационных и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи информации.

Современные информационные системы по своей природе всегда являются распределенными системами. Рабочие станции пользователей, серверы приложений, серверы баз данных и прочие сетевые узлы распределены по большой территории, при этом используются различные коммуникации, технологии, сетевые устройства, программное обеспечение. Главной задачей администрирования в данном случае будет являться обеспечение надежности, бесперебойности, безопасной работы всей системы с требуемым уровнем производительности. Информационная система обязательно базируется на компьютерной сети, при этом сети могут быть условно трех видов:

- локальные (LAN, Local Area Network);
- глобальные (WAN, Wide Area Network);
- городские (MAN, Metropolitan Area Network).

Однако с точки зрения администратора данное деление выполняется исходя не из принципа расстояния, а из скорости передачи информации. Наиболее быстрыми будут локальные сети, а глобальные наиболее медленными.

Вся сетевая инфраструктура строится из различных компонент, которые условно можно разнести по следующим уровням:

- кабельная система и средства коммуникаций;
- активное сетевое оборудование;
- сетевые протоколы;
- сетевые службы;
- сетевые приложения.

Каждый из этих уровней может состоять из различных подуровней и компонент.

Активное сетевое оборудование включает в себя такие виды устройств как повторители (репитеры), мосты, концентраторы, коммутаторы, маршрутизаторы. В корпоративной сети может быть использован богатый набор сетевых протоколов: TCP/IP, SPX/IPX, NetBEUI, AppleTalk и др.

Основу работы сети составляют так называемые сетевые службы (или сервисы). Базовый набор любой корпоративной сети состоит из следующих служб:

- службы сетевой инфраструктуры DNS, DHCP, WINS;
- службы файлов и печати;
- службы каталогов (например, Novell NDS, MS ActiveDirectory);
- службы обмена сообщениями;
- службы доступа к базам данных.

Взаимодействие между различными видами компьютерных систем осуществляется благодаря стандартизированным методам передачи данных, которые в основном базируются на моделях ISO/OSI, TCP/IP.

Основные цели и задачи сетевого администрирования:

1. Планирование сети. Несмотря на то, что планированием и монтажом больших сетей обычно занимаются специализированные компании-интеграторы, сетевому администратору часто приходится планировать определенные изменения в структуре сети: добавление новых рабочих мест, добавление или удаление сетевых протоколов, добавление или удаление сетевых служб, установка серверов, разбиение сети на сегменты и т.д. Данные работы должны быть тщательно спланированы, чтобы новые устройства, узлы или протоколы включались в сеть или исключались из нее без нарушения целостности сети, без снижения производительности и без нарушения инфраструктуры сетевых протоколов, служб и приложений.

2. Установка и настройка сетевых узлов (устройств активного сетевого оборудования, персональных компьютеров, серверов, средств

коммуникаций). Данные работы могут включать в себя замену сетевого адаптера в ПК с соответствующими настройками компьютера, перенос сетевого узла (ПК, сервера, активного оборудования) в другую подсеть с соответствующими изменениями сетевых параметров узла, добавление или замену сетевого принтера с соответствующей настройкой рабочих мест.

3. Установка и настройка сетевых протоколов. Данная задача включает в себя выполнение таких работ как планирование и настройка базовых сетевых протоколов корпоративной сети, тестирование работы сетевых протоколов, определение оптимальных конфигураций протоколов.

4. Установка и настройка сетевых служб. Корпоративная сеть может содержать большой набор сетевых служб. Кратко перечислим основные задачи их администрирования:

- установка и настройка служб сетевой инфраструктуры (службы DNS, DHCP, WINS, службы маршрутизации, удаленного доступа и виртуальных частных сетей);
- установка и настройка служб файлов и печати, которые в настоящее время составляют значительную часть всех сетевых служб;
- администрирование служб каталогов (Novell NDS, Microsoft ActiveDirectory), составляющих основу корпоративной системы безопасности и управления доступом к сетевым ресурсам;
- администрирование служб обмена сообщениями (системы электронной почты);
- администрирование служб доступа к базам данных.

5. Поиск неисправностей. Сетевой администратор должен уметь обнаруживать широкий спектр неисправностей – от неисправного сетевого адаптера на рабочей станции пользователя до сбоев отдельных портов коммутаторов и маршрутизаторов, а также неправильные настройки сетевых протоколов и служб.

6. Поиск узких мест сети и повышение эффективности работы сети. В задачи сетевого администрирования входит анализ работы сети и определение наиболее узких мест, требующих либо замены сетевого оборудования, либо модернизации рабочих мест, либо изменения конфигурации отдельных сегментов сети.

7. Мониторинг сетевых узлов. Мониторинг включает в себя наблюдение за функционированием сетевых узлов и корректностью выполнения возложенных на данные узлы функций.

8. Мониторинг сетевого трафика. Позволяет обнаружить и лик-

видировать различные виды проблем: высокую загруженность отдельных сетевых сегментов, чрезмерную загруженность отдельных сетевых устройств, сбои в работе сетевых адаптеров или портов сетевых устройств, нежелательную активность или атаки злоумышленников (распространение вирусов, атаки хакеров и др.).

9. Обеспечение защиты данных. Защита данных включает в себя большой набор различных задач:

- резервное копирование и восстановление данных;
- разработка и осуществление политик безопасности учетных записей пользователей и сетевых служб (требования к сложности паролей, частота смены паролей);
- построение защищенных коммуникаций (применение протокола IPSec, построение виртуальных частных сетей, защита беспроводных сетей);
- планирование, внедрение и обслуживание инфраструктуры открытых ключей (PKI).

1.2. Модели межсетевого взаимодействия (модель OSI, модель TCP/IP)

Модели межсетевого взаимодействия предназначены для формального и в то же время наглядного описания взаимодействия сетевых узлов между собой. В настоящее время получили наибольшее распространение и являются стандартами для описания межсетевого взаимодействия две сетевые модели: модель OSIи модель TCP/IP. Обе разбивают процесс взаимодействия сетевых узлов на несколько уровней, каждый конкретный уровень одного узла обменивается информацией с соответствующим уровнем другого узла.

Модели OSI и TCP/IP подробно рассмотрены в литературе [2, главы 3 и 4].

Контрольные вопросы

1. Дайте определение информационной системы.
2. Перечислите основные цели и задачи сетевого администрирования.
3. Опишите модель межсетевого взаимодействия OSI.
4. Опишите модель межсетевого взаимодействия TCP/IP.

ТЕМА 2. СЕТЕВЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ WINDOWSSERVER. ИНСТРУМЕНТЫ АДМИНИСТРИРОВАНИЯ

План

1. Понятие ролей сервера.

2. Особенности Windows Server 2008.

3. Инструменты администрирования компьютерных систем на базе Windows Server.

Данная тема рассчитана на одну лекцию.

2.1. Серверные ОС Windows

Установка и настройка серверной ОССильно зависит от поставленных задачч, которые необходимо решить администраторам. Типовые задачи объединяются в так называемые роли.

Все роли можно увидеть при запуске мастеров «Мастер настройки сервера» и «Управление данным сервером». К этим **ролям** относятся:

1. Файловый сервер (сервер, предоставляющий доступ к файлам и управляющий им); выбор этой роли позволит быстро настроить параметры квотирования и индексирования.

2. Сервер печати (сервер, организующий доступ к сетевым принтерам и управляющий очередями печати и драйверами принтеров); выбор этой роли позволит вам быстро настроить параметры принтеров и драйверов.

3. Сервер приложений (сервер, на котором выполняются Web-службы XML, Web-приложения и распределенные приложения; при назначении серверу этой роли на нем автоматически устанавливаются IIS, COM+ и Microsoft.NET Framework; при желании вы можете добавить к ним серверные расширения MicrosoftFrontPage, а также включить или выключить ASP.NET).

4. Почтовый сервер (сервер, на котором работают основные почтовые службы POP3 (Post Office Protocol 3) и SMTP (Simple Mail Transfer Protocol), благодаря чему почтовые POP3-клиенты домена могут отправлять и получать электронную почту); выбрав

эту роль, вы определяете домен по умолчанию для обмена почтой и создаете почтовые ящики.

5. Сервер терминалов (сервер, выполняющий задачи для клиентских компьютеров, которые работают в режиме терминальной службы); выбор этой роли приводит к установке служб терминалов, работающих в режиме сервера приложений.

6. Сервер удаленного доступа / сервер виртуальной частной сети (сервер, осуществляющий маршрутизацию сетевого трафика и управляющий телефонными соединениями и соединениями через виртуальные частные сети (virtual private network, VPN)); выбрав эту роль, вы запустите Мастер настройки сервера маршрутизации и удаленного доступа (Routing and Remote Access Server Setup Wizard); с помощью параметров маршрутизации и удаленного доступа вы можете разрешить только исходящие подключения, входящие и исходящие подключения или полностью запретить доступ извне.

7. Служба каталогов (контроллер домена Active Directory – сервер, на котором работают службы каталогов и располагается хранилище данных каталога); контроллеры домена также отвечают за вход в сеть и поиск в каталоге; при выборе этой роли на сервере будут установлены DNS и Active Directory.

8. Система доменных имен (сервер, на котором запущена служба DNS, разрешающая имена компьютеров в IP-адреса и наоборот); при выборе этой роли на сервере будет установлена DNS и запущен Мастер настройки DNS-сервера.

9. Сервер протокола динамической настройки узлов (сервер, на котором запущена служба DHCP (Dynamic Host Configuration Protocol), позволяющая автоматизировать назначение IP-адресов узлам сети); при выборе этой роли на сервере будет установлена служба DHCP и запущен Мастер создания области.

10. Сервер Windows Internet Naming Service (сервер, на котором запущена служба WINS, разрешающая имена NetBIOS в IP-адреса и наоборот); выбор этой роли приводит к установке службы WINS.

11. Сервер потокового мультимедиа-вещания (сервер, предоставляющий мультимедийные потоки другим системам сети или Интернета); выбор этой роли приводит к установке служб Windows Media; эта роль поддерживается только в версиях Standard Edition и Enterprise Edition.

2.2. Основные улучшения Windows Server 2008

К основным улучшениям, реализованным в Windows Server 2008 по сравнению с предыдущими версиями, относятся:

1. Active Directory Federation Service – позволяет устанавливать соответствующие отношения между объединяющимися доменами.

2. Read-Only domain Controllers – контроллеры предполагается использовать в средах, где невозможно гарантировать физическую сохранность контроллера.

3. Server Core Installation (установка серверного ядра) –в данном режиме можно использовать DHCP Server, Print Server, Streamingmedia Services, File Server, Active Directory Domain Services. Графический интерфейс отсутствует, присутствует только командная строка.

4. Windows Bit Locker Drive Encryption, позволяет зашифровать все жесткие диски на сервере

5. Новый и улучшенный сервер IIS 7.0, улучшения сводятся к изменению интерфейса. Предполагается, что новый интерфейс позволяет решить исходные задачи быстрее, улучшение сделано в стороне безопасности, оценки безопасности системы

6. Быстрое развертывание клиентской ОС с помощью WDS. Все **компоненты служб Windows Deployment Services** подразделяются на три следующие категории:

a) **серверные компоненты**. Обеспечивается загрузка клиентского компьютера и установка на него ОС. Также входит хранилище общих папок и образов

b) **клиентские компоненты**. В состав этой группы входит графический пользовательский интерфейс среды представит установки Windows(Windows Pre-Installation Environment, Windows PE). Когда пользователь выбирает образ ОС, клиентские компоненты взаимодействуют с серверными, чтобы выполнить установку этого образа.

c) **управляющие компоненты**. Инструменты, которые используются для управления серверными образами ОС, учетными записями клиентских компонентов.

7. Безопасность в сети с помощью Network Access Protection может быть помещена в каком-либо заданном месте и будет контролировать выполнение некоторых требований, прежде чем произойдет соединение с сервером. Если клиент не удовлетворен требованиями, он может быть помещен в карантин, либо ему может быть отказано в доступе, при этом на стороне клиента могут быть установлены дополнения и обновления. Служба NAP может таким образом работать

не только с локальным компьютером, но и компьютером, удаленно присоединенным к серверу.

8. Улучшения терминальных служб Window Server, Windows Server Virtualization – новое свойство ОС, позволяет виртуализировать любую ОС на одном сервере. На данный момент возможность виртуализации несколько уступает Virtual Box, однако использование данной службы позволяет избежать проблемы взаимодействия виртуальных ОС.

2.3. Инструменты администрирования

Операционная система Windows Server 2003 предоставляет системному администратору широкий набор инструментов для решения задач управления. Основными из них являются следующие инструменты:

- консоль управления (Microsoft Management Console, MMC);
- мастера (Wizards);
- утилиты командной строки.

Консоль управления MMC представляет собой унифицированную среду для выполнения административных задач. Администратор, имея в распоряжении такую среду, может помещать в нее одну или несколько утилит, называемых *оснастками* (snap-in), для решения текущей проблемы. Консоль управления позволяет одинаково отображать любые оснастки и использовать для управления ими похожие приемы. Таким образом, смысл применения консоли управления в том, чтобы сделать среду выполнения административных утилит единообразной и удобной.

С той же целью в Windows Server 2003 применяются *мастера*. Мастер представляет собой программу, которая проводит администратора по всем этапам решения какой-либо задачи. На каждом этапе возможен выбор одного или нескольких способов решения или параметров настройки. Часто мастера предоставляют возможность выбора параметров по умолчанию. Использование мастеров позволяет сократить время установки и настройки компонентов операционной системы или время решения другой административной задачи. Кроме того, параметры по умолчанию чаще всего обеспечивают вполне работоспособный режим, хотя, возможно, и не самый эффективный.

Утилиты командной строки являются самыми старыми инструментами администрирования, ведущими свою историю от первых операционных систем без графического интерфейса. В то время аль-

тернативы утилитам командной строки не было. Сегодня большинство задач управления можно решить без использования утилит, однако многие администраторы считают, что утилиты командной строки удобнее графического интерфейса. Кроме того, такой вид утилит, как утилиты диагностики стека протоколов TCP/IP, не имеют стандартного графического аналога (эти утилиты рассматриваются во второй лекции).

Большинство административных задач возможно решить, используя любой из представленных инструментов (консоль управления, мастер или утилиту командной строки). Выбор инструмента обусловливается, в основном, личными предпочтениями системного администратора.

Контрольные вопросы

1. Какова основная цель сетевого администрирования?
2. Чем отличаются понятия сетевое администрирование и системное администрирование?
3. Назовите основные виды задач сетевого администрирования. Приведите примеры конкретных задач на каждый вид.
4. Чем отличаются операционные системы Microsoft Windows Server 2003 и 2008?
5. Что такое роль сервера?
6. Что такое оснастка (snap-in)?
7. Назовите основные инструменты администрирования. Приведите примеры.

ТЕМА 3. RAID-МАССИВЫ

План

1. **Понятие RAID-массива. Основные принципы.**
2. **Одиночные RAID-массивы.**
3. **Составные RAID-массивы.**

Данная тема рассчитана на две лекции (лекция 3, 4).

3.1. Понятие RAID-массива. Основные принципы

В переводе с английского «RAID» (Redundant Arrays of Inexpensive Disks) означает избыточный массив независимых дисков.

Первоначальное предназначение массива – это создание на базе нескольких массивов одного диска с большим объемом и увеличением скорости доступа. Гораздо позднее к двум первым основным целям добавилась цель, связанная с сохранением данных в случае отказа части оборудования. Именно это сделалоRAID-массивы востребованными, первоначально в военной промышленности, а затем и в различных компьютерных системах. Однако достаточно сложно при построении RAID-массива найти оптимальное решение по надежности, скорости, емкости и цене.

В основе теории RAID лежат пять основных принципов. Это *массив* (Array), *зеркалирование* (Mirroring), *дуплекс* (Duplexing), *чередование* (Striping) и *четность* (Parity).

Массивом называют несколько накопителей, которые централизованно настраиваются, форматируются и управляются. Логический массив – это уже более высокий уровень представления, на котором не учитываются физические характеристики системы. Соответственно, логические диски могут по количеству и объему не совпадать с физическими. Но лучше все-таки соблюдать соответствие: физический диск – логический диск. Наконец, для операционной системы весь массив является одним большим диском.

Зеркалирование – технология, позволяющая повысить надежность системы(рис. 3.1). В RAID-массиве с зеркалированием все данные одновременно пишутся не на один, а на два жестких диска. То есть создается «зеркало» данных. При выходе из строя одного из дисков вся информация остается сохраненной на втором. За такую стопроцент-

ную защиту приходится дорого платить: один винчестер работает просто так, не увеличивая доступную емкость ни на мегабайт. При этом нет никакого выигрыша в производительности.

Дуплекс – развитие идеи зеркалирования (рис. 3.2). В этом случае так же высок уровень надежности и требуется в два раза больше жестких дисков, но появляются дополнительные затраты: для повышения надежности в систему устанавливаются два независимых RAID-контроллера. Выход из строя одного диска или контроллера не сказывается на работоспособности системы. Столь дорогое решение используется только во внешних RAID-массивах, предназначенных для ответственных приложений.

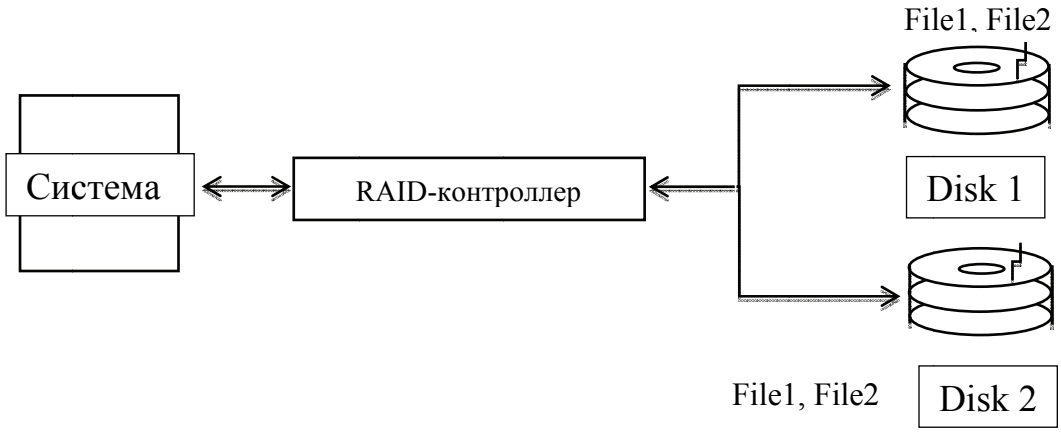


Рис. 3.1. Принципиальная схема технологии «зеркалирования»

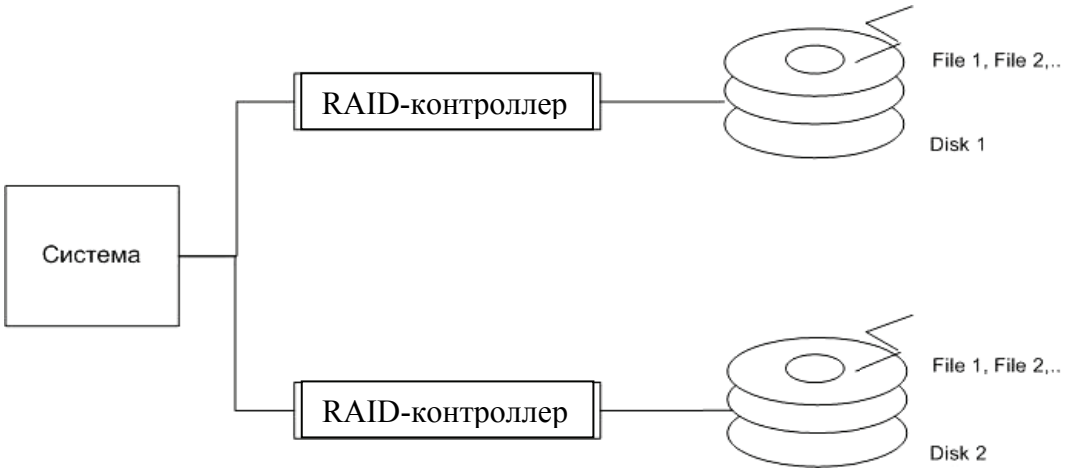


Рис. 3.2. Принципиальная схема технологии «дуплекс»

Чередование. Согласно данной технологии запись ведется на несколько жестких дисков, при этом записываемый файл разбивается на

части определенного размера и посылается на несколько накопителей, в таком фрагментированном виде файлы и хранятся (рис. 3.3).

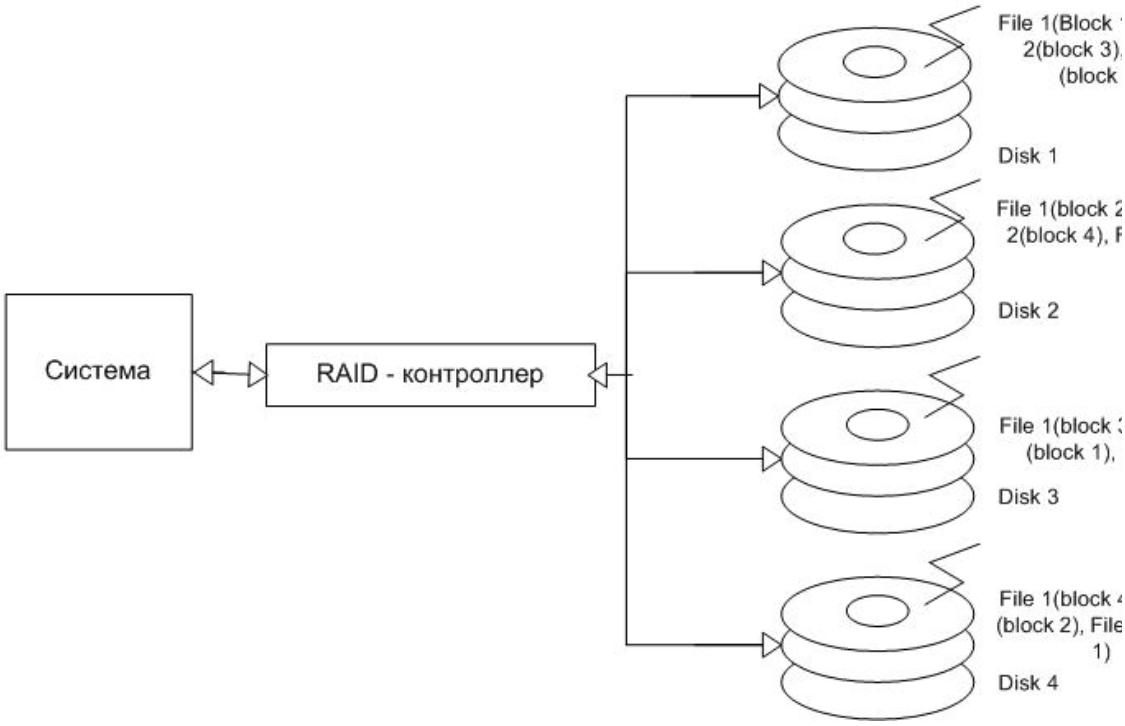


Рис. 3.3. Принципиальная схема технологии «чередование»

Данная технология позволяет увеличить линейную скорость записи чтения. Основной проблемой является надежность – вывод из строя любого накопителя приводит к потере информации.

Четность является альтернативным решением, которое соединяет в себе достоинства и недостатки зеркалирования и чередования, используется тот же принцип, что и в избыточных кодах, основанных на свертке по модулю 2. Согласно данной технологии используется $n + 1$ накопитель, при этом на n накопителей записывается информация в виде отдельных блоков (как в чередовании). На $n + 1$диске хранится так называемый экстраблок, который является контрольной суммой соответствующих n блоков.

Плюсы четности очевидны. За счет использования чередования повышается скорость работы. За счет использования экстраблоков повышается надежность, но при этом «нерабочий» объем массива достаточно мал и одинаков при любом количестве дисков – составляет емкость одного диска, то есть при 5 дисках в массиве «теряется» всего 20% емкости.

Основным недостатком является необходимость выполнения вычислений налету. В наилучшем варианте вычисления должны выполнятся RAID-контроллером.

3.2. Одиночные RAID-массивы

RAID-массив принято обозначать цифрами. Существуют одиночные RAID-массивы и комбинированные (составные).

RAID 0 – это простой массив, использующий чередование. Вся информация разбивается на блоки фиксированной длины. При наличии двух-четырех дисков, RAID 0 дает ощутимый выигрыш в скорости передачи данных, но совершенно не обеспечивает надежность. Для его построения подойдет любой дешевый и даже программный RAID-контроллер. Данный RAID-массив целесообразно использовать при необходимости получения максимума производительности при минимальных затратах (рис.3.4).

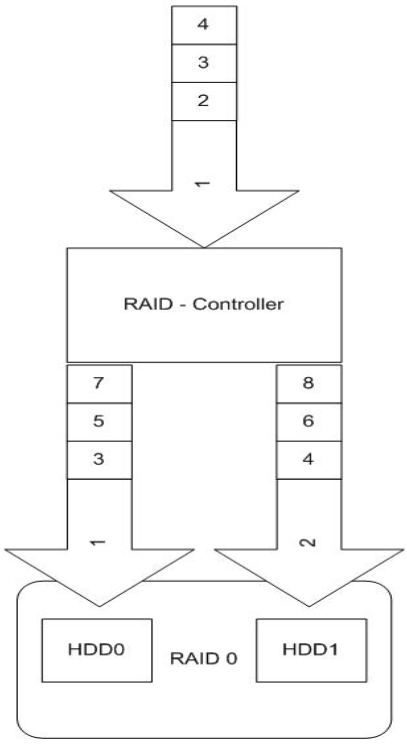


Рис. 3.4. Принципиальная схема RAID 0

RAID 1. Данный RAID-массив повторяет идею зеркалирования со всеми ее достоинствами и недостатками. На два жестких диска пишутся две одинаковые копии данных. При этом можно ис-

пользовать дешевый RAID-контроллер или даже его программную реализацию (рис.3.5).

RAID 1 позволяет надежно защитить данные и обеспечить работу системы даже при поломке одного из дисков. Вот почему он получил широкое распространение среди пользователей, желающих защитить от потери личные данные. Выигрыш в скорости при использовании RAID 1 может быть достигнут лишь при считывании данных в многозадачном режиме.

RAID 2. Данный RAID-массив использует технологии чередования и четности в виде кода Хеминга. Теоретически массив должен быть хорошим по надежности и емкости, но его реализация требует использования специальных дорогостоящих контроллеров. В силу этого практического применения он не нашел.

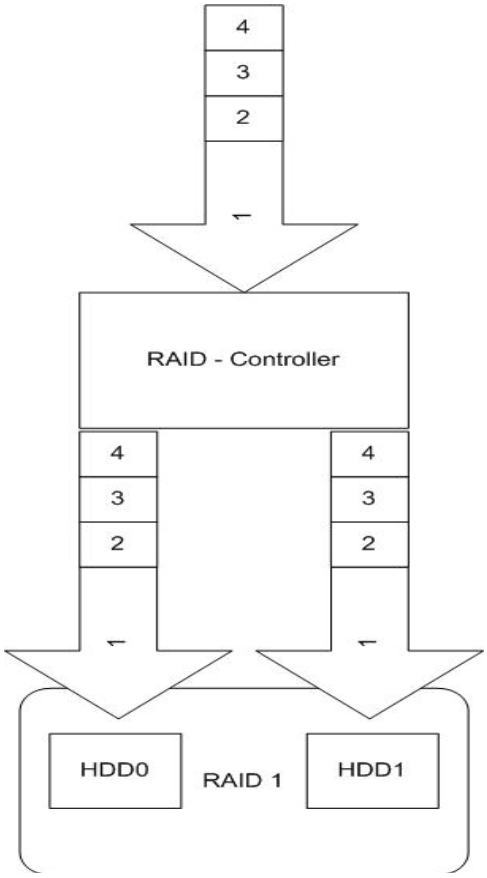


Рис. 3.5. Принципиальная схема RAID 1

RAID 3 и RAID 4. В данном случае хранение осуществляется на одном диске, применяется соединение чередования и четности (рис. 3.6).

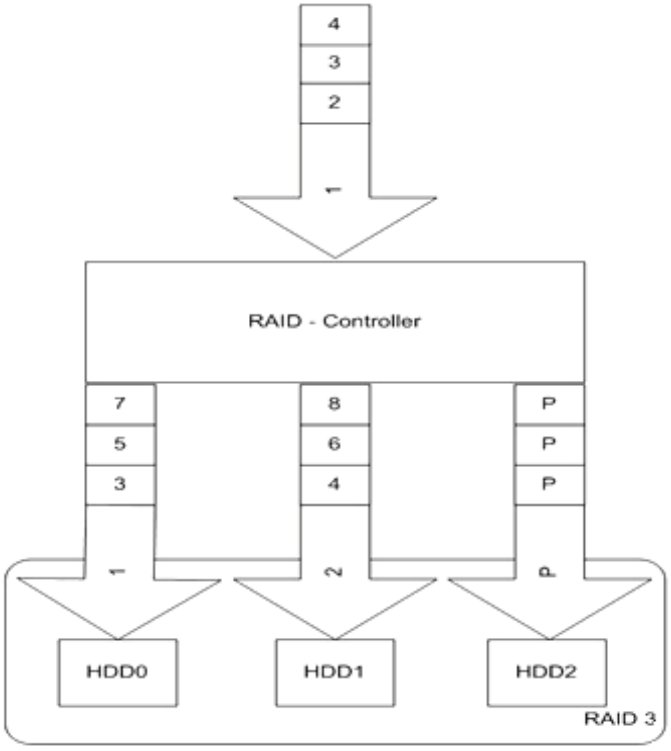


Рис. 3.6. Принципиальная схема RAID 3 и RAID 4

В RAID 3 блоки данных имеют длину меньше 1024 байт. Наиболее слабое место – низкая скорость случайной записи. Достоинством же является работа при отказе одного из дисков. RAID4 отличается только размером блока данных.

RAID 5. Наиболее распространенный массив, так же характеризующийся применением идей чередования и четности, но контрольные суммы хранятся не на одном диске, а разбрасываются по всем. Главный принцип распределения электроблоков заключается в следующем: они не должны располагаться на том же диске, с которого была закодирована информация (рис. 3.7).

Характеризуется высокой скоростью записи с достаточно высокой надежностью, при этом информационная емкость RAID 5 рассматривается как количество дисков минус единица, умноженное на объем минимального диска.

Недостатки RAID 5 проявляются при выходе из строя одного из дисков: весь том переходит в критический режим (degrade), все операции записи и чтения сопровождаются дополнительными манипуляциями, резко падает производительность. При этом уровень надежности снижается до надежности RAID 0 с соответсвующим количеством дисков (то есть в *n* раз ниже надежности одиночного диска).

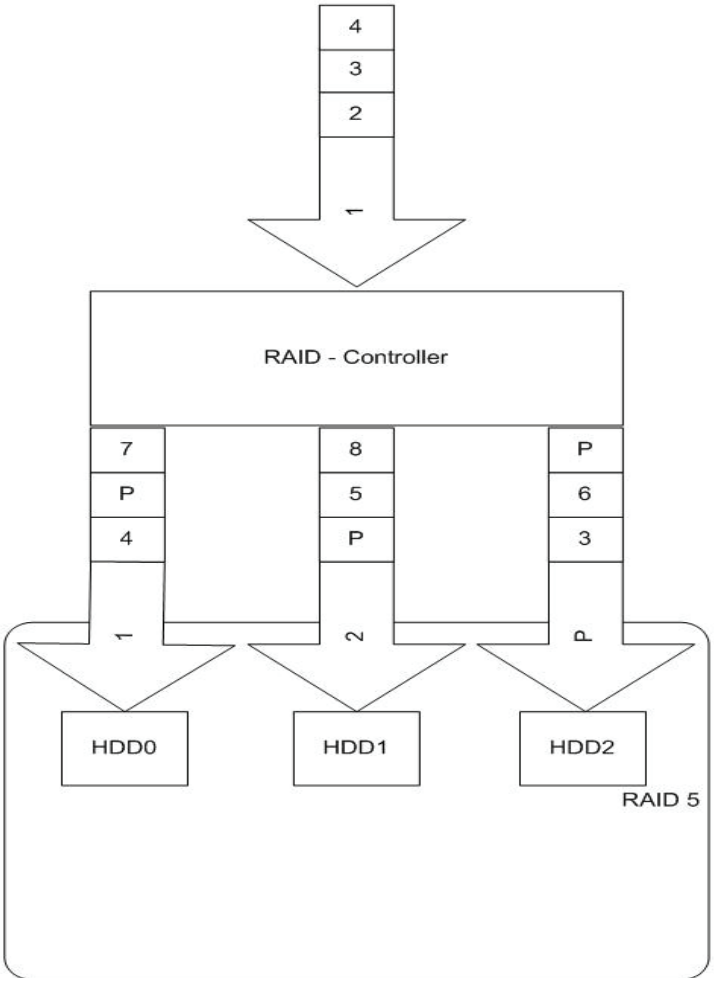


Рис. 3.7. Принципиальная схема RAID 5

Если до полного восстановления массива произойдет выход из строя или возникнет невосстановимая ошибка чтения хотя бы еще на одном диске, то массив разрушается, и данные на нем восстановлению обычными методами не подлежат. Минимальное количество используемых дисков равно трем.

С томом RAID 5 можно использовать диск Hot Spare. Основное время дополнительный диск простаивает, но при выходе из строя одного из дисков массива, его восстановление начинается немедленно с использованием spare-диска. При использовании одного тома RAID5 данная конфигурация дисков является расточительной, эффективнее использовать RAID6. Целесообразность использования spare-диска проявляется в системе из нескольких томов RAID5, в которой spare-диск проинициализирован для каждого из томов RAID5 и может быть

использован в случае необходимости для немедленного восстановления любого из томов.

Таким образом, надежность и скорость работы такой системы оказываются очень даже высокими, и при восстановлении информации всю работу на себя берет RAID-контроллер, так что операция проходит довольно быстро.

RAID 6. Для некоторых особо критичных приложений требуется повышенная надежность. В RAID 6 используются все те же технологии чередования и четности. Но контрольная сумма вычисляется два раза и копируется на два разных диска. В итоге данные окажутся потерянными только в случае выхода из строя сразу трех жестких дисков. По сравнению с RAID 5, это более дорогое и медленное решение, которое может показать себя разве что при случайном чтении. На практике RAID 6 почти не используется, так как выход из строя сразу двух дисков – слишком редкий случай, а повысить надежность можно другими способами.

3.3. Составные RAID-массивы

У основных уровней RAID есть свои достоинства и недостатки. Для объединения достоинств различных RAID-массивов и нивелирования недостатков были предложены составные RAID-массивы. Составной RAID-массив – это чаще всего сочетание быстрого RAID 0 с надежным RAID 1, 3 или 5. Итоговый массив действительно обладает улучшенными характеристиками, но и «платить» за это приходится повышением стоимости и сложностью решения.

Составные RAID-массивы строятся по следующему принципу: сначала диски разделяются на наборы (сеты, set), затем на основе сетов строятся одиночные (простые) массивы, а в завершении все объединяется в составной массив. Запись типа X+Y означает, что сначала диски объединяются в уровни X, а затем несколько RAIDX-массивов объединяются в RAID-массивы уровня Y.

RAID 1+0 (0+1). RAID 0+1 часто называют зеркалом страйпов (рис. 3.8), а RAID 0+1 страйпом зеркал.RAID 0+1 – обладает повышенной надежностью и высокой скоростью, поддерживается даже дешевыми RAID-контроллерами и почти всеми материнскими платами, но по надежности RAID 1+0 считается лучше. Говорят, что при построении RAID-массивов на 10 дисках, система может оставаться работоспособной при отказе 5 жестких дисков. Основным недостатком обоих массивов является достаточно низкий процент использования емкости накопителя.

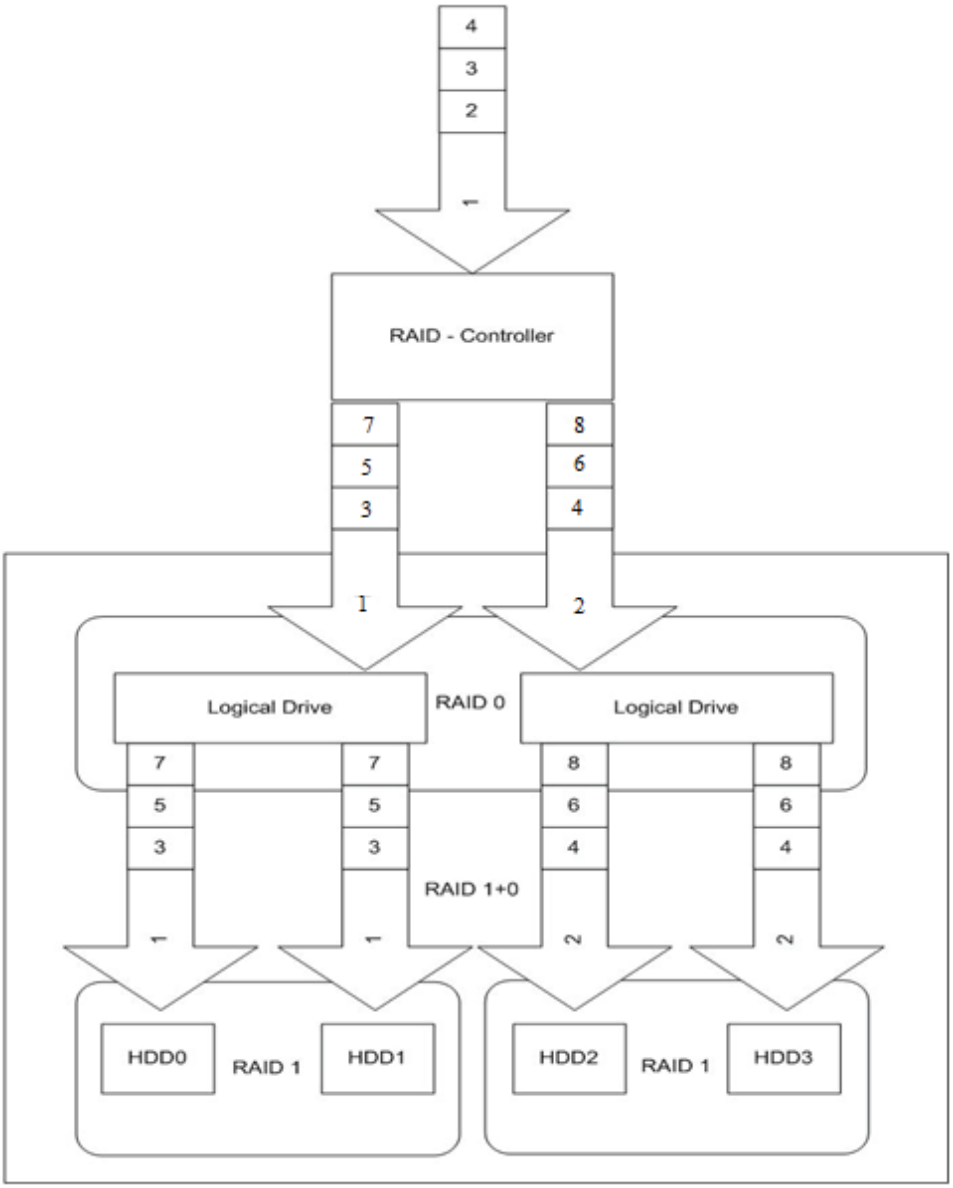


Рис. 3.8. Принципиальная схема RAID 1+0

RAID 3+0 (0+3).RAID 0+3 является массивом с выделенной четностью над чередованием, в котором данные блоками разбиваются и пишутся на массивы RAID 0.

RAID 3+0 является страйпом из массивов RAID 3. Обладает достаточно высокой скоростью передачи данных, характеризуется неплохой отказоустойчивостью. Данные сначала разбиваются на блоки, как в RAID 0, и попадают в сеты. Там они снова делятся на блоки, считается их четность, блоки пишутся на все диски кроме одного, на последний диск пи-

шется значение четности. В данном случае из строя может выйти один из дисков или каждый RAID 3. Считается, что RAID 3+0 с точки зрения надежности лучше, чем RAID 0+3. Достоинства данных массивов заключаются в высоком проценте использования емкости дисков, а так же достаточно высокой скорости чтения данных. Основной недостаток – сложность реализации RAID-контроллера и цена.

RAID 5+0 (0+5). RAID 0+5 представляет собой набор страйпов, на основе которых построен RAID 5 (рис. 3.9), такая комбинация используется крайне редко, так как не дает выигрыша ни в скорости, ни в надежности. Широкое распространение получил RAID 5+0. Чаще всего такой массив выстраивается по следующему принципу: 2RAID-массива уровня 5 объединяются в страйп. Такая комбинация позволяет получить высокую производительность при работе с файлами малого размера, поэтому RAID 5+0 часто используется на Web-серверах.

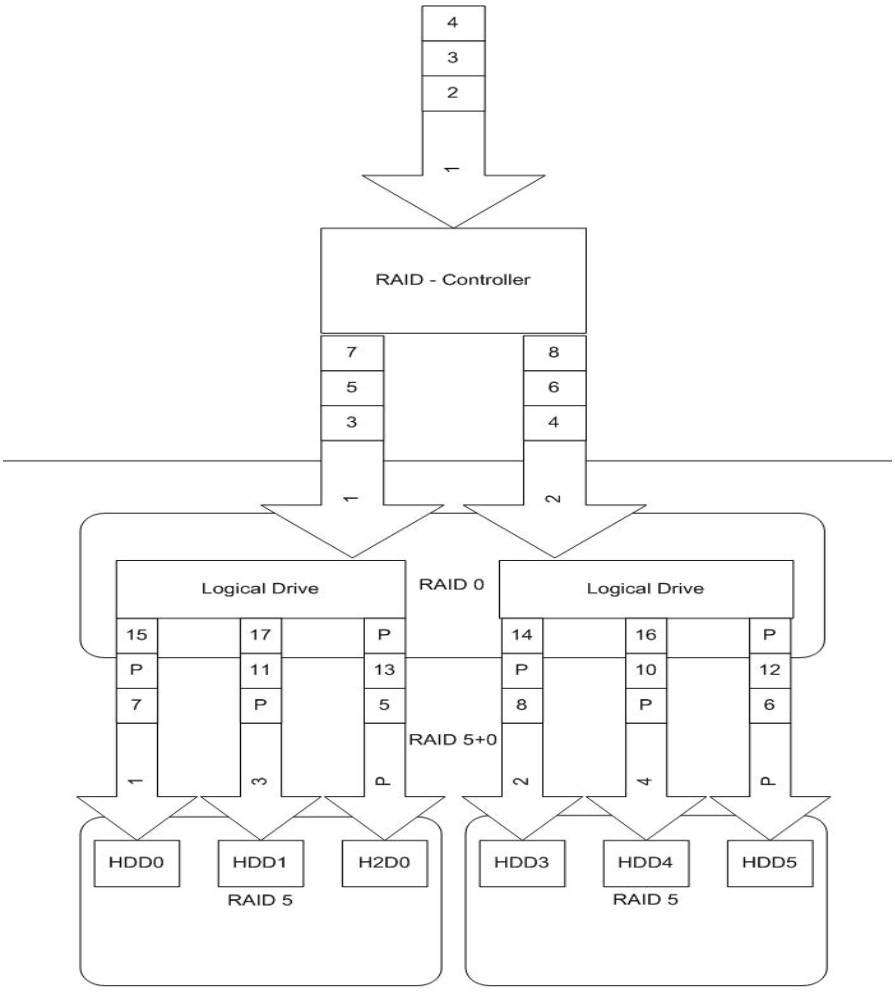


Рис. 3.9. Принципиальная схема RAID 5+0

RAID 1+5 (5+1). Этот уровень построен на сочетании зеркалирования и дуплекса и чередования с четностью. Основная цель – построение массива, значительно повышающего надежность. Массив 1+5 продолжает работать при отказе трех накопителей, а 5+1 даже при потере пяти из восьми накопителей. Основной недостаток – низкое использование емкости дисков и общая дороговизна. Чаще всего используются 5+1, при этом два аппаратных RAID-контроллера уровня 5 зеркалируются на программном уровне.

RAID 6+0.Фактически, это страйп из RAID 6, но так как RAID 6 особого применения не нашел, а также достаточно эффективным является RAID 0+5, RAID 6+0 распространения не получил.

RAID 10+0 (1+0+0). RAID 100, также пишущийся как RAID 10+0, является страйпом из RAID 10. По своей сути он схож с более широким массивом RAID 10, где используется вдвое больше дисков. Но именно такой «трехэтажной» структуре есть свое объяснение. Чаще всего RAID 10 делают аппаратным, то есть силами контроллера, а уже страйп из них делают программно. К такой уловке прибегают, чтобы избежать проблемы ограничений по масштабируемости контроллеров. Программный же RAID 0 позволяет создать его на базе двух контроллеров, каждый из которых держит на борту RAID 10.

JBOD (Just Bunch of Disks).Не повышает ни быстродействия, ни надежности, но позволяет для работы использовать доступное пространство жестких дисков. В случае выхода из строя одного из жестких дисков, информация на другом не повреждается (рис. 3.10).

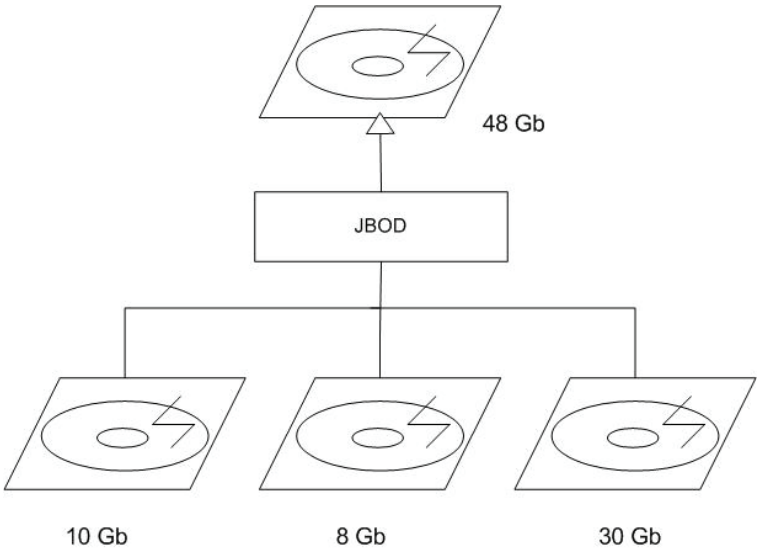


Рис. 3.10. Принципиальная схема RAID JBOD

MATRIX RAID. Эта технология реализуется фирмой Intel начиная с чипсетов ICH6R. Позволяет на некотором количестве дисков организовать один или несколько массивов уровня RAID 1, или 5, или 0. Не повышает ни быстродействия, ни надежности, но позволяет использовать все доступное пространство жестких дисков. В случае выхода из строя одного из жестких дисков, информация на другом не повреждается.

Контрольные вопросы

1. Что такое RAID-массив?
2. Опишите основные технологии, используемые при построении RAID-массивов. Приведите примеры.
3. Опишите RAID 0. Назовите основные достоинства и недостатки.
4. Опишите RAID 1. Назовите основные достоинства и недостатки.
5. Опишите RAID 2. Назовите основные достоинства и недостатки.
6. Опишите RAID 3 и 4. Назовите основные достоинства и недостатки.
7. Опишите RAID 5. Назовите основные достоинства и недостатки.
8. Опишите RAID 0+1 и 1+0. Назовите основные достоинства и недостатки.
9. Опишите RAID 3+0 и 0+3. Назовите основные достоинства и недостатки.
10. Опишите RAID 5+0 и 0+5. Назовите основные достоинства и недостатки.
11. Опишите RAID 5+1 и 1+5. Назовите основные достоинства и недостатки.
12. Опишите RAID 1+1+0. Назовите основные достоинства и недостатки.
13. Что такое JBOD RAID. Каково его назначение?
14. Что такое Matrix RAID. Каково его назначение?

ТЕМА 4. IP-АДРЕСАЦИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

План

1. **Представление IPv4-адреса.**
2. **Использование масок в IPv4.**
3. **Архитектура адресации IPv6.**
4. **Модель адресации IPv6.**
5. **Предоставление адресов.**
6. **Unicast адреса.**
7. **Anycast адреса.**
8. **Multicast адреса.**

Данная тема рассчитана на четыре лекции (лекции 5–8).

4.1. Протокол IPv4

4.1.1. Представление IPv4-адреса

Адрес IP представляет собой 32-разрядное двоичное число, разделенное на группы по 8 бит, называемые **октетами**. Например, 00010001 11101111 00101111 01011110.

Обычно IP-адреса записываются в виде четырех десятичных октетов и разделяются точками. Таким образом, приведенный выше IP-адрес можно записать в следующей форме: 17.239.47.94.

Следует заметить, что максимальное значение октета равно 11111111_2 (двоичная система счисления), что соответствует в десятичной системе 255_{10}. Поэтому IP-адреса, в которых хотя бы один октет превышает это число, являются недействительными. Пример: 172.16.123.1 – действительный адрес, а 172.16.123.256 – несуществующий, поскольку 256 выходит за пределы допустимого диапазона: от 0 до 255.

IP-адрес состоит из двух логических частей – ***номера подсети*** (IDподсети) и ***номера узла*** (ID хоста) в этой подсети. При передаче пакета из одной подсети в другую используется ID подсети. Когда пакет попал в подсеть назначения, ID хоста указывает на конкретный узел в рамках этой подсети.

Чтобы записать ID подсети, в поле номера узла в IP-адресе ставят нули. Чтобы записать ID хоста, в поле номера подсети ставят нули. Например, если в IP-адресе 172.16.123.1 первые два байта отводятся под номер подсети, остальные два байта – под номер узла, то номера записываются следующим образом: ID подсети 172.16.0.0; ID хоста 0.0.123.1.

По числу разрядов, отводимых для представления номера узла (или номера подсети), можно определить общее количество узлов (или подсетей) по простому правилу: если число разрядов для представления номера узла равно *N*, то общее количество узлов равно $2^N - 2$. Два узла вычитаются вследствие того, что адреса со всеми разрядами, равными нулям или единицам, являются особыми и используются в специальных целях.

Например, если под номер узла в некоторой подсети отводится два байта (16 бит), то общее количество узлов в такой подсети равно $2^{16} - 2 = 65\,534$ узла.

Для определения того, какая часть IP-адреса отвечает за ID подсети, а какая за ID хоста, применяются два способа: с помощью классов и с помощью масок. В настоящее время используется второй метод.

***Общее правило**:* под ID подсети отводятся первые несколько бит IP-адреса, а оставшиеся биты обозначают ID хоста.

4.1.2. Использование масок в IPv4

Маска подсети (subnet mask) – это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети.

Маска подсети записывается либо в виде, аналогичном записи IP-адреса, например 255.255.255.0, либо совместно с IP-адресом с помощью указания числа единичных разрядов в записи маски, например 192.168.1.1/24, т. е. в маске содержится 24 единицы (255.255.255.0).

Пример 1. Пусть задан IP-адрес 17.239.47.94, маска подсети 255.255.0.0 (другая форма записи: 17.239.47.94/16).

Требуется определить ID подсети и ID хоста в обеих схемах адресации.

1) *Адресация с использованием классов*. Двоичная запись IP-адреса имеет вид:

00010001.11101111.00101111.01011110.

Так как первый бит равен нулю, адрес относится к *классу А*. Следовательно, первый байт отвечает за ID подсети, остальные три байта – за ID хоста:

ID подсети: 17.0.0.0. ID хоста: 0.239.47.94.

2) *Адресация с использованием масок*. Запишем IP-адрес и маску подсети в двоичном виде:

IP-address: 17.239.47.94 = 00010001.11101111.00101111.01011110,
Subnetmask: 255.255.0.0 = 11111111.11111111.00000000.00000000.

Вспомнив определение маски подсети, можно интерпретировать номер подсети как те биты, которые в маске равны 1, т. е. первые два байта. Оставшаяся часть IP-адреса будет номером узла в данной подсети.

ID подсети: 17.239.0.0. ID хоста: 0.0.47.94.

Номер подсети можно получить другим способом, применив к IP-адресу и маске операцию логического умножения или *конъюнкции* (AND):

AND 00010001. 11101111. 00101111. 01011110,
11111111. 11111111. 00000000. 00000000 .
00010001. 11101111. 00000000. 00000000
17 239 0 0

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8.

Пример 2. Задан IP-адрес 192.168.89.16, маска подсети – 255.255.192.0 (другая форма записи: 192.168.89.16/18).

Требуется определить ID подсети и ID хоста. Воспользуемся операцией AND:

IP-address: 192.168.89.16 = AND11000000.10101000.01011001.00010000
Subnet mask: 255.255.0.0 = 11111111.11111111.11000000.00000000.
Subnet ID: 11000000.10101000.01000000.00000000
192 168 64 0

Чтобы получить номер узла, нужно в битах, отвечающих за номер подсети, поставить нули:

Host ID: 00000000.00000000.00011001.00010000 = 0.0.25.16.

Ответ: ID подсети = 192.168.64.0, ID хоста = 0.0.25.16.

Для масок существует важное правило: разрывы в последовательности единиц или нулей недопустимы.

Например, не существует маски подсети, имеющей следующий вид:

11111111.11110111.00000000.00001000 (255.247.0.8),

так как последовательности единиц и нулей не являются непрерывными.

С помощью масок администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. Допустим, организации выделена сеть класса *B*: 160.95.0.0 (рис. 4.1).

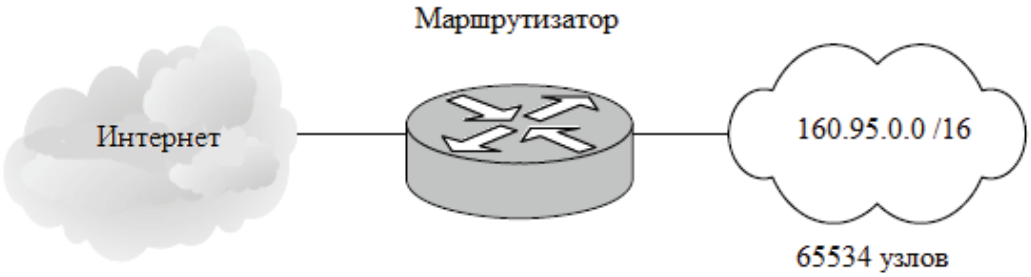


Рис. 4.1. Сеть класса B до деления на подсети

В такой сети может находиться до 65 534 узлов. Однако организации требуется три независимые сети с числом узлов в каждой не более 254. В этой ситуации можно применить деление на подсети с помощью масок. Например, при использовании маски 255.255.255.0 третий байт адреса будет определять номер внутренней подсети, а четвертый байт – номер узла (рис. 4.2).

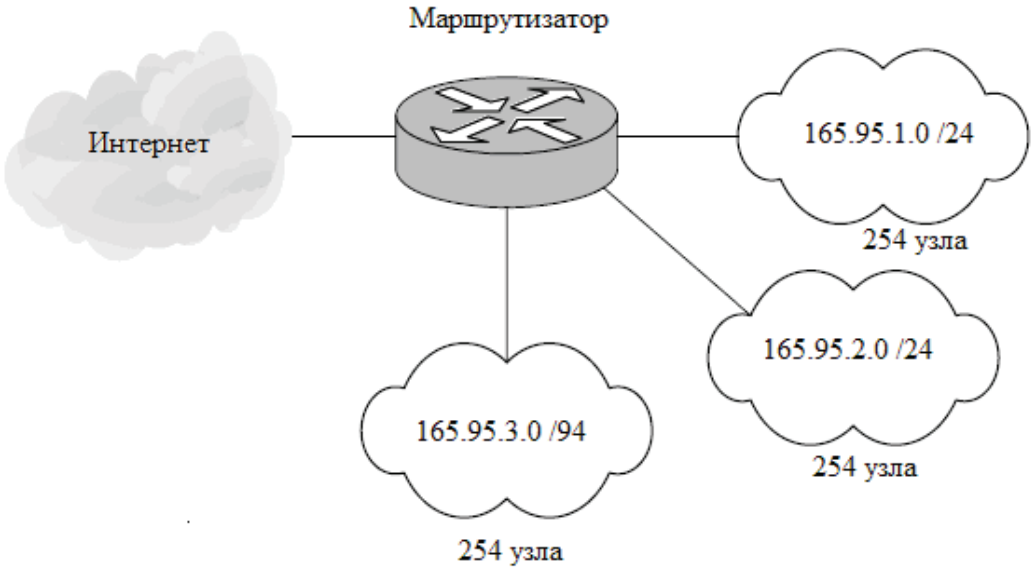


Рис. 4.2. Сеть класса B после деления на подсети

Маршрутизаторы во внешней сети (Интернет) ничего «не знают» о делении сети 160.95.0.0 на подсети, все пакеты направляются на маршрутизатор организации, который переправляет их в требуемую внутреннюю подсеть.

Более подробно протокол IPv4, а также особенности его использования и конфигурирования представлены в литературе [2, глава 5]

4.2. Протокол IPv6

Основная проблема протокола Ipv4 – это дефицит адресов в сети Интернет. Использование масок явилось временным решением данной проблемы, так как адресное пространство протокола IP не увеличивалось, а лишь более экономно использовалось. Для принципиального решения проблемы требуется существенное увеличение количества IP-адресов.

Для преодоления ограничений IPv4 был разработан протокол IP 6-й версии – IPv6 (RFC 2373, 2460).

Протокол IPv6 имеет следующие основные особенности:

- длина адреса – 128 бит, такая длина обеспечивает адресное пространство 2^{128}, или примерно $3 \cdot 4 \cdot 10^{38}$ адресов. Такое количество адресов позволит присваивать в обозримом будущем уникальные IP-адреса любым устройствам;
- автоматическая конфигурация. Протокол IPv6 предоставляет средства автоматической настройки IP-адреса и других сетевых параметров даже при отсутствии таких служб, как DHCP;
- встроенная безопасность. Для передачи данных является обязательным использование протокола защищенной передачи IPsec.

Протокол IPv4 может использовать IPsec, но не обязан этого делать.

В настоящее время многие производители сетевого оборудования включают поддержку протокола IPv6 в свои продукты, однако преобладающим остается протокол IPv4. Связано это с тем, что IPv6 обратно несовместим с IPv4, и процесс перехода сопряжен с определенными трудностями.

4.2.1. Архитектура адресации IPv6

Существует три типа адресов:

unicast: идентификатор одиночного интерфейса. Пакет, посланный по юникастному адресу, доставляется интерфейсу, указанному в адресе.

anycast: идентификатор набора интерфейсов (принадлежащих разным узлам). Пакет, посланный по эникастному адресу, доставляется одному из интерфейсов, указанному в адресе (ближайший, в соответствии с мерой, определенной протоколом маршрутизации).

multicast: идентификатор набора интерфейсов (обычно принадлежащих разным узлам). Пакет, посланный по мультикаст-адресу, доставляется всем интерфейсам, заданным этим адресом.

В IPv6 не существует широковещательных адресов, их функции переданы мультикастинг-адресам.

В IPv6, все нули и все единицы являются допустимыми кодами для любых полей, если не оговорено исключение.

Интерфейс-это средство подключения узла к каналу.

Модель адресации.

1. IPv6-адреса всех типов ассоциируются с интерфейсами, а не с узлами. Так как каждый интерфейс принадлежит только одному узлу, юникастный адрес интерфейса может идентифицировать узел.

2. IPv6 юникастный адрес соотносится только с одним интерфейсом. Одному интерфейсу могут соответствовать много IPv6-адресов различного типа (юникастные, эникастные и мультикстные). Существует два исключения из этого правила:

– одиночный адрес может приписываться нескольким физическим интерфейсам, если приложение рассматривает эти несколько интерфейсов как единое целое при представлении его на уровне Интернет;

– маршрутизаторы могут иметь ненумерованные интерфейсы (например, интерфейсу не присваивается никакого IPv6-адреса) для соединений точка-точка, чтобы исключить необходимость вручную конфигурировать и объявлять эти адреса. Адреса не нужны для соединений точка-точка маршрутизаторов, если эти интерфейсы не используются в качестве точки отправления или назначения при посылке IPv6-дейтограмм.

3. IPv6 соответствует модели IPv4, где подсеть ассоциируется с каналом. Одному каналу могут соответствовать несколько подсетей.

4.2.2. Представление адресов

Выделяют 3 формы записи IP-адресов:

1. Основная имеет следующий вид:

$x{:}\,x{:}\,x{:}\,x{:}\,x{:}\,x{:}\,x{:}\,x$, где x – шестнадцатеричные 16-битовые числа

Пример: fedc:ba98:7654:3210:FEDC:BA98:7654:3210,

1080:0:0:0:8:800:200C:417A.

Отметим, что писать начальные нули в конкретном поле не принято, но в каждом поле должна быть хотя бы одна цифра

2. IPv6-адреса очень часто длинные последовательности нулевых бит. Для того, чтобы сделать запись более удобной и читаемой, имеется специальный синтаксис для удаления нулей. В записи используется «::», тем самым указывается на наличие некоторого количества групп из 16 нулевых бит, однако данная запись может применяться только один раз.

Пример:

1080:0:0:0:8:800:200C:417A – unicast-адрес,

ff01:0:0:0:0:0:0:43 – multicast-адрес,

0:0:0:0:0:0:0:1 –адрес обратной связи,

0:0:0:0:0:0:0:0 – неспецифицированый адрес.

Более удобная и читаемая запись:

1080::8:800:200C:417A,

ff01::43,

::1,

::

3. Альтернативная форма записи. В данной форме записи младшая часть имеет стандартный вид для IPv4, т. е. представляется 4-мявосьмибитными октетами, остальные же части адреса – 6 шестнадцатеричных 16-битовых чисел:

x: x: x: x: x: x: d. d. d. d

0:0:0:0:0:0:13.1.68.3 ::13.1.68.3

0:0:0:0:FFFF:120.144.52.38 ::FFFF.120.144.52.38.

Тип IPv6 определяется по лидирующим битам адреса, называющимся префиксом. Данное поле может иметь переменную длину (представлено в табл. 4.1).

Таблица 4.1

Префиксы вIPv6-адресах

Тип адреса	Префикс	Часть адресного пространства
Зарезервировано	00000000	1/256
Не определено	00000001	1/256
Зарезервировано для NSAP	0000001	1/128
Зарезервировано для IPX	0000010	1/128
Не определено	0000011	1/128
Не определено	00001	1/32

Окончание табл. 4.1

Тип адреса	Префикс	Часть адресного пространства
Не определено	0001	1/16
Не определено	001	1/8
Провайдер юникаст-адресов	010	1/8
Не определено	011	1/8
Зарезервировано для географических юникаст-адресов	100	1/8
Не определено	101	1/8
Не определено	110	1/8
Не определено	1110	1/16
Не определено	11110	1/32
Не определено	111110	1/64
Не определено	1111110	1/128
Не определено	111111100	1/512
Локальный канальный адрес	1111111010	1/1024
Локальный адрес	1111111011	1/1024
Мультикаст-адрес	11111111	1/256

Unicast от multicast-адресов фактически отличаются значением старшего октета (у последнего первые 8 бит равны 1). Anycast-адреса берутся из пространства адресов unicast и синтаксически не отличаются от них.

4.2.3. Unicast-адреса

Существует несколько форм представления unicast-адресов в IPv6:

1. Глобальные unicastадреса провайдера.
2. Географические unicast адреса.
3. IPX иерархическийадрес.
4. IPv4 compatible host address. Предполагается, что данный список будет в дальнейшем расширятся.

Узлы IPv6 могут иметь существующую структуру, в зависимости от выполняемой роли. В общем виде адрес IPv6 будет следующим (рис. 4.3):

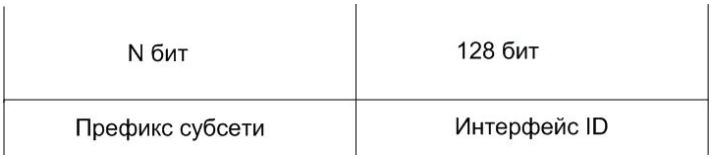


Рис. 4.3. Общий вид unicast-адресов

Для локальной системы сети, где применимы MAC-адреса, используются IP-адреса следующего типа (рис. 4.4):

N бит	80 – n бит	48
Префикс подписчика	ID субсети	Интерфейс ID

Рис. 4.4. Unicast-адреса с MAC-адресом

Включение MAC-адреса делается достаточно простой автоконфигурацией адресов, в данном случае MAC-адрес является идентификатором интерфейса.

Также применяются и другие варианты адреса (в случае, если сеть имеет сложную иерархическую структуру). В примере, представленном на рис. 4.5, идентификатор подсети делится на идентификатор области и идентификатор подсети. Формат такого адреса имеет вид:

S бит	n бит	m бит	128–S-n-m бит
Префикс получаетля	ID области	ID субсети	Интерфейс ID

Рис. 4.5. Unicast-адреса для сложноструктурированных сетей

Допускается использование в качестве ID интерфейса количество меньшее 48, при этом больший бит оставляется полям.

Есть также так называемый **неспецифицированный адрес**, который состоит из всех нулей (0:0:0:0:0:0:0:0).Он не может присваиваться какому-либо узлу, как правило, используется для записи поля IPv6-диаграммы, отправляемой интерфейсом для определения своего адреса у DHCP-сервера.

Юникастный адрес 0:0:0:0:0:0:0:1 называется адресом обратной связи. Используется для посылки диаграмм самому себе, данный адрес не может применяться для отправки дейтограмм за пределы узла.

IPv6 с вложенными IPv4 адресами

При необходимости организации туннелей для пересылки пакетов через маршрутную инфраструктуру IPv4 используются IPv6 unicast-адреса, которые в младших 32 битах содержит IPv4-адреса. Первые из них называются **IPv4-compatible IPv6-address** (рис. 4.6).

80 бит	16 бит	32 бита
0000….0000	0000	IPv4 адр.

Рис. 4.6. Unicast-адрес с вложенным IPv4-адресом

Определен и второй тип IPv6-адреса, который содержит внутри IPv4-адрес. Этот адрес используется для представления IPv6-адресов узлам IPv4 (тем, что не поддерживают IPv6). Этот тип адреса называется **IPv4-mapped IPv6-address** и имеет формат, показанный на рис. 4.7.

80 бит	16 бит	32 бита
0000….0000	FFFF	IPv4 адр.

Рис. 4.7. Unicast-адрес с вложенным IPv4-адресом

Глобальные unicast-адреса (провайдеров)

Данный адрес имеет следующий формат (рис. 4.8):

3 бита	n бит	m бит	S бит	125-S-n-m бит
010	ID регистратора	ID провайдера	ID клиента	Внутренний адрес

Рис. 4.8. Глобальные unicast-адреса

ID регистратора определяет организацию регистратора, который задает провайдерскую часть адреса. Использующийся термин префикс регистрации относится к старшей части адреса, включая поле ID регистратора. ID провайдера задает специфического провайдера, который определяет часть адреса клиента. Префикс провайдера – это старшая часть адреса, включая ID провайдера. ID клиента позволяет разделить клиентов подключенных к одному и тому же провайдеру. Префикс клиента – старшая часть адреса, включаяID клиента.

Внутренний адрес определяется клиентом, администратором, согласно топологии локальной сети.

Локальные unicast-адреса

Существует два типа таких адресов: первые – локальные адреса сети, вторые – локальные адреса каналов(предназначены для работы с каналом, а сети – с определенной сетью) (рис. 4.9).

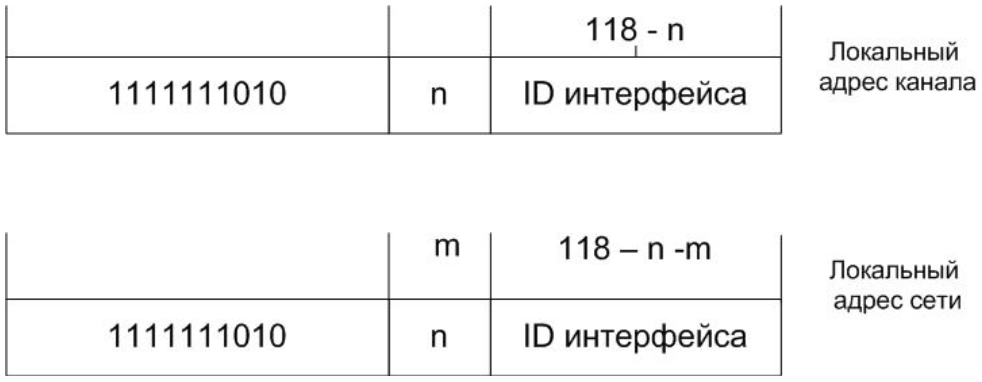


Рис. 4.9. Локальные unicast-адреса

4.2.4. Anycast–адреса

Anycast является адресом, принадлежащим нескольким интерфейсам, при этом пакет, посланный по anycast-адресу, будет доступен ближайшему интерфейсу в соответствии с метрикой маршрутизатора. Anycast-адреса выделяются из пространства unicast-адресов и используют один из известных форматов адресов. В итоге получаем следующее: если один unicast-адрес приписан нескольким интерфейсам, то он автоматически становится anycast-адресом. Anycast-адрес не может использоваться в качестве адреса отравителя пакета. Очень часто anycast-адрес приписывается маршрутизаторам, в таком случае его внешний вид будет следующим (рис. 4.10):

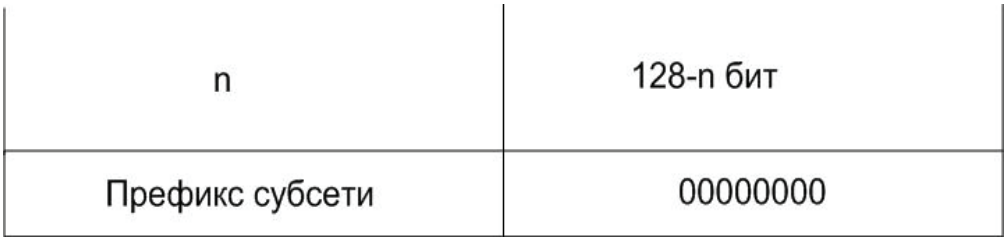


Рис. 4.10. Структура anycast-адреса

Префикс подсети (субсети) в anycast-адресе является префиксом, который идентифицирует определенный канал, поэтому синтаксически данный адрес идентичен адресу канала с идентификатором канала, равным 0.

Пакеты, посланные группе маршрутизаторов по anycast-адресу, будут отправлены всем маршрутизаторам, но реальный обмен данными состоится с первым ответившим.

4.2.5. Multicast-адреса

Является идентификатором IDгруппы узлов, узел может принадлежать любому числу multicast-групп. Общий вид multicast-адреса представлен на рис. 4.11.

Рис. 4.11. Структура multicast-адреса

Старшие три бита флага зарезервированы и пока используются в виде нулей. Четвертый бит флага указывает, что данный адрес является стандартным multicast-адресом, выделением из глобального пространства. T = 1 говорит о том, что данный адрес временный.

Поле scope предназначено для определения предельной зоны действия multicast-групп (табл. 4.2).

Таблица 4.2

Значение поля Scope

Значение поля scope	Зона действия
0	Зарезервирован
1	Область действия ограничена локальными узлами
2	Область действия ограничена локальным каналом
3	Не определено
4	Не определено
5	Область действия ограничена локальной сетью
6	Не определено
7	Не определено
8	Область действия ограничена локальной организацией
9	Не определено
A	Не определено
B	Не определено
C	Не определено
D	Не определено
E	Глобальные пределы
F	Зарезервировано

Значение постоянно присвоенного multicast-адреса не зависит от поля scope.

Пример:

FF01:0:0:0:0:0:0:43 означает, что все NTP-серверы одного и того же узла рассматриваются как отправители.

FF02:0:0:0:0:0:0:43 означает, что все NTP-серверы работают с тем же каналом, что и отправители.

FF05:0:0:0:0:0:0:43 означает, что все NTP-серверы принадлежат той же сети, что и отправитель.

FF0E:0:0:0:0:0:0:43 означает, что все NTP-серверы находятся в Интернете.

Временно выделенные multicast-адреса имеют значения только в пределах ограничений scope.Например, группа, определенная временным локальным multicastадресом FF05:0:0:0:0:0:0:43, не имеет никакого смысла для другой локальной сети или временной группы, использующей тот же групповой идентификатор.

Multicast, так же как и anycast-адреса, не могут использоваться в качестве адреса отправителя пакета.

Предопределенные multicast-адреса

Выделенные зарезервированные multicast-адреса, которые не будут присваиваться каким-либо multicast-группам:

FF01:0:0:0:0:0:0:0;
FF02:0:0:0:0:0:0:0;
FF03:0:0:0:0:0:0:0;
FF04:0:0:0:0:0:0:0;
FF05:0:0:0:0:0:0:0;
FF06:0:0:0:0:0:0:0;
FF07:0:0:0:0:0:0:0;
FF08:0:0:0:0:0:0:0;
FF09:0:0:0:0:0:0:0;
FF0A:0:0:0:0:0:0:0;
FF0B:0:0:0:0:0:0:0;
FF0C:0:0:0:0:0:0:0;
FF0D:0:0:0:0:0:0:0;
FF0E:0:0:0:0:0:0:0;
FF0F:0:0:0:0:0:0:0.

Примеры других multicast-адресов:

1. Адреса для обращения ко всем узлам:

FF01::1;
FF02::1.

Идентифицирует группу, включающую в себя все IPv6 (в пределах группы 1 – локальные узлы, группы 2 – локальные связанные узлы).

2. Адреса всех маршрутизаторов:

FF01::2;
FF02::2.

Идентифицирует группу всех IPv6-маршрутизаторов в пределах области 1 (локальные узлы) и области 2 (локальные связи).

3. DHCP Server/ relay agent:

FF02::C.

Идентифицирует группу всех IPv6 DHCP серверов и ретранслирующих агентов в пределах области 2 (локальный канал);

4. Адрес активного узла:

FF02::1:xxxx:xxxx.

4.2.6.Необходимые адреса узлов

Хост должен распознавать следующие адреса, как обращенные к нему:

- локальный адрес канала для каждого из интерфейсов;
- адрес обратной связи;
- выделенные unicast-адреса;
- multicast-адреса для обращения по всем узлам;
- multicast-адрес активного узла для каждого из присвоенных unicast- и anycast-адресов;
- multicast-адрес всех групп, к которым принадлежит хост.

Маршрутизатор должен распознавать следующие адреса:

- его локальный адрес канала для каждого из интерфейсов;
- выделенные anycast-адреса;
- адрес обратной связи;
- anycast-адрес маршрутизатора подсети для каналов, где он имеет интерфейсы;

– все другие unicast-адреса, которые использовались при маршрутизации;

– multicast-адрес для обращения ко всем узлам;

– multicast-адрес для обращения ко всем маршрутизаторам;

– multicast-адрес активного узла для каждого приписанного ему unicast- и anycast-адресов;

– multicast-адрес всех прочих групп, принадлежащих маршрутизатору.

Приложение должно предопределить следующие префиксы адресов:

– префикс не специфицированный адрес;

– префикс адрес обратной связи;

– префикс multicast-адреса (FF);

– локальные используемые префиксы;

– предопределенные multicast-префиксы;

– префиксы, совместимые с IPv4.

Контрольные вопросы

1. Каково назначение IP-адреса?
2. Какова структура IPv4-адреса?
3. Опишите понятия NETWORKIDи HOSTIDв IPv4.
4. Использование масок для определения NETWORKID и HOSTID.
5. Назовите особые IP-адреса.
6. Что такое частные адреса? Приведите примеры.
7. Приведите пример структуризации сети с помощью маски.
8. Приведите особенности IPv6-адресации.
9. Опишите архитектуру адресации IPv6.
10. Опишите формы представления IPv6-адресов.
11. Назначение и структура unicast-адресов.
12. Назначение и структура anycast-адресов.
13. Назначение и структура multicast-адресов.
14. Приведите перечень необходимых адресов, которые должны распознавать узлы.
15. Приведите перечень необходимых адресов, которые должны распознавать маршрутизаторы.
16. Приведите перечень необходимых адресов, которые должны распознавать приложения.

ТЕМА 5. РАСПРЕДЕЛЕНИЕ IP-АДРЕСОВ. ПРОТОКОЛ DHCP

План

1. Реализация DHCP в Windows.

2. Параметры DHCP.

3. Сравнение протоколов BOOTP и DHCP.

4. Принцип работы DHCP.

5. Статистика DHCP сервера.

6. Создание резервной копии БД DHCP-сервера.

Данная тема рассчитана на две лекции (лекции 9–10).

5.1. Реализация DHCP в Windows

Одной из основных задач системного администратора является настройка стека протоколов TCP/IP на всех компьютерах сети. Есть несколько необходимых параметров, которые следует настроить на каждом компьютере: IP-адрес, маска подсети, шлюз по умолчанию, IP-адреса DNS-серверов. Назначенные IP-адреса должны быть уникальны. В случае каких-либо изменений (например, изменился IP-адрес DNS сервера или шлюза по умолчанию), их нужно отразить на всех компьютерах. Если какие-либо параметры не указаны или не верны, сеть не будет работать стабильно. Если в сети менее нескольких десятков компьютеров, администратор может успешно справляться с задачей настройки стека TCP/IP вручную, т. е. на каждом компьютере отдельно вводить параметры. IP-адрес, назначенный таким образом, называется статическим. При числе узлов сети более нескольких десятков (а многие сети включают сотни и тысячи хостов) задача распределения параметров вручную становится трудной или вовсе невыполнимой.

В стеке TCP/IP существует протокол, позволяющий автоматизировать процесс назначения IP-адресов и других сетевых параметров, который называется DHCP – Dynamic Host Configuration Protocol (протокол динамической конфигурации хоста).

Использование этого протокола значительно облегчает труд системного администратора по настройке сетей средних и боль-

ших размеров. Описание протокола DHCP приводится в документе RFC 2131.

Протокол DHCP реализуется по модели «клиент-сервер», т. е. в сети должны присутствовать DHCP-сервер (роль которого может исполнять компьютер с операционной системой Windows Server 2003) и DHCP-клиент.

На компьютере-сервере хранится база данных с сетевыми параметрами, и работает служба DHCP-сервера. Компьютер-клиент (точнее, служба клиента DHCP) осуществляет запросы на автоматическую конфигурацию, и DHCP-сервер при наличии свободных IP-адресов выдает требуемые параметры.

Набор IP-адресов, выделяемых для компьютеров одной физической подсети, называется областью действия (scope). На одном сервере можно создать несколько областей действия. Важно только отслеживать, чтобы области действия не пересекались.

При запросе клиента, DHCP-сервер выделяет ему произвольный свободный IP-адрес из области действия, совместно с набором дополнительных сетевых параметров. При необходимости некоторые адреса из области действия можно зарезервировать (reserve) за определенным MAC-адресом.

В этом случае только компьютеру с этим MAC-адресом (например, DNS-серверу, адрес которого не должен меняться) будет выделяться зарезервированный IP-адрес.

Адреса выделяются клиентам на определенное время, поэтому предоставление адреса называется арендой (lease). Время аренды в Windows Server 2003 может быть от 1 минуты до 999 дней (или не ограничено), и устанавливается администратором.

Выделяют три типа областей:

– стандартные (описывает одну IP-сеть);
– суперобласть (совокупность стандартных);
– многоадресные (описывают IP-сети, предназначенные для многократной рассылки).

Стандартные служат для объединения компьютеров в логические подсети в рамках одной физической сети. При этом администратор сначала создает область для каждой подсети, а затем использует ее для определения параметров клиентов.

Любая стандартная область характеризуется следующими свойствами:

1) диапазон IP-адресов, из которых службой DHCP выбираются либо исключаются IP-адреса;

2) маска подсети;

3) срок аренды, назначаемый клиентам DHCP, которые динамически получают адреса.

В большинстве случаев на DHCP-сервере настраивается одна стандартная область, но если один DHCP сервер обслуживает несколько сетей, то создается несколько стандартных областей, которые в дальнейшем объединяются в суперобласти. При этом важно следить, чтобы диапазоныIP-адресов отдельных стандартных областей не пересекались.

Суперобласти. С помощью них можно получить ряд дополнительных возможностей:

1. Поддержка DHCP-клиентов, расположенных на отдельном сегменте физической сети, в которой используется несколько логических IP-сетей. Если в каждой физической сети или подсети используется несколько логических сетей или подсетей, то такие конфигурации называются **мультисетевыми.**

2. Поддержка удаленных DHCP-клиентов, расположенных на удаленной стороне агентов ретрансляторов.

Суперобласти позволяют разрешать следующие проблемные ситуации:

1) доступный диапазон в настоящее время исчерпан почти полностью, исходная область включает весь диапазон IP-сети для расширения адресного пространства одного и того же физического сегмента сети с последующим объединением в суперобласти;

2) клиенты должны перейти со временем на другую область, например, для перенумерации текущей IP-сети, в таком случае также создается новая область с последующим объединением в суперобласти;

3) необходимость использовать два DHCP-сервера в физическом сегменте для управления различными логическими сетями.

Многоадресная область. В качестве диапазона адресов многоадресной групповой рассылки используется класс адресов D. Данные адреса не могут использоваться в стандартных областях.

Во всех TCP/IPсетях каждый узел сначала должен получить индивидуальный IP (классы A, B, C).Без назначения такого адреса настройка узла на поддержку и использование вторичных IP-адресов (адреса многоадресной рассылки) невозможна.

Членство в группе многоадресной рассылки является динамическим, что означает возможность присоединения в любое время IP-узлов или их выход.

Создается область многоадресной рассылки, которая будет назначать клиенту групповой адрес после получения индивидуального.

В DHCP-серверах можно резервировать за определенным MAC-адресом соответствующий IP-адрес, также можно в области добавлять исключения.

Исключения – это диапазон IP-адресов, из которого клиентам адреса не будут выдаваться. Как правило, в диапазон исключений попадают все статически заданные IP-адреса в сети.

5.2. Параметры DHCP

Основная функция протокола DHCP – предоставление в аренду IP-адреса. Однако для правильной работы в сети TCP/IP хосту необходим еще ряд параметров, которые также можно распространять посредством DHCP. Набор параметров указан в RFC 2132.

Перечислим только **основные параметры**:

- Subnet mask – маска подсети;
- Router – список IP-адресов маршрутизаторов;
- Domain Name Servers – список адресов DNS-серверов;
- DNS Domain Name – DNS-суффикс клиента;
- WINS Server Names – список адресов WINS-серверов;
- Lease Time – срок аренды (в секундах);
- Renewal Time (T1) – период времени, через который клиент начинает продлевать аренду;
- Rebinding Time (T2) – период времени, через который клиент начинает осуществлять широковещательные запросы на продление аренды.

Параметры могут применяться на следующих **уровнях**:

- уровень сервера;
- уровень области действия;
- уровень класса;
- уровень клиента (для зарезервированных адресов).

Параметры, определенные на нижележащем уровне, перекрывают параметры вышележащего уровня, например, параметры клиента имеют больший приоритет, чем параметры сервера. Самый высокий приоритет имеют параметры, настроенные вручную на клиентском компьютере.

Уровень класса используется для объединения клиентов в группы и применения для групп отдельных параметров. Отнести клиента к определенному классу можно, применив утилиту IPconfig с ключом /setclassid.

5.3. Принцип работы DHCP

Процесс функционирования служб DHCP заключается в обмене сообщениями между сервером и клиентом. Список используемых сообщений представлен в табл. 5.1.

Таблица 5.1

Типы DHCP-сообщений

Тип сообщения	Направление	Значение
DHCPDISCOVER (DHCP-обнаружение)	Клиент → сервер	Широковещательный запрос для обнаружения DHCP-сервера
DHCPOFFER (DHCP-предложение)	Сервер → клиент	Ответ на DHCPDISCOVER, содержит предлагаемые сетевые параметры
DHCPREQUEST (DHCP-запрос)	Клиент → сервер	Запрос предложенных параметров
DHCPACK (DHCP-подтверждение)	Сервер → клиент	Подтверждение сетевых параметров
DHCPNAK (DHCP-несогласие)	Сервер → клиент	Отклонение запроса клиента
DHCPDECLINE (DHCP-отказ)	Клиент → сервер	Отказ клиента от предложенных параметров
DHCPRELEASE (DHCP-освобождение)	Клиент → сервер	Освобождение арендованного IP-адреса
DHCPINFORM (DHCP-информация)	Клиент → сервер	Запрос дополнительных параметров

Диаграмма переходов, иллюстрирующая принципы работы протокола DHCP, приведена на рис. 5.1. На схеме овалами обозначены состояния, в которых может находиться DHCP-клиент. Из одного состояния в другое клиент может переходить только по дугам. Каждая дуга помечена дробью, числитель которой обозначает событие (чаще всего это сообщение от DHCP-сервера), после которого клиент переходит в соответствующее состояние, а знаменатель

описывает действия DHCP-клиента при переходе. Черточка в числителе означает безусловный переход.

Начальное состояние, в котором оказывается служба DHCP-клиента при запуске, – это «Инициализация». Из этого состояния происходит безусловный переход в состояние «Выбор» с рассылкой широковещательного сообщения DHCPDISCOVER. DHCP-серверы (в одной сети их может быть несколько), принимая сообщение, анализируют свою базу данных на предмет наличия свободных IP-адресов.

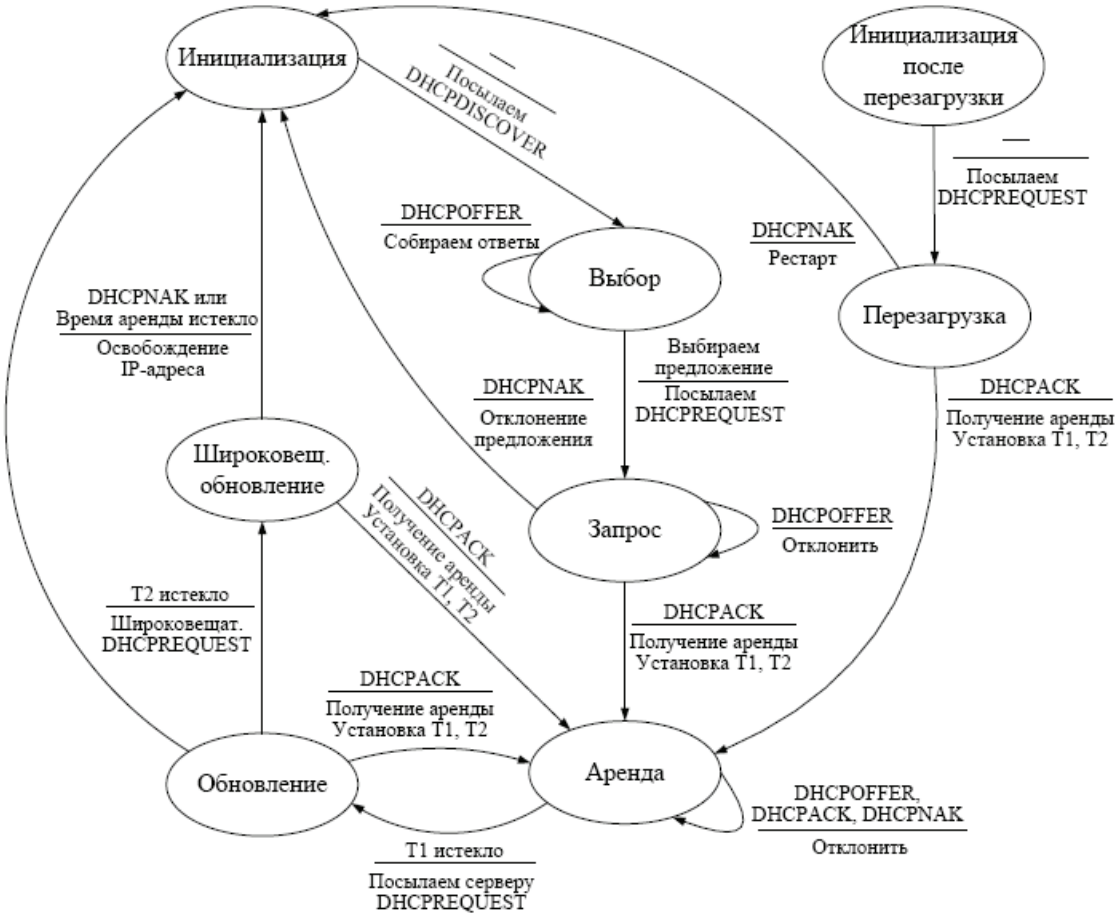


Рис. 5.1. Диаграмма переходов, иллюстрирующая принципы работы протокола DHCP

В случае успеха, серверы отправляют сообщение DHCPOFFER, которое помимо IP-адреса содержит дополнительные параметры, призванные помочь клиенту выбрать лучшее предложение. Сделав выбор, клиент посылает широковещательное сообщение DHCPREQUEST, запрашивая предложенный IP-адрес и требуемые параметры (напри-

мер, маска подсети, шлюз по умолчанию, IP-адреса DNS-серверов и др.) и переходит в состояние «Запрос». Данное сообщение требуется посылать широковещательно (т. е. оно должно доставляться всем компьютерам подсети), чтобы DHCP-серверы, предложения которых клиент отклонил, знали об отказе.

В состоянии «Запрос» клиент ожидает подтверждение сервера о возможности использования предложенных сетевых параметров. В случае прихода такого подтверждения (сообщение DHCPACK), клиент переходит в состояние «Аренда», одновременно начиная отсчет интервалов времени T1 и T2. Если сервер по каким-либо причинам не готов предоставить клиенту предложенный IP-адрес, он посылает сообщение DHCPNAK. Клиент реагирует на это сообщение переходом в исходное состояние «Инициализация», чтобы снова начать процесс получения IP-адреса.

Состояние «Аренда» является основным рабочим состоянием – у клиента присутствуют все необходимые сетевые параметры, и сеть может успешно функционировать.

Через временной интервал T1 от момента получения аренды (обычно T1 равно половине общего времени аренды) DHCP-клиент переходит в состояние «Обновление» и начинает процесс обновления аренды IP-адреса. Сначала клиент посылает DHCP-серверу сообщение DHCPREQUEST, включающее арендованный IP-адрес. Если DHCP-сервер готов продлить аренду этого адреса, то он отвечает сообщением DHCPACK и клиент возвращается в состояние «Аренда» и заново начинает отсчитывать интервалы T1 и T2.

В случае, если в состоянии «Обновление» по истечении интервала времени T2 (который обычно устанавливается равным 87,5% от общего времени аренды) все еще не получено подтверждение DHCPACK, клиент переходит в состояние «Широковещательное обновление» с рассылкой широковещательного сообщения DHCPREQUEST. Такая рассылка делается в предположении, что DHCP-сервер поменял свой IP-адрес (или перешел в другую подсеть) и передал свою область действия другому серверу. В этом состоянии получение DHCPACK возвращает клиента в состояние «Аренда» и аренда данного IP-адреса продлевается. Если клиент получает от сервера сообщение DHCPNAK или общее время аренды истекает, то происходит переход в состояние «Инициализация» и клиент снова пытается получить IP-адрес.

В процессе работы может оказаться, что время аренды не истекло, а служба DHCP-клиента прекратила работу (например, в слу-

чае перезагрузки). В этом случае DHCP-клиент начинает работу в состоянии «Инициализация после перезагрузки», рассылает широковещательное сообщение DHCPREQUEST и переходит в состояние «Перезагрузка». В случае подтверждения продления аренды (сообщение DHCPACK от DHCP-сервера), клиент переходит в состояние «Аренда». Иначе (сообщение DHCPNAK) клиент оказывается в состоянии «Инициализация».

5.4. Адреса для динамической конфигурации

При настройке областей действия перед администратором встает вопрос о том, какой диапазон адресов выбрать для сети своей организации. Ответ зависит от того, подключена ли сеть к Интернету. Если сеть имеет доступ в Интернет, диапазон адресов назначается провайдером (ISP – Internet Service Provider, поставщик Интернет-услуг) таким образом, чтобы обеспечить уникальность адресов в Интернете.

Чаще всего бывает так, что провайдер выделяет один или несколько адресов для прямого доступа в Интернет и они присваиваются прокси-серверам, почтовым серверам и другим хостам, которые являются буферными узлами между сетью организации и Интернетом.

Большинство остальных хостов получают доступ к Интернет-трафику через эти буферные узлы. В этом случае диапазон внутренних адресов организации должен выбираться из множества частных адресов.

Частные адреса (Private addresses), описанные в RFC 1918, специально выделены для применения во внутренних сетях и не могут быть присвоены хостам в Интернете. Существует три диапазона частных адресов:

- *ID подсети – 10.0.0.0,*
- *ID подсети – 172.16.0.0,*
- *ID подсети – 192.168.0.0.*

Внутри этих диапазонов адресов можно организовывать любые возможные подсети. Если сеть не имеет доступа в Интернет, то, теоретически, можно выбрать любой диапазон IP-адресов, не учитывая наличия хостов с такими же адресами в Интернете.

Также следует отметить, что помимо описанных частных адресов существует диапазон *автоматических частных адресов* APIPA (Automatic Private IP Address): ID подсети – 169.254.0.0, маска подсети: 255.255.0.0.

Адрес из этого диапазона выбирается хостом TCP/IP случайно, если отсутствует статический IP-адрес, DHCP-сервер не отвечает и не указан альтернативный статический адрес. После выбора IP-адреса, хост продолжает посылать запросы DHCP-серверу каждые пять минут.

5.5. Статистика DHCP-сервера

Статистику удобно использовать для оценки текущего состояния сервера. Это особенно полезно для определения количества свободных и занятых адресов.

Статистику можно просматривать как для отдельных областей, так и для суперобластей (рис. 5.2).

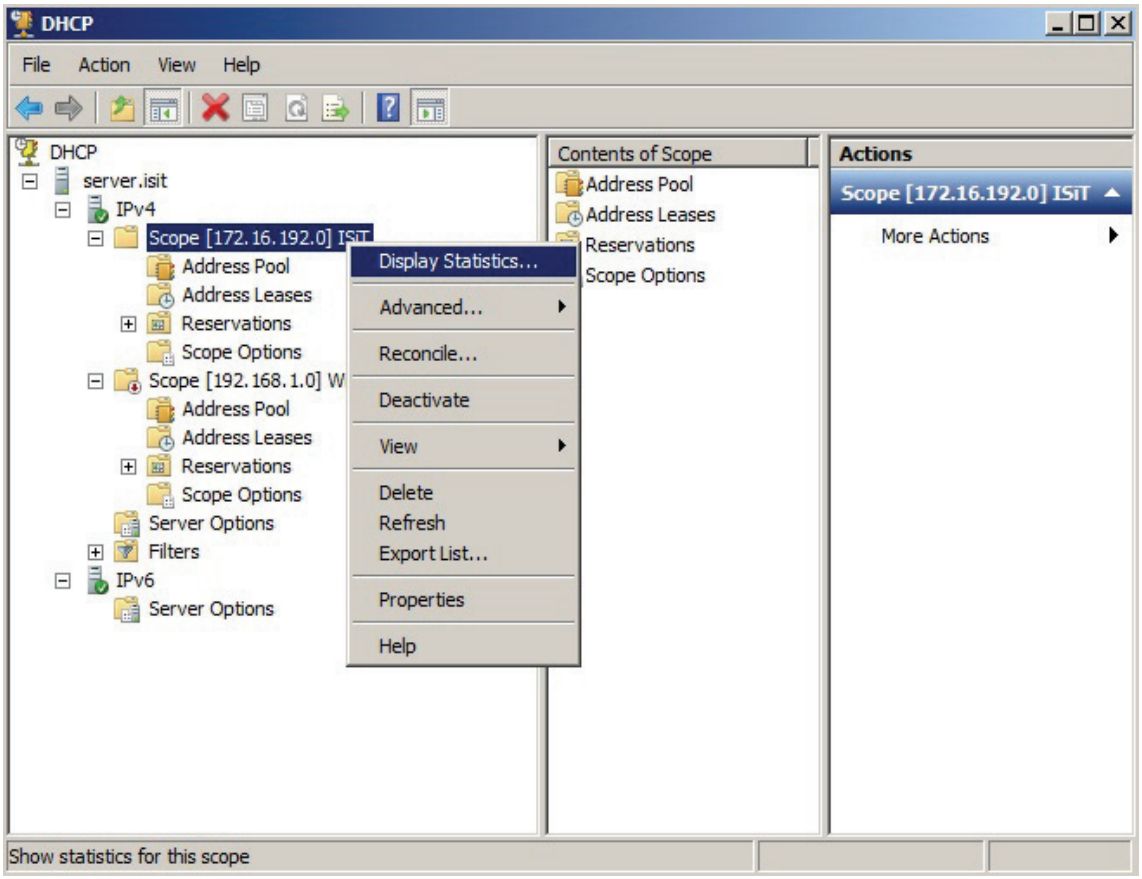


Рис. 5.2. Пример просмотра статистики DHCP-сервера

В статистике могут быть отражены следующие **параметры**:

– время запуска – время последнего запуска или перезапуска DHCP;

– время работы – общее время работы DHCP сервера прошедшее с момента запуска;

– найдено – количество обработанных сообщений DHCPDISCOVER;

– предложено – DHCPOFFER;

– запрещено – DHCPREQUEST;

– подтверждено – DHCPACK;

– не подтверждено – DHCPNACK;

– отклонено – DHCPDECLINE;

– освобождено – DHCPRELEASE;

– всего областей – количество областей, определенных для сервера или суперобласти;

– всего адресов – общее количество адресов в области, суперобласти или областях DHCP;

– используется – количество и процент использования адресов;

– доступно – количество и процент доступных адресов.

5.6. Журналы DHCP-сервера

Файлы журнала DHCP-сервера Windows Server 2003 разработаны с учетом необходимости их использования без дополнительного наблюдения, администрирования и сохранения дисковых ресурсов. Файлы журналов по умолчанию хранятся в папке %systemroot%\system32\dhcp и имеют имена DhcpSrvLog.day, где вместо day подставляется сокращенное название дня недели. По истечении недели файлы журналов записываются поверх уже существующих, что значительно сокращает объем дискового пространства, используемого журналами.

Анализ журналов DHCP-сервера бывает полезен при возникновении различных проблем в работе сервера. Например, при невозможности авторизовать сервер в Active Directory, можно выяснить точную причину сбоя в файле журнала. Ниже описывается формат файлов журнала DHCP-сервера и их использование для сбора дополнительных сведений об операциях службы DHCP-сервера в сети.

Коды основных событий журналов DHCP:

00 – начало ведения журнала;

01 – остановлено ведение журнала;

02 – ведение журнала временно остановлено из-за отсутствия дискового пространства;

10 – клиенту выделен новый IP;

11 – аренда адреса продлена клиентом;

12 – аренда адреса прекращена клиентом;

13 – найденный IP-адрес используется в сети;

14 – запрос на аренду не может быть удовлетворен.

Журналы DHCP-сервера представляют собой текстовые файлы, использующие в качестве разделителей запятые, в которых каждая запись журнала представляет одну строку текста. Каждая запись имеет следующий формат:

Код, дата, время, описание, IP-адрес, имя узла, MAC-адрес.

Подробное описание каждого из этих полей приведено в табл. 5.2. Ниже приведен краткий отрывок журнала, созданного службой DHCP-сервера:

Код, дата, время, описание, IP-адрес, имя узла, MAC-адрес
00,08/22/02,12:43:06,Запущена,,,
60,08/22/02,12:43:21,Нет контроллеров домена, поддерживающих службы каталога,,TEST,
63,08/22/02,12:43:28,Перезапуск случайной проверки,,,
01,08/22/02,13:11:13,Остановлена,,,
00,08/22/02,12:43:06,Запущена,,,
55,08/22/02,12:43:54,Авторизовано (обслуживается),,TEST,

Таблица 5.2

Коды событий в журналах DHCP-сервера

Поле	Описание
Код	Код события DHCP-сервера
Дата	Дата занесения записи в журнал на DHCP-сервере
Время	Время занесения записи в журнал на DHCP-сервере
IP-адрес	IP-адрес DHCP-клиента
Имя узла	Имя узла DHCP-клиента
MAC-адрес	Аппаратный адрес сетевого адаптера компьютера DHCP-

В данном примере DHCP-сервер не был авторизован при первоначальном запуске и впоследствии был остановлен. После авторизации в Active Directory,сервер был запущен и начал обслуживать DHCP-клиентов.

5.7. База данных DHCP-сервера

DHCP-серверы используют для хранения и доступа к базе данных механизм ESE (Extensible Storage Engine). Он же используется для хранения и доступа к данным каталога Active Directory и базе данных Microsoft Exchange.

База данных DHCP-сервера не имеет встроенного ограничения числа записей. Число хранимых записей определяется только объемом свободного пространства на дисках сервера. Размер БД зависит от числа DHCP-клиентов в сети. Она растет со временем в результате запуска и остановки клиентов в сети. Размер БД DHCP не пропорционален числу активных записей аренды адресов клиентами. С течением времени, поскольку некоторые записи DHCP-клиентов устаревают и удаляются, остается неиспользуемое пространство.

Чтобы восстановить неиспользуемое дисковое пространство, БД DHCP должна быть сжата. Динамическое сжатие происходит автоматически во время простоя сервера. Хотя динамическое сжатие значительно уменьшает необходимость автономного сжатия, необходимость последнего не исключается. Автономное сжатие более эффективно восстанавливает пространство на диске. Для очень больших и нагруженных сетей с числом DHCP-клиентов больше 1000 узлов оно должно выполняться примерно один раз в месяц. Для сетей меньшего размера сжатие БД вручную целесообразно выполнять один раз в несколько месяцев.

Создание резервной копии БД DHCP-сервера:

1. Остановите DHCP-сервер.
2. Отключите службу DHCP-сервера в списке служб. Это предотвратит запуск DHCP сервера после переноса БД.
3. Скопируйте папку %systemroot% \system32 \dhcpи вложенные папки во временную папку на сервере-получателе.
4. Запустите редактор реестра regedit32.exe и раскройте HKEY-LOCALMACHINE\SYSTEM\CurrentControlSet\Services\DHCP Server. Сохраните в текстовом файле.
5. Удалите папку %systemroot% \system32 \dhcp на исходном сервере.
6. Удалите службу DHCP-сервера на исходном компьютере.
7. Если служба DHCP еще не установлена, то установите и перезагрузите сервер.
8. Остановите службу DHCP-сервера. Переименуйте файл System.mdb в System.svr во временной папке, содержащей копию БД исходной DHCP.

9. Скопируйте временную папку, содержащую копию БД DHCP во вложенную папку %systemroot% \system32 \dhcp для замены существующей БД DHCP.

10. Запуститеregedit32.exe.

Восстановление базы данных DHCP-сервера.

Необходимо выполнить следующие действия на компьютере, используемом в качестве получателя БД DHCP.

1. Если служба DHCP-сервера еще не установлена, то установите ее и перезагрузите сервер.

2. Остановите службу DHCP-сервера.

3. Переименуйте файл System.mdb в System.src во временной папке, содержащей копию БД исходного DHCP-сервера.

4. Скопируйте временную папку, содержащую копию БД DHCP-сервера, и все вложенные папки в папку %systemroot%\system32\dhcp для замены существующей БД DHCP-сервера.

5. Запустите редактор реестра regedt32.exe и раскройте раздел HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DHCP Server. Восстановите сведения данного раздела реестра из файла, сохраненного при создании резервной копии DHCP-сервера.

6. Запустите службу DHCP-сервера.

7. Откройте Консоль управления DHCP и выполните согласование всех областей сервера.

Контрольные вопросы

1. Для решения какой проблемы предназначен протокол DHCP?
2. Что такое область действия?
3. Почему адреса предоставляются в аренду на время, а не навсегда?
4. Перечислите основные параметры DHCP.
5. Назовите диапазоны частных адресов. Для чего они нужны?
6. Поясните значение сообщений DHCPDISCOVER, DHCPOFFER,DHCPREQUEST, DHCPACK.
7. По диаграмме переходов объясните принципы работы DHCP-клиента.
8. Каково назначение статистики DHCP-сервера?
9. Какая информация содержится в журнале DHCP-сервера?
10. Опишите структуру журнала DHCP-сервера.
11. Опишите структуру БД DHCP-сервера.
12. Опишите правила переноса БД DHCP-сервера.

ТЕМА 6. ИМЕНА В TCP/IP. СИСТЕМА ИМЕН DNS И NETBIOS. СЛУЖБЫ DNS И WINS

План

1. **Система доменных имен.**
2. **Процесс разрешения имен (итеративный, рекурсивный).**
3. **База данных DNS.**
4. **Разрешенные символы DNS-имена.**
5. **Мониторинги устранения неполадок.**
6. **NetBios и служба WINS.**

Данная тема рассчитана на три лекции (лекции 11–13).

6.1. Система доменных имен

Символьный адрес (имя) не является обязательным, но он упрощает работу пользователей в сети. Существует два типа символьных имен, которые используются в IP-сетях:

– DNS-имя(RFC 1034, 1035);

– NetBIOS-имена.

Фрагмент системы доменных имен пространства Интернет представлен на рис. 6.1.

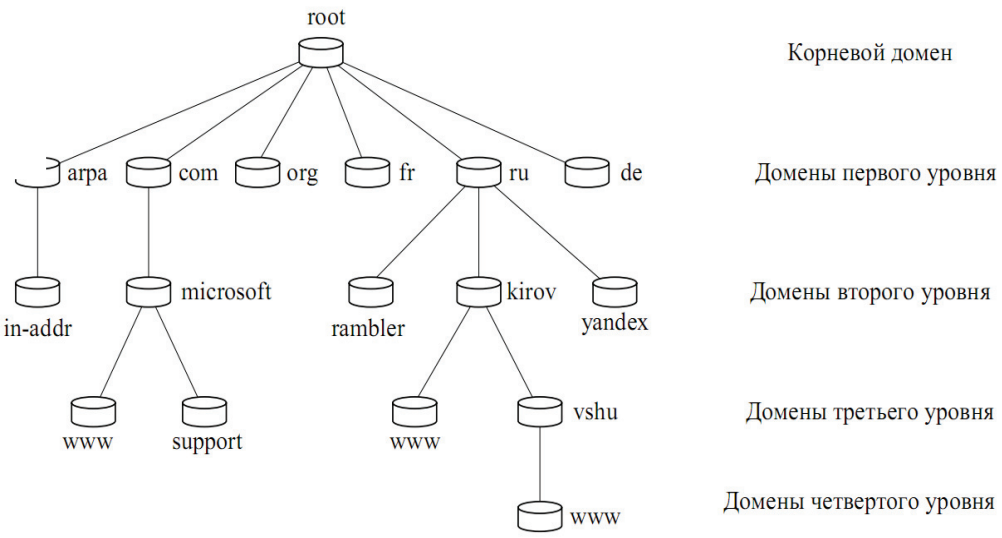


Рис. 6.1. Система доменных имен

Корневой домен как реальный узел не существует, он исполняет роль вершины дерева. Его потомки (поддомены) – это домены 1-го (верхнего) уровня. Их можно условно разделить на 3 группы:

– ARPA –домен, который используется для преобразования IP-адресов в доменное имя (обратное преобразование);

– домены организаций (com, org, net и т.д.);

– географические домены стран – имена для доменов, зарегистрированных в соответствующих странах (например, *ru* – для России, *ua* – для Украины, *uk* – для Великобритании и т. д.).

Домен 1-го уровня включает в себя только домены 2-го, записи об отдельных хостах могут содержатся в доменах, начиная со 2-го. Созданием и управлением домена 1-го уровня занимается международная организацияICANN.Домены 2-го уровня, находящиеся в географических доменах, распределяются специальными национальными организациями. Управлением доменами третьего и следующего уровней занимаются владельцы соответствующих доменов второго уровня.

Полностью доменное имя FQDN записывается следующим образом: имя хоста(лист в дереве пространства имен), затем через точку следует DNS-суффикс, запись заканчивается точкой, после которой подразумевается корневой домен. Например, www.vshu.kirov.ru. При этом www – имя хоста, а vshu.kirov.ru – DNS-суффикс.

Для согласования двух систем адресаций необходима служба, которая занимается преобразованием доменных имен в IP-адреса и обратно. Данные функции выполняет служба DNS.

Процесс преобразования доменного имени в IP-адрес называется разрешением доменного имени. Простейшим способом разрешения доменного имени является файл hosts. Такой прием используется, как правило, в небольших сетях.

Для больших сетей обязательным условием является автоматизация регистрации каких-либо изменений, что и привело к созданию службы DNS.

Служба поддерживает распределенную базу данных, которая хранится на специальных компьютерах (DNS-серверах). Вся информация не хранится в одном месте, ее части распределены по отдельным DNS-серверам. Так, например, за домены 1-го уровня отвечают 13 корневых серверов, имеющих имена от A.ROOT-SERVERS.NET до M.ROOT-SERVERS.NET, расположенных по всему миру. Такие части пространства называются зонами. Деление на зоны осуществляется исходя из удобства администрирования. Одна зона может содержать

несколько доменов, в тоже самое время информация о домене может быть рассредоточена по нескольким зонам. В целях повышения надежности и производительности, зона может быть размещена одновременно на нескольких серверах, в этом случае один из серверов является главным и хранит основную копию зоны (primary zone), остальные являются дополнительными, на них содержатся вспомогательные копии зоны (secondary zone).

Для преобразования IP-адресов в DNS существуют зоны обратного преобразования(reverse lookup zone). На верхнем уровне пространства имен Интернета этим зонам соответствует домен in-addr.arpa, имеющий структуру, представленную на рис. 6.2.

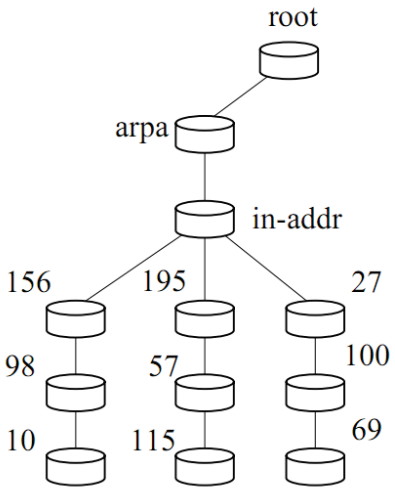


Рис. 6.2. Зона обратного просмотра доменных имен

Следуя правилам формирования DNS-имен, зона обратного преобразования, соответствующая подсети 156.98.10.0, будет называться 10.98.156.in-addr.arpa.

6.2. Процесс разрешения имен

В процессе разрешения участвуют DNS-клиент и DNS-сервер. Системный компонент DNS-клиента, называется DNS-распознавателем, отправляет запросы на DNS-серверы. Распознаватели бывают двух видов:

- интерактивные – DNS-сервер обращается к DNS-серверу с просьбой разрешить имя без обращения к другим DNS-серверам;
- рекурсивные – всю работу по разрешению имени выполняет DNS-сервер, путем отправки запросов другим DNS-серверам. DNS-

сервер всегда сначала ищет имя в собственной базе данных или в кэше, в случае отсутствия обращается к другим серверам.

В основном DNS-клиентами используются рекурсивные запросы. На рис. 6.3 проиллюстрирован процесс разрешения доменного имени с помощью рекурсивного запроса.

Сначала DNS-клиент осуществляет поиск DNS-имен в собственном локальном кэше. Это память для временного хранения ранее разрешенных запросов. В эту же память переносится содержимое файла HOSTS (каталог windows/system32/drivers/etc). Утилита IPconfigсключом /displaydnsотображает содержимое DNS-кэша. Если кэш не содержит требуемой информации, DNS-клиент обращается с рекурсивным запросом к предпочитаемому DNS-серверу (Preferred DNS server), адрес которого указывается при настройке стека TCP/IP. DNS-сервер просматривает собственную базу данных, а также кэш-память, в которой хранятся ответы на предыдущие запросы, отсутствующие в базе данных. В том случае, если запрашиваемое доменное имя не найдено, DNS-сервер осуществляет итеративные запросы к DNS-серверам верхних уровней, начиная с корневого DNS-сервера.

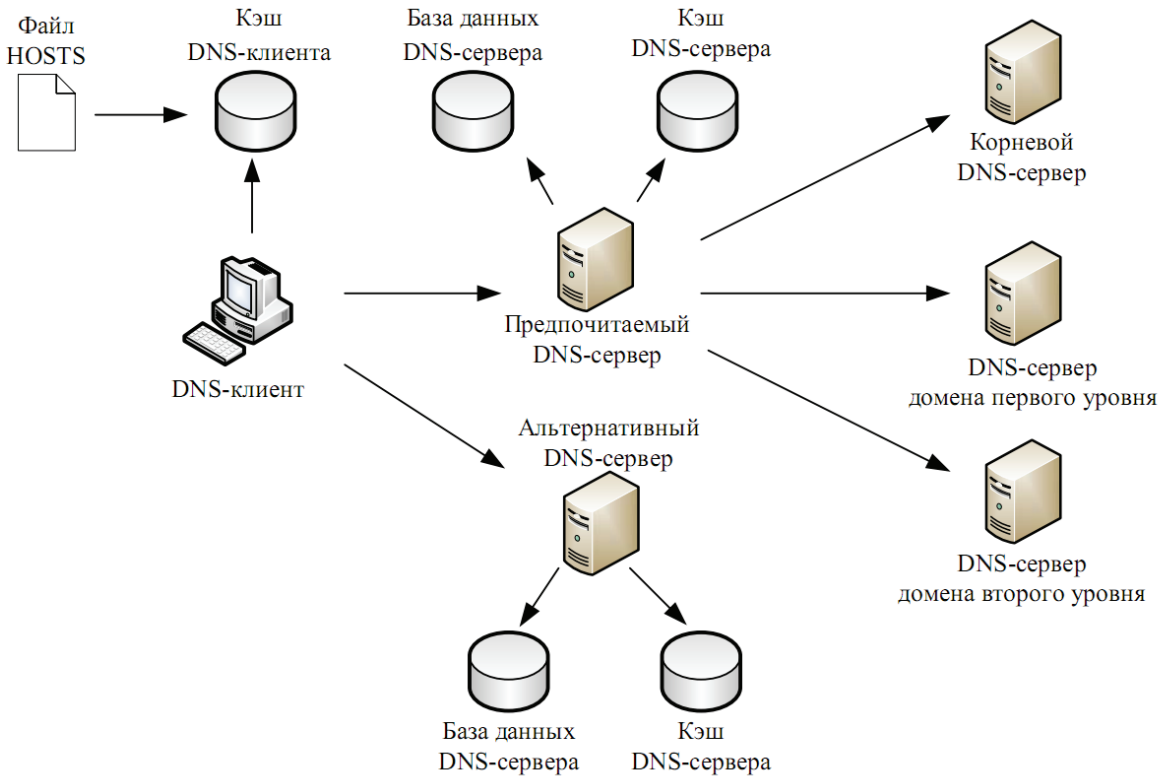


Рис. 6.3. Процесс рекурсивного разрешения имен

Рассмотрим процесс разрешения доменного имени на примере. Пусть, требуется разрешить имя www.microsoft.com. Корневой домен содержит информацию о DNS-сервере, содержащем зону .com. Следующий запрос происходит к этому серверу, на котором хранятся данные о всех поддоменах зоны .com, в том числе о домене microsoft и его DNS-сервере. Сервер зоны microsoft.com может непосредственно разрешить имя www.microsoft.com в IP-адрес.

Обращение к альтернативному серверу осуществляется только если основной сервер недоступен.

Просмотр DNS-кэша осуществляется утилитой ipconfig /displaydns. Очистка кэша – ipconfig / flushdns**.**

6.3. База данных DNS

Единичный информационный объект базы данных DNS называют запись о ресурсах (resourcerecord). Каждая запись имеет ассоциируемый с ней тип, описывающий категорию данных и тип сети, которой принадлежит описываемый объект. При этом допускаются различные схемы адресации.

Каждый узел может иметь несколько доменных имен. Одно из этих имен должно быть объявлено официально каноническим именем узла, остальные будут называться псевдонимами, ссылающимися на каноническое имя узла. Имя домена идентифицирует ресурс системы. Эта ассоциация хранится в БДDNS в виде отдельной записи. Компоненты ресурсных записей представлены в табл. 6.1, а типы ресурсных записей – в табл. 6.2.

Таблица 6.1

Компоненты ресурсных записей

Name	Имя, которое идентифицирует ресурс. В нем содержится имя домена или хоста, в котором расположен данный ресурс.
Type	Тип ресурса. Обозначает группу принадлежности ресурса, например адрес хоста или id маршрутизатора почтового роутера.
CLASS	Идентифицирует формат данных ресурса. Для различных типов ресурсов класс может означать разные понятия. Например, класс IN использует только 32-битные IP-адреса, CSNet использует как 32-битные, так и адреса в соответствии с протоколом X25 и номера телефонов. Можно сделать вывод, что поле class указывает, как использовать информацию хранящуюся в данном ресурсе.

Окончание табл. 6.1

TTL	Задает временной интервал продолжительности нахождения данной записи в кэше. При просмотре этой записи в кэше интервал уменьшается, а если становится меньше или равным 0, то запись удаляется из кэша.
RDLENSTH	Длина поля данных.
RDATA	Данные ресурса. Максимальная длина поля составляет 65535 байт, формат представленных данных определяется полями типа и класса.

Таблица 6.2

Типы ресурсных записей

Тип ресурсной записи	Величина	Описание
A	1	Отображает имя узла на IP-адрес (например, для домена *microsoft.com* узлу с именем *www.microsoft.com* сопоставляется IP-адрес с помощью такой записи:www A207.46.199.60)
NS	2	Сервер имен домена определяет имя хоста, который управляет пространством имен и адресов домена и тем самым устанавливает нижнюю границу зоны открытой записью SOA(точнее, определяет 1-ую запись вне данной зоны)
MX	15	Идентификатор почтового ресурса(определяет хост, который служит в домене в качестве обработчика всей полученной почты)
CNAME	5	Псевдоним ресурса
SOA	6	Имя домена, определяющего начало зоны пространства имен данного домена(владельца записи). Кроме тех случаев, когда сервер имен передает полномочия самому себе. SOA определяет верхнюю границу области полномочий. Кроме того, поле данных может содержать дополнительную информацию о зоне, используемую сервером имен. Записи SOA никогда не кэшируются и создаются при инсталляции сервера имен
PTR	12	Указатель другой части пространства имен домена
WKS	11	Описание серверов хоста
MINFO	13	Id процессора

Кроме вышеперечисленных, тип может принимать и другие значения. Иногда бывает, что значения устаревают, заменяются другими. Например, ND и NF были заменены на MX.

Записи RR хранятся в базе данных DNS и передаются в пакетах DNS-протокола в двоичном виде. Однако, как известно, RR модифицируются администратором в файлах главного архива в текстовом формате. Текстовый формат представления состояния базы данных значительно упрощает процедуры вставки, модификации или удаления записей.

Текстовый файл содержит последовательность записей, которые располагаются в строчки, заканчивающиеся символом перевода строки – <CRLF>. Для размещения информации на нескольких строках используются скобки. В табл.6.3 перечислены некоторые из символов, имеющих специальное значение.

Таблица 6.3

Специальные символы для текстового представления БД DNS

Символы	Значение
.	Отдельно стоящая точка в поле name обозначает текущий домен
@	Отдельно стоящий символ "@" в поле name обозначает текущий исходный домен
( )	Скобки используются для размещения поля data на нескольких строках (когда поле data занимает несколько строк)
*	Метасимвол. Заменяет любой набор символов
;	Символ комментария. От этого символа и до конца строки информация игнорируется.

Отметим, что в записях ресурсов доменное имя, не заканчивающееся точкой, считается относительным. При обработке оно прибавляется к текущему домену. Поэтому, когда задается полное имя, его необходимо заканчивать точкой.

6.4. Разрешенные символы в DNS-именах

Изначально имена узлов ограничивались набором символов, указанных в документах RFC 952 и 1123. Эти ограничения следующие:

- прописные и строчные буквы латинского алфавита;
- цифры;
- дефис.

Первым символом в именах DNS могла быть цифра, а имена должны были кодироваться и представляться с помощью набора ASCII. Эти требования сохранялись и когда система DNS была введена

как часть документа RFC 1035, который содержал на тот момент одну из спецификаций DNS.

В силу того, что в этих стандартах зачастую использовались расширения символов,Microsoft пошла на расширения поддержки символов в DNS за рамки спецификаций RFC 1035.В настоящее время в именах DNS используется расширенный набор символов UTF-8.

6.5. Мониторинги устранения неполадок

Для проверки способности DNS-серверов выполнять разрешение имен, используется утилита nslookup. Утилита может работать в двух режимах:

• режим командной строки – обычный режим запуска утилит командной строки (выполняется в этом режиме, если указан какой-либо ключ);

• интерактивный режим – в этом режиме возможен ввод команд и ключей утилиты без повторения ввода имени утилиты.

Команды утилиты nslookup:

• help или ? – вывод справки о командах и параметрах утилиты;

• set – установка параметров работы утилиты;

• server <имя> – установка сервера по умолчанию (Default Server), используемого утилитой, с помощью текущего сервера по умолчанию;

• lserver <имя> – установка сервера по умолчанию утилиты с помощью первоначального;

• root – установка сервера по умолчанию утилиты на корневой сервер;

• ls <домен> – вывод информации о соответствии доменных имен IP-адресам для заданного домена;

• exit – выход из интерактивного режима.

Журнал событий DNS-сервера регистрирует информацию об ошибках. Его можно посмотреть в консоли DNS, в окне свойств журнала можно выбирать тип регистрируемых событий, а для упорядочивания отображения можно использовать фильтр.

Помимо журнала событий на DNS-серверах ведется отдельный журнал DNS. Для активации записи в этот журнал необходимо в настройках DNS-сервера смонтировать функции записи пакетов в журнал и выбрать типы пакетов, в зависимости от их направления движения, содержания, используемого транспортного протокола.

Устранять ошибки репликации данных в DNS-зонах, интегрированных в доменах, служит утилита Replication monitor. Данная утилита устанавливается дополнительно и входит в пакет средств по поддержке Windows Replmon.

Производительность DNS-сервера контролируется с помощью утилиты Системный монитор и ряда счетчиков. Всего есть 62 счетчика, относящихся к производительности DNS, в том числе:

- счетчик общей статистики;
- счетчик TCP и UDP;
- счетчик использования памяти;
- счетчик рекурсивного поиска;
- счетчик зонных передач.

6.6. NetBios и служба WINS

Пространство имен NetBios не базируется ни на какой иерархии, это простой линейный список имен работающих на компьютере служб. Имена состоят из 15 видимых символов и 16-го служебного. Если видимых символов меньше 15, то оставшиеся символы заполняются байтом нулей. 16-й символ соответствует службе, работающей на компьютере с данным именем.

Просмотреть список пространства NetBios можно с помощью команды nbtstat –n. Рассмотрим пример на рис. 6.4. На рисунке изображен вывод команды «*nbtstat –n*» на сервере *dc1.world.ru*, являющийся списком NetBIOS-имен, сгенерированных данным сервером.

В угловых скобках указан шестнадцатеричный код 16-го служебного символа.

DCI с кодом <00> соответствует службе *Рабочая станция*, которая выполняет роль клиента при подключении к ресурсам файлов или печати, представляемых другими компьютерами.

Код <20> на компьютере DCI соответствует службе *Сервер*, который предоставляет ресурсы для другого компьютера сети.

Процесс разрешения имен в пространстве NetBios может быть выполнен одним из трех способов:

1. Широковещательный запрос.
2. Обращение к локальной базе данных NetBios-имен (LMhosts), хранящихся в той же папке, что и файл hosts, отображающий FQDN-имена.

3. Обращение к централизованной базе данных имен NetBios, хранящихся на сервере WINS.

```
C:\nbtstat -n

Подключение по локальной сети:
Адрес IP узла: [192.168.0.1] Код области: []

                Локальная таблица NetBIOS-имен

       Имя                Тип          Состояние
    ---------------------------------------------------
DC1            <00>  Уникальный  Зарегистрирован
WORLD<00>  Группа      Зарегистрирован
WORLD<1C>  Группа      Зарегистрирован
DC1            <20>  Уникальный  Зарегистрирован
WORLD<1B>  Уникальный  Зарегистрирован
WORLD<1E>  Группа      Зарегистрирован
WORLD<1D>  Уникальный  Зарегистрирован
..__MSBROWSE__.<01>  Группа      Зарегистрирован
```

Рис. 6.4. Пример работы утилиты nbtstat

В зависимости от типа узла NetBios, разрешение имен осуществляется с помощью различных комбинаций перечисленных способов.

Выделяют четыре типа узла:

- b-узел (broadcast node, широковещательный) – разрешает имена в IP-адресах посредством широковещательных сообщений broadcast node;
- p-узел (peer node) – разрешает имена в IP-адреса с помощью WINS-сервера;
- m-узел(mixednode, смешанный узел) – комбинирует запросы b- и p-узлов, первоначально пытается применить широковещательный запрос, а в случае неудачи обращается к WINS-серверу;
- h-узел (hybrid node, гибридный) – комбинирует запросы b- и p-узлов, но при этом сначала обращается к WINS-серверу, а при неудаче выполняет широковещательную рассылку.

Наиболее эффективным является h-узел.Тип узла определяется следующим образом: если в свойствах протокола TCP/IP нет адреса WINS-сервера, то данный компьютер считается b-узлом, в противном случае он является h-узлом. Использование других типов узлов настраивается через реестр Windows. В больших сетях для распределения нагрузки по регистрации и разрешению имен NetBios необходимо использовать несколько WINS-серверов.

Считается, что один WINS-сервер должен обслуживать порядка нескольких сотен компьютеров. При использовании нескольких серверов часть клиентов настраивается на регистрацию и разрешение имен на один WINS-сервер, вторая – на другой, а между серверами, по аналогии с системойDNS, настраивается репликация.

Контрольные вопросы

1. Для чего необходимы доменные имена?
2. Для чего нужна служба DNS?
3. Что такое корневой домен?
4. Каково было предназначение файла hosts?
5. Чем отличается служба DNS от системы имен DNS?
6. Объясните принцип действия итеративного запроса.
7. Объясните принцип действия рекурсивного запроса.
8. В чем отличие доменных имен от имен NetBIOS?
9. Опишите принципы разрешения NetBios имен.
10. Назначение утилиты NSLOOKUP. Примеры ее использования.
11. Какие символы разрешены в DNS-именах?
12. Опишите БД DNS.
13. Как реализовано текстовое представление БД DNS?

ТЕМА 7. СЛУЖБА КАТАЛОГА ACTIVE DIRECTORY. ПЛАНИРОВАНИЕ ACTIVE DIRECTORY. ПРОСТРАНСТВО ИМЕН DNS

План

1. **Понятие Active Directory. Служба Active Directory.**
2. **Структура каталога Active Directory.**
3. **Объекты каталога и их наименования.**
4. **Иерархия доменов.**
5. **Доверительные отношения между доменами.**
6. **Организационные подразделения.**

Данная тема рассчитана на две лекции (лекции 14–15).

7.1. Понятие Active Directory. Служба Active Directory

Ранее отмечалось, что в средних и крупных сетях задача настройки параметров протокола TCP/IP является очень сложной для администратора и вручную практически невыполнима. Для решения этой проблемы был разработан протокол DHCP, реализованный посредством службы DHCP.

Однако настройка сетевых параметров – лишь одна из множества задач, встающих перед системным администратором. В частности, в любой сети важнейшей является задача управления ресурсами (файлами и устройствами, предоставленными в общий доступ), а также компьютерами и пользователями.

Для решения задач управления ресурсами в сетях под управлением Windows Server применяется служба каталога Active Directory (Активный Каталог). Данная служба обеспечивает доступ к базе данных (каталогу), в которой хранится информация обо всех объектах сети, и позволяет управлять этими объектами.

Группа компьютеров, имеющая общий каталог и единую политику безопасности, называется *доменом* (domain). Под политикой безопасности понимают набор правил по применению средств обеспечения сетевой безопасности: паролей, учетных записей, протоколов аутентификации и защищенной передачи информации, шифрованной файловой системы и т. д.

Каждый домен имеет один или несколько серверов, именуемых *контроллерами домена* (domain controller), на которых хранятся копии каталога.

Основные преимущества, предоставляемые службой каталога Active Directory:

- централизованное управление: если в сети развернута служба Active Directory, системный администратор может выполнять большинство своих задач, используя единственный компьютер – *контроллер домена;*
- простой доступ пользователей к ресурсам: пользователь, зарегистрировавшись в домене на произвольном компьютере, может получить доступ к любому ресурсу сети при условии наличия соответствующих прав;
- обеспечение безопасности: служба Active Directory совместно с подсистемой безопасности Windows Server 2003 предоставляет возможность гибкой настройки прав пользователей на доступ к ресурсам сети;
- масштабируемость – это способность системы повышать свои размеры и производительность по мере увеличения требований к ним. При расширении сети организации служба каталога Active Directory способна наращивать свои возможности – увеличивать размер каталога и число контроллеров домена.

Таким образом, служба каталога Active Directory, подобно службе DHCP, существенно облегчает работу системного администратора по управлению сетевыми объектами. Кроме того, пользователи получают возможность использовать ресурсы сети, не заботясь об их месторасположении, так как все запросы обрабатываются службой Active Directory.

7.2. Структура каталога Active Directory

Вся информация об объектах сети содержится в каталоге Active Directory. Физически эта база данных представляет собой файл **Ntds.dit,** который хранится на контроллере домена.

Каталог ActiveDirectory может рассматриваться с двух позиций: с точки зрения логической структуры и с точки зрения физической структуры.

Логическая структура каталога Active Directory представлена на рис. 7.1. Цель такой структуризации – облегчение процесса администрирования.

Все сетевые объекты (пользователи, группы пользователей, компьютеры, принтеры) объединяются в домен, который является основной структурной единицей каталога. Для удобства управления объекты также могут быть сгруппированы при помощи *организационных подразделений (ОП).* Несколько иерархически связанных доменов образуют *дерево доменов.* Совокупность деревьев, имеющих общие части каталога ActiveDirectory и общих администраторов, называется *лесом доменов.*

Имея возможность такой логической структуризации, администратор может планировать и выбирать конфигурацию сети в зависимости от своих задач и масштабов организации.

Основной целью *физической структуризации* каталога Active Directory является оптимизация процесса копирования изменений, произведенных на одном из контроллеров домена, на все остальные контроллеры. Этот процесс называется *репликацией* (replication).

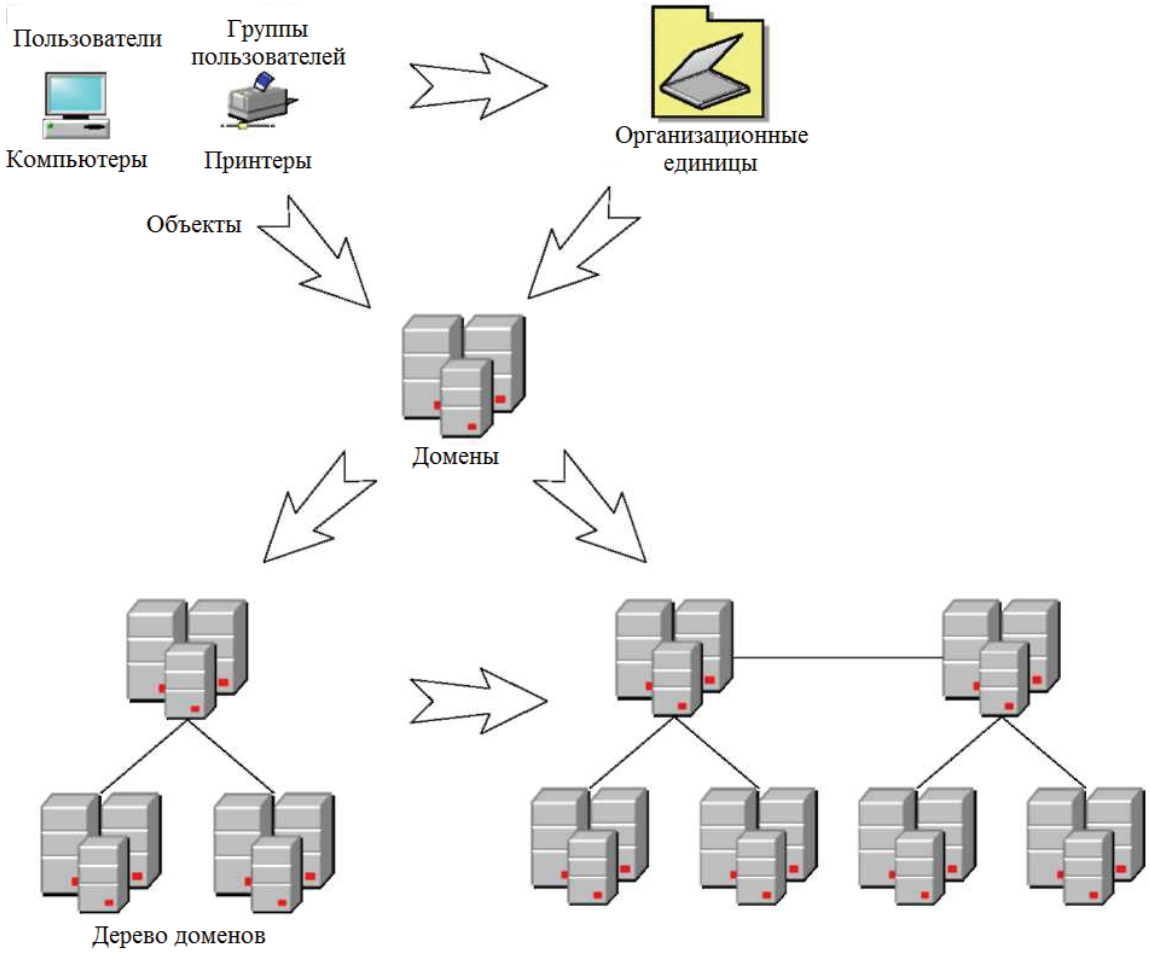


Рис. 7.1. Пример логической структуризации домена

Основой физической структуры является *сайт* (site) – это часть сети, все контроллеры домена которой связаны высокоскоростным соединением. Между сайтами, наоборот, установлены более медленные линии связи (рис. 7.2).

Подобная структура позволяет планировать процесс репликации следующим образом: внутри сайта репликация осуществляется часто и могут передаваться большие объемы информации без сжатия; между сайтами изменения реплицируются редко, и данные требуется сжимать.

Логическая и физическая структуры предназначены для решения разных задач и поэтому между собой практически не связаны: в одном домене может быть несколько сайтов, так же как один сайт может содержать несколько доменов. Общим объектом для той и другой структуры является контроллер домена с хранящимся на нем файлом каталога **Ntds.dit**.

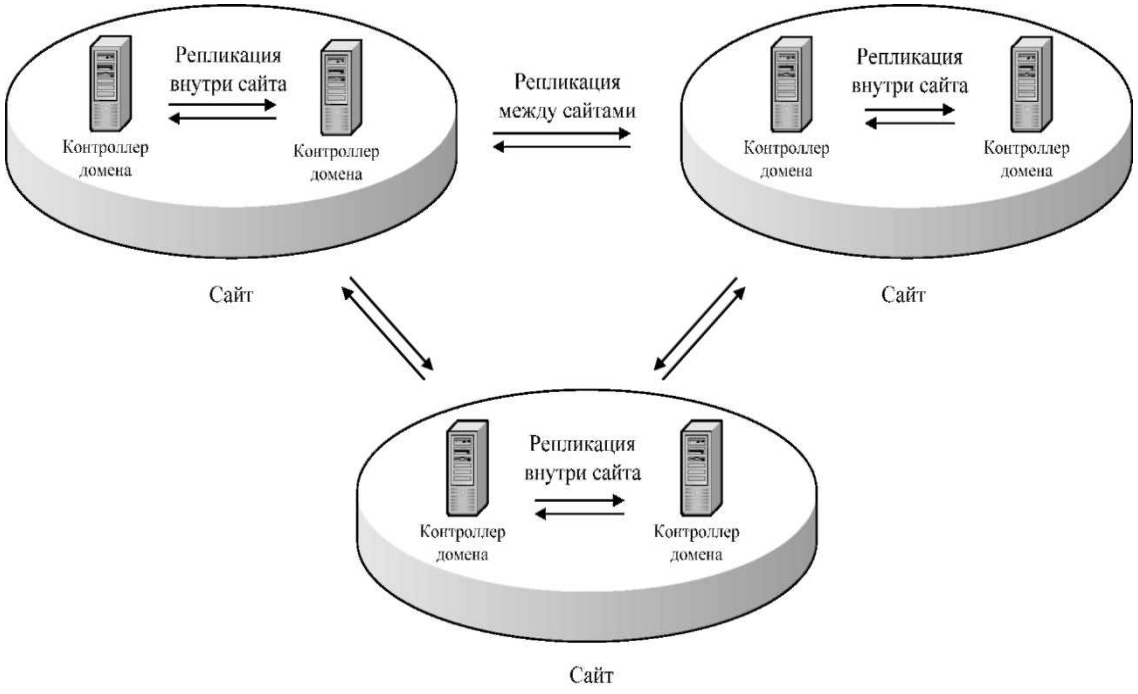


Рис. 7.2. Пример физической структуризации домена

В файле каталога Active Directory содержится информация как о логической, так и о физической структурах. Этот файл состоит из нескольких разделов:

- раздел домена (domain partition) – содержатся данные обо всех объектах домена (пользователях, компьютерах, принтерах и т. д.);

• раздел схемы (schema partition) – хранится информация о типах всех объектов, которые могут быть созданы в данном лесе доменов;

• раздел конфигурации (configuration partition) – описывается конфигурация леса доменов – информация о сайтах, соединениях между сайтами и направлениях репликации;

• раздел приложений (application partition) – специальный раздел для хранения данных приложений, не относящихся к службе Active Directory. По умолчанию здесь создается подраздел для службы DNS;

• раздел глобального каталога (global catalog partition). *Глобальный каталог* – это база данных, в которой содержится список всех объектов леса доменов без информации об атрибутах этих объектов. Глобальный каталог необходим для поиска ресурсов леса в любом принадлежащем ему домене.

В зависимости от принадлежности к разделу, информация реплицируется между контроллерами доменов следующим образом:

- раздел домена реплицируется между контроллерами одного домена;
- разделы схемы, конфигурации и глобального каталога реплицируются на все контроллеры леса;
- репликацией раздела приложений можно управлять – указывать, какие контроллеры будут получать реплику данного раздела.

7.3. Объекты каталога и их наименования

Объект каталога Active Directory – это элемент, содержащийся в базе данных Active Directory и имеющий набор атрибутов (характеристик). Например, объектом является пользователь, а его атрибутами – имя, фамилия и адрес электронной почты.

Некоторые объекты являются контейнерами. Это означает, что данные объекты могут содержать в своем составе другие объекты. Например, объект *домен* является контейнером и может включать пользователей, компьютеры, другие домены и т. д.

Каталог Active Directory содержит следующие основные типы объектов, не являющихся контейнерами:

- пользователь (user);
- группы пользователей (group);
- контакты (contact);
- компьютеры (computer);

- принтеры (printer);
- общедоступные папки (shared folder).

В Active Directory для именования объектов используется несколько способов.

Различающееся имя (Distinguished Name, DN) – состоит из нескольких частей, например, для пользователя **Петрова,** принадлежащего к организационному подразделению **Teachers** домена **faculty.ru,** различающееся имя выглядит так:

DC = ru, DC = faculty, OU = teachers, CN = users, CN = petrov.

При этом используются следующие сокращения:

- DC (Domain Component) – домен;
- OU (Organizational Unit) – организационное подразделение;
- CN (Common Name) – общее имя.

Различающиеся имена являются уникальными в пределах всего каталога Active Directory. В целях упрощения именования может использоваться *относительное различающееся имя* (Relative Distinguished Name, RDN). Для приведенного примера это имя **CN = petrov.** Имя RDN должно быть уникально в рамках объекта-контейнера, т. е. в пределах контейнера **CN = users** пользователь **petrov** должен быть единственным.

Основное имя пользователя (User Principal Name, UPN) – используется для входа пользователя в систему и состоит из двух частей: имени учетной записи пользователя и имени домена, к которому принадлежит пользователь. Например: **petrov@faculty.ru.**

Глобальный уникальный идентификатор (Global Unique Identifier, GUID) – это 128-битовое шестнадцатеричное число, которое ассоциируется с объектом в момент его создания и никогда не меняется. В случае перемещения или переименования объекта, его GUID остается прежним.

7.4. Иерархия доменов

Домен является основным элементом в логической структуре Active Directory. В рамках домена действуют единые административные полномочия и политика безопасности, применяется общее пространство доменных имен.

Каждый домен имеет по крайней мере один контроллер домена, на котором хранится каталог Active Directory с информацией о домене.

Для организаций со сложной структурой может создаваться иерархия доменов. Первый образованный домен называется *корневым* (root domain). У него могут быть дочерние домены, имеющие общее пространство доменных имен. В свою очередь, у дочерних доменов могут быть свои домены-потомки. Таким образом, создается иерархия доменов, называемая *доменным деревом* (domain tree).

Если требуется в рамках одной организации создать еще одно пространство имен, то создается отдельное дерево доменов. При этом несколько деревьев, входящих в состав одного каталога Active Directory, образуют *лес доменов* (forest).

Для именования доменов используются правила, принятые в системе доменных имен DNS. Вследствие этого доменная структура организации может при необходимости (и при соблюдении требования уникальности имен) встраиваться в доменную структуру Интернета. Кроме того, для разрешения доменных имен становится возможным использование службы DNS.

На рис. 7.3 приведен фрагмент доменной структуры университета.

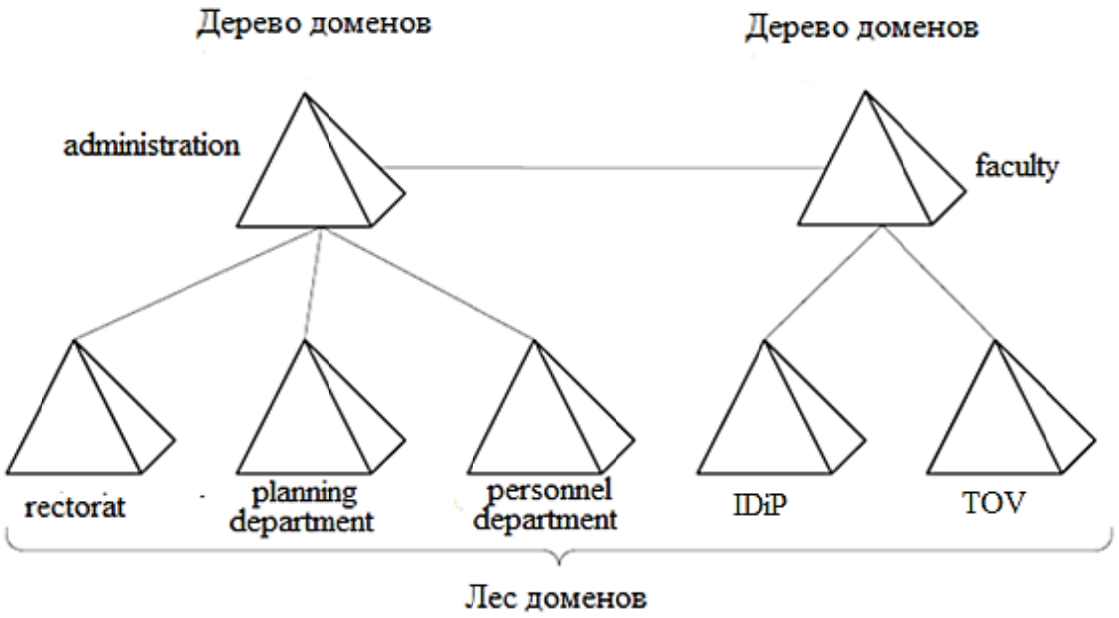


Рис. 7.3. Фрагмент возможной доменной структуры вуза

В данном примере лес состоит из двух деревьев: дерева управления университета (домен **administration)** и дерева факультетов (домен

faculty). Корневой домен головной организации имеет три дочерних домена: **rectorat** (ректорат), planning department (плановый отдел), personal department(отдел кадров). Корневой домен **faculty** является родителем для двух доменов – **IDiP**(ИДИП) и **TOV**(ТОВ).

Следуя правилам DNS, полное имя (FQDN) домена **rectorat** будет иметь следующий вид: **rectorat.administration,** а полное имя домена ИДиП**: IDiP.faculty**.

7.5. Доверительные отношения между доменами

Для доступа к ресурсам своего домена пользователю достаточно ввести имя своей учетной записи и пройти процедуры аутентификации и авторизации. *Аутентификация* (authentication) – это процесс проверки подлинности пользователя, т. е. подтверждение того, что пользователь является тем, за кого себя выдает. Аутентификация в Windows Server 2003 осуществляется путем предъявления системе пароля. В случае успешной аутентификации наступает этап *авторизации* (authorization) – определение набора прав, которыми обладает пользователь.

При наличии необходимых прав (подробнее о правах доступа – в следующей лекции) пользователь может получить доступ к любому ресурсу домена. Однако для доступа к ресурсам другого домена между доменами должны быть установлены *доверительные отношения* (trust relationship).

Существует два вида доверительных отношений: *односторонние* (one-way trust relationship) и *двусторонние* (two-way trust relationship). Односторонние доверительные отношения означают, что пользователь одного домена (доверенного, trusted domain) получает доступ к ресурсам другого домена (доверяющего, trusting domain), но обратное неверно. Иначе говоря, доверяющий домен делегирует право аутентификации пользователей доверенному домену.

Двусторонние доверительные отношения предполагают обоюдный процесс делегирования права аутентификации.

При создании доменной структуры некоторые доверительные отношения устанавливаются автоматически, другие приходится настраивать вручную.

Перечислим автоматически устанавливаемые двусторонние доверительные отношения:

- внутри дерева доменов;

- между корневыми доменами деревьев одного леса;
- между деревьями одного леса (эти отношения являются следствием доверительных отношений между корневыми доменами деревьев).

В остальных случаях доверительные отношения следует устанавливать вручную (например, между лесами доменов или между лесом и внешним доменом, не принадлежащим этому лесу).

7.6. Организационные подразделения

Структурирование сетевых ресурсов организации при помощи доменов не всегда бывает оправданно, так как домен подразумевает достаточно крупную часть сети. Часто для администратора возникает необходимость группировки объектов внутри одного домена. В этом случае следует использовать *организационные подразделения* (organizational unit).

Организационные подразделения можно использовать в качестве контейнера для следующих объектов:

- пользователей;
- групп пользователей;
- контактов;
- компьютеров;
- принтеров;
- общих папок;
- других организационных подразделений.

Объекты группируются с помощью ОП для следующих целей:

- управление несколькими объектами как одним целым – для этого используются групповые политики (см. подраздел 8.5);
- делегирование прав администрирования, например, начальнику отдела можно делегировать административные права на его отдел при условии объединения всех объектов отдела в организационную единицу.

В качестве примера структуризации с использованием ОП можно привести возможную структуру домена факультета ИДиП (см. рис. 7.4). В данной ситуации выделение из домена **IDiP** дочерних доменов кафедр (ISiT, POiSOI) не имеет смысла, так как факультет слишком мал. С другой стороны, требуется отразить в Active Directory внутреннюю структуру факультета.

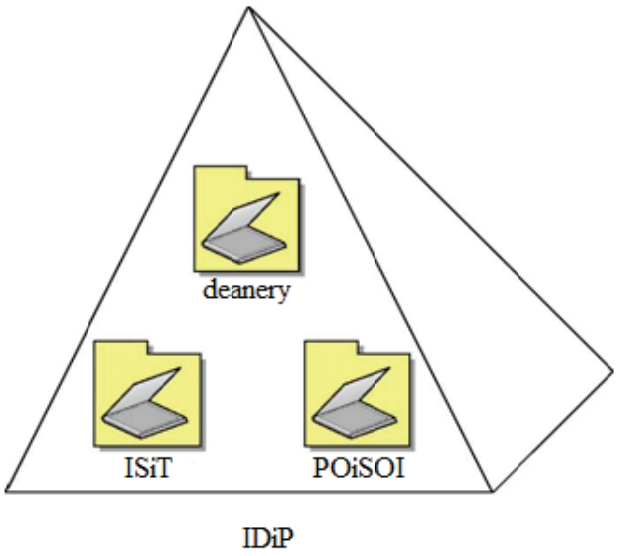


Рис. 7.4. Домен факультета FPP (ИДиП)

Решением является структуризация с применением организационных подразделений – в домене создаются ОП deanery (деканат) и кафедр: ISiT (ИСиТ) и POiSOI (ПОиСОИ). При этом для каждого подразделения администратор может назначить собственный набор правил (например, общие требования к паролям).

Контрольные вопросы

1. Какая информация хранится в каталоге Active Directory? Где находится сам каталог?
2. Что такое домен?
3. Чем отличается контроллер домена от других узлов сети?
4. Какова цель логической структуризации каталога Active Directory?
5. По какому принципу следует осуществлять деление на сайты?
6. Для чего нужна репликация?
7. Сколько всего может быть создано глобальных идентификаторовGUID?
8. Чем аутентификация отличается от авторизации?
9. Объясните понятия «доверенный» и «доверяющий» домен. В каком случае один домен может быть доверенным и доверяющим одновременно?
10. Для чего используют организационные подразделения?

ТЕМА 8. ПЛАНИРОВАНИЕ И УПРАВЛЕНИЕ ACTIVE DIRECTORY

План

1. **Планирование Active Directory.**
2. **Планирование пространства имен Active Directory.**
3. **Учетные записи пользователей.**
4. **Группы пользователей.**
5. **Групповые политики.**

Данная тема рассчитана на две лекции (лекции 16–17).

8.1. Планирование Active Directory

Успешная работа пользователей сетевых ресурсов, а также служб, реализующих протоколы TCP/IP, зависит от правильного функционирования Active Directory. Поэтому крайне важной становится задача планирования структуры каталога Active Directory. Удачно спроектированный каталог позволит сделать работу сети более эффективной и стабильной, а также намного облегчит труд администратора.

В процессе планирования Active Directory можно выделить два основных этапа (рис. 8.1):

1) планирование логической структуры, включающее проектирование доменов и организационных подразделений, а также проблему именования;

2) планирование физической структуры, состоящее из разделения сети на сайты и размещения контроллеров домена.

8.1.1. Планирование логической структуры

При планировании доменной структуры нужно определить количество и способ организации доменов. Возможны три варианта: единственный домен, дерево доменов или лес. Критерии выбора следующие.

1. Размер организации: один домен может содержать от нескольких до сотен тысяч пользователей, не рекомендуется допускать превышение данного условного предела.

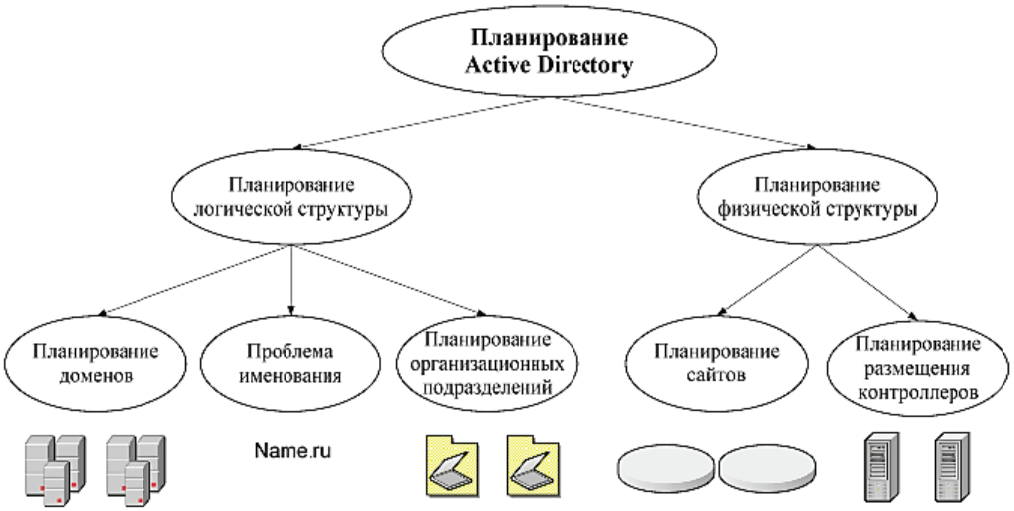


Рис. 8.1. Планирование Active Directory

2. Географическое расположение: имеются ли у организации филиалы или отделы, находящиеся на большом расстоянии и связанные с центральным звеном низкоскоростными каналами связи. Наличие таких филиалов при единственном в организации домене, скорее всего, вызовет перегрузку линий связи из-за трафика репликации.

3. Стабильность организации: насколько высока подвижность кадрового состава, не планируется ли в ближайшее время структурное преобразование организации.

4. Потребности в разных доменных именах: в некоторых случаях в рамках одной организации требуются разные доменные имена. Например, в случае создания единой компьютерной системы двух университетов каждый из них, вероятно, захочет иметь свое собственное доменное имя.

5. Способ управления сетью: может быть централизованным и децентрализованным. Централизованный способ предполагает сосредоточение всей административной власти у одного коллектива администраторов и наличие однодоменной модели. При децентрализованном способе полномочия делегируются нескольким слабосвязанным удаленным группам администраторов, управляющих доменами дерева или леса.

6. Единство политики безопасности. Чаще всего политика безопасности в одной организации едина для всех отделов и сотрудников, однако бывают исключения.

Исходя из перечисленных критериев, можно выделить те признаки, по которым выбирается вариант с одним доменом:

1) в организации менее сотни тысяч пользователей;

2) отсутствие удаленных филиалов;
3) относительная стабильность структуры организации;
4) отсутствие потребности в разных доменных именах;
5) централизованный способ администрирования;
6) единая политика безопасности.

Отсутствие первых четырех признаков существенно склоняет к выбору в пользу многодоменной модели. Последние два признака в меньшей степени должны влиять на выбор, так как задачи делегирования, администрирования и разделения политик безопасности можно решить средствами организационных подразделений в рамках одного домена.

При выборе модели с несколькими доменами в большинстве ситуаций нужно использовать дерево доменов. Лес доменов приемлем в том случае, когда две независимые организации хотят иметь общие сетевые ресурсы.

После выбора доменной структуры следует продумать *имена для создаваемых доменов.* Особенно важно имя корневого домена. Правил для выбора доменного имени немного: во-первых, оно должно отражать специфику организации, во-вторых, быть понятным всем пользователям ресурсов домена, а не только администратору и, в-третьих, не должно быть слишком сложным. Для имени очень часто используют аббревиатуры, например bstu и т. д. Планирование пространства имен доменов с точки зрения безопасности рассмотрено в 8.2.

Планирование структуры организационных подразделений в каждом домене является важным шагом.

Как отмечалось в предыдущей теме, ОП применяются в том случае, если для задач управления группой объектов или делегирования административных прав образование новых доменов нецелесообразно.

В связи с тем, что организационные подразделения можно использовать в качестве контейнеров, допускается строить иерархию ОП с несколькими уровнями вложений.

Иерархию можно строить с помощью двух основных подходов: либо следуя организационной структуре предприятия *(организационный подход),* либо исходя из задач управления сетевыми объектами *(административный подход).* Оба способа используются на практике, и задача администратора состоит в том, чтобы выяснить, какой из подходов (или их комбинация) применим в данной ситуации.

8.1.2. Планирование физической структуры

Основная цель планирования физической структуры – оптимизация трафика репликации. Цель достигается путем продуманного расположения сайтов и контроллеров домена.

Основной объем данных репликации присутствует в рамках одного домена, междоменный же трафик репликации существенно ниже внутридоменного. Для оптимизации процесса репликации рекомендуется использовать механизм сайтов.

На начальном этапе следует проанализировать существующую сеть: ее структуру, количество пользователей и компьютеров, пропускную способность, колебания трафика. Все эти данные нужно учитывать при планировании. Чем больше пользователей и компьютеров в сети, тем больше объем передаваемой информации при репликации. Линии с большой пропускной способностью могут быть сильно загружены, и большой трафик репликации внесет существенные проблемы, в то время как низкоскоростные каналы, возможно, практически свободны и выдержат дополнительный объем данных репликации.

Во время анализа следует учитывать возможность расширения сети и увеличения числа пользователей. Считается достаточным принимать коэффициент расширения в пределах 30–50%.

Основной критерий при выделении сайтов – пропускная способность линий связи. Части домена, связанные высокоскоростными линиями, помещаются в один сайт. Если между частями домена имеются каналы с низкой скоростью передачи данных, их следует разместить в разных сайтах. При этом трафик межсайтовой репликации сжимается и его передача происходит во время наименьшей загрузки низкоскоростных линий.

Вопрос о необходимом количестве и размещении контроллеров домена решается тогда, когда известна доменная структура и расположение сайтов. Общее правило таково, что для каждого домена необходимо не менее двух контроллеров (при этом в случае отказа одного из контроллеров второй обеспечит работу сети). Количество контроллеров зависит от числа пользователей (следовательно, от числа обращений на контроллеры домена), принадлежащих данному домену или сайту. Например, если домен включает два сайта, связанных модемной линией, и одному из сайтов принадлежит всего несколько пользователей, то совсем не обязательно в этом сайте располагать отдельный контроллер домена (при условии, что загрузка модемной линии невысока).

8.2. Планирование пространства имен Active Directory

При планировании имен доменов верхнего уровня можно использовать различные стратегии и правила, но необходимо учитывать вопросы интеграции внутреннего пространства и пространства имен сети Интернет.

Существует несколько базовых подходов при планировании пространства имен.

1. Один домен, одна зона DNS.

На рисунке (рис. 8.1) в левой части – внутренняя сеть компании, справа – сеть Интернет, две сети разделены маршрутизатором «*R*» (кроме маршрутизатора, на границе могут быть также прокси-сервер или межсетевой экран).

В данном примере используется одна и та же зона DNS (*company.ru*) как для поддержки внутреннего домена AD с тем же именем (записи DC, SRV-1, SRV-2, WS-1), так и хранения ссылок на внешние ресурсы компании: веб-сайт, почтовый сервер (записи www, mail).

Такой способ максимально упрощает работу системного администратора (по настройкам), но при этом DNS-сервер, доступный для всей сети Интернет, хранит зону *company.ru* и предоставляет доступ к записям этой зоны всем пользователям Интернета. Как следствие, можно достаточно легко получить информацию о структуре внутри сети, что может привести к проблемам безопасности.

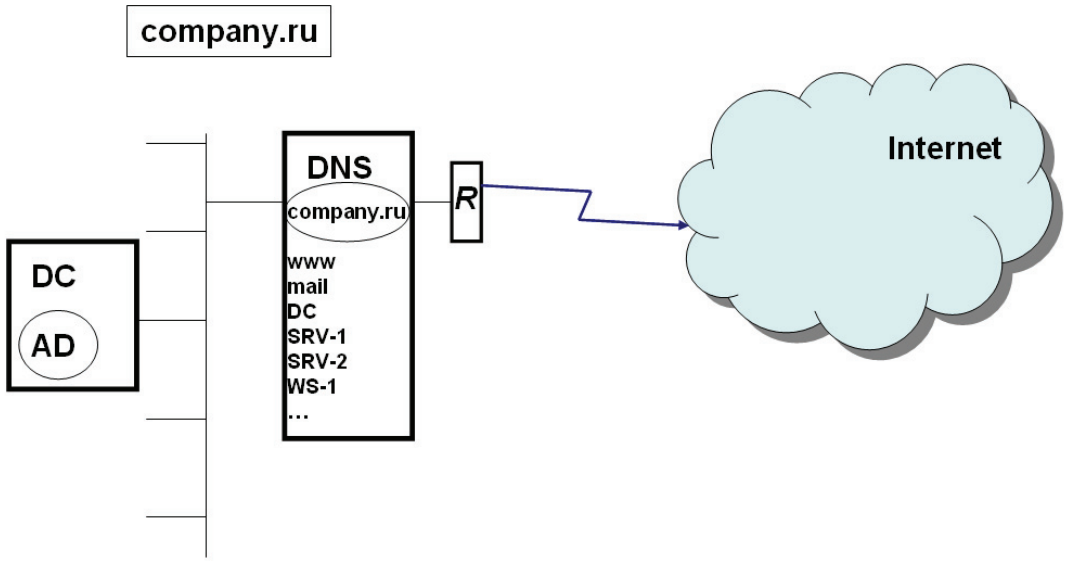


Рис. 8.1. Общая схема планирования имен типа «один домен, одна зона DNS»

2. Одно имя домена, две зоны DNS.

В данном случае на различных серверах DNS (рис. 8.2) создаются различные зоны с одним и тем же именем company.ru. На внутреннем DNS-сервере функционирует зона company.ru для Active Directory, на внешнем – для ссылок на внешние ресурсы. Данные зоны никак между собой не связаны – ни механизмами репликации, ни ручной синхронизацией.

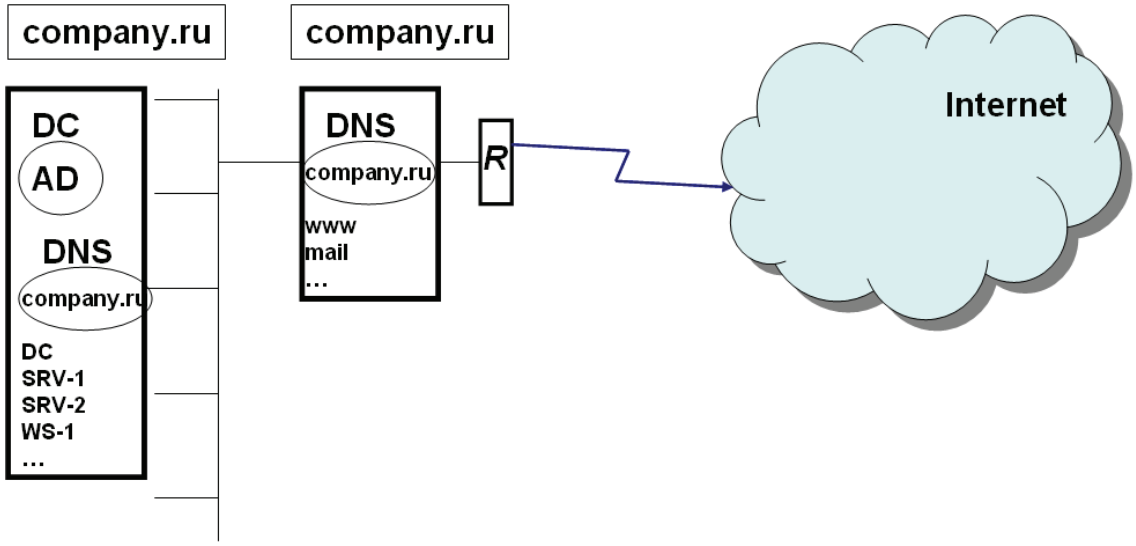


Рис. 8.2. Общая схема планирования имен
типа «одно имя домена, две зоны DNS»

Данный вариант несложно реализовать, но для сетевого администратора возникает нагрузка управления двумя разными доменами с одним именем. С точки зрения безопасности вариант является достаточно надежным.

3. «Расщепление» пространства имен DNS.

В данном примере корневой домен компании *company.ru* служит для хранения ссылок на внешние ресурсы (рис. 8.3). В нем же настраивается делегирование управления поддоменом *corp.company.ru* на внутренний DNS-сервер, и именно на базе домена *corp.company.ru* создается домен Active Directory.

В данном случае во внешней зоне хранятся ссылки на внешние ресурсы, а также ссылка на делегирование управления поддоменом на внутренний DNS-сервер. Пользователям сети Интернет доступен минимум информации о внутренней сети. Такой вариант организации пространства имен довольно часто используется компаниями.

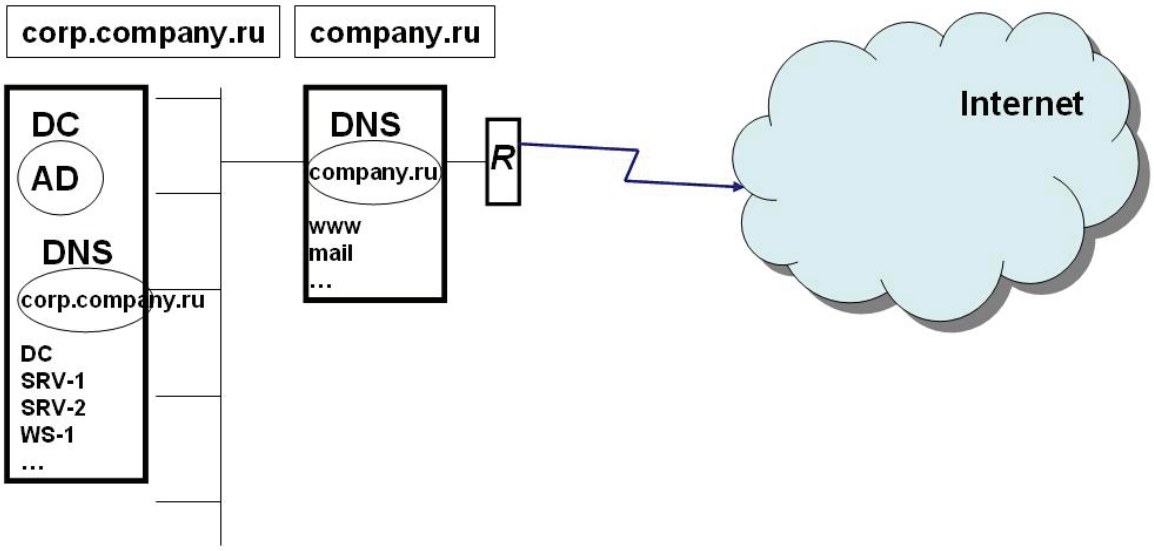


Рис. 8.3. Общая схема планирования имен
типа «расщепление пространства имен»

4. Два различных домена DNS для внешних ресурсов и Active Directory.

В данном сценарии предполагается регистрация двух отдельных доменных имен (рис. 8.4): одно для публикации внешних ресурсов, другое – для развертывания Active Directory. Данная схема является достаточно надежной, но требует дополнительных средств, исключает возможные в дальнейшем проблемы появления конфликтов с другими компаниями.

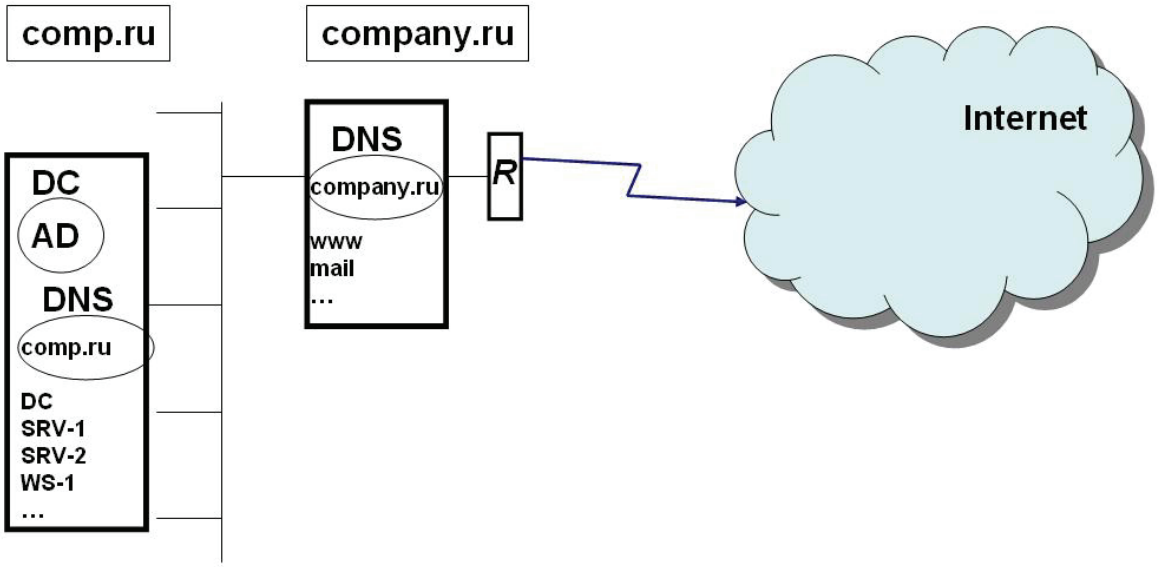


Рис. 8.4. Общая схема планирования имен
типа «два различных домена»

8.3. Учетные записи пользователей

После реализации спроектированной структуры Active Directory администратор должен добавить в каталог учетные записи всех пользователей системы и назначить каждой из них определенные права. *Учетная запись пользователя* – это набор атрибутов, сопоставленных с определенным пользователем. Самые важные атрибуты следующие:

- имя учетной записи, с помощью которого пользователь осуществляет вход в систему (в пределах домена должно быть уникальным);
- полное имя пользователя;
- пароль;
- группы, в которые входит пользователь;
- права пользователя.

Создав все необходимые учетные записи, администратору следует продумать, какими правами должен обладать тот или иной пользователь. *Права пользователя* – это список действий, которые может выполнять пользователь. Права бывают следующих видов:

- *привилегия* (privilege) – право выполнения операций по изменению состояния или параметров системы (например, выключение компьютера или изменение системного времени);
- *право на вход в систему* (logon right);
- *разрешение доступа* (access permission) – право осуществления действий с файлами, папками, принтерами, объектами Active Directory, реестром (при условии, что используется файловая система NTFS).

При условии, что количество пользователей составляет порядка десяти человек, определить необходимые права достаточно просто. Однако гораздо чаще на практике встречаются компьютерные системы с сотнями и тысячами учетных записей. В таких масштабах задача распределения прав отдельным пользователям становится невыполнимой. В этом случае на помощь администратору приходит механизм групп пользователей.

8.4. Группы пользователей

Группа пользователей (группа безопасности, Security Group) – это объединение учетных записей пользователей, которому можно назначать права. С использованием групп распределение прав осуществляется следующим образом. Сначала выбираются такие пользователи, спи-

сок прав которых должен быть одинаковым. Затем создается группа, членами которой являются выбранные пользователи. Требуемые права назначаются уже не отдельным пользователям, а группе, и эти права автоматически распространяются на всех пользователей группы.

Следует отметить, что группы пользователей и организационные подразделения представляют собой разные механизмы, предназначенные для различных целей. Создание групп безопасности преследует цель распределения прав доступа к ресурсам пользователям сети, в то время как основное назначение организационных подразделений – управление пользователями (а также компьютерами) (рис. 8.5).

Рис. 8.5. Использование ОП и групп безопасности

Группы пользователей различаются по области действия. Выделяют три области действия:

- *доменную локальную* (domain local scope);
- *глобальную* (global scope);
- *универсальную* (universal scope).

Доменные локальные группы действуют только в рамках своего домена. За его пределами указывать локальную доменную группу нельзя. Такие группы обычно применяются для управления доступом к файлам, общим папкам и принтерам.

Глобальные группы могут использоваться в рамках всего леса доменов. Однако глобальная группа принадлежит определенному домену, и в ее состав могут входить только объекты этого домена.

Применяются глобальные группы в том случае, если пользователям одного домена нужно получить доступ к ресурсам другого домена.

Универсальные группы привязаны к корневому домену леса, но в их состав могут входить пользователи любого домена. Чаще всего универсальные группы используются для объединения глобальных групп.

8.5. Групповые политики

В заключение рассмотрим один из наиболее эффективных и удобных инструментов администрирования – групповые политики.

Групповые политики (group policy) – это способ автоматизации работы по настройке рабочих столов пользователей и параметров компьютеров.

Групповые политики представляют собой наборы правил конфигурирования, применяемых к компьютеру или пользователю. Каждый такой набор правил называется *объектом групповой политики* (Group Policy Object, GPO).

Один или несколько объектов групповой политики могут применяться к трем видам объединений:

- сайтам;
- доменам;
- организационным подразделениям.

Кроме того, для каждого компьютера может быть определен *объект локальной групповой политики* (Local Group Policy Object, LGPO).

Объекты групповых политик являются наследуемыми. Это означает, что, например, GPO, применяемый к домену, наследуется всеми его организационными подразделениями. В том случае, если правила одного объекта групповой политики конфликтуют с правилами другого, наибольший приоритет имеет GPO организационного подразделения, ниже по уровню GPO домена, затем следует GPO сайта, а наименьший приоритет у LGPO. Приведем краткий обзор возможностей, предоставляемых групповыми политиками (рис. 8.6).

Объект групповой политики содержит две основные части:

- конфигурация компьютера (Computer Configuration);
- конфигурация пользователя (User Configuration).

Каждая из частей включает три раздела:

- настройки приложений (Software Settings);
- настройки Windows (Windows Settings);
- административные шаблоны (Administrative Templates).

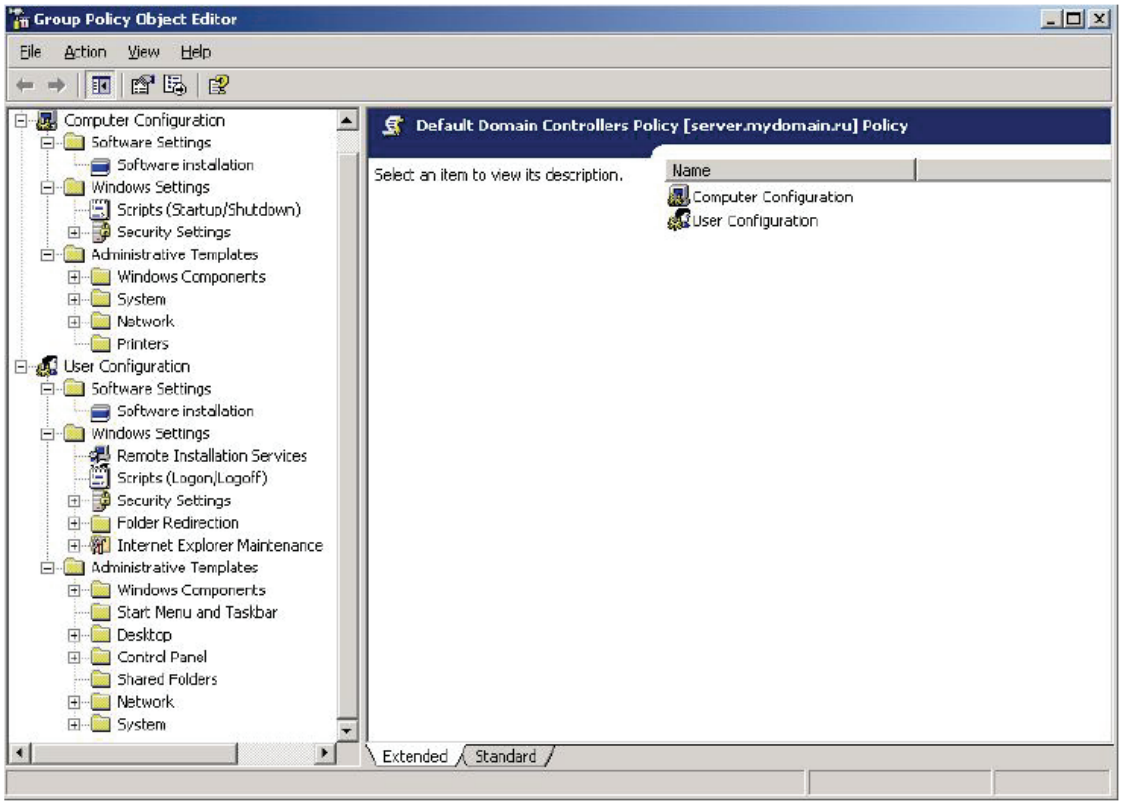


Рис. 8.6. Пример окна с настройками групповых политик

В разделе **Настройки приложений** находится подраздел **Установка приложений (Software Installation),** позволяющий автоматически устанавливать выбранные программы на компьютеры пользователей.

Правила, создаваемые в разделе **Настройки Windows,** позволяют:

- выполнять задаваемые сценарии **(Scripts)** при включении-выключении компьютера, при входе пользователя в систему и выходе из нее;
- настраивать параметры безопасности **(Security Settings)** компьютера и пользователя (требования к паролям, доступ к реестру, политику аудита событий);
- конфигурировать Internet Explorer **(Internet Explorer Maintenance);**
- изменять места расположения папок пользователей **(Folder Redirection).**

Раздел **Административные шаблоны** предназначен для настройки рабочего стола пользователя, ограничения доступа к системным компонентам и компонентам приложений.

Таким образом, Windows Server предоставляет мощный набор инструментов администрирования, способствующий эффективному управлению сетью любой организации.

Контрольные вопросы

1. В чем цель планирования логической структуры каталога?
2. В чем цель планирования физической структуры каталога?
3. Назовите признаки, по которым следует осуществлять выбор многодоменной модели?
4. Какой подход предпочтительнее при проектировании структуры организационных подразделений: организационный или административный?
5. Каким образом деление на сайты влияет на процесс репликации?
6. Как выбирается число и расположение контроллеров домена?
7. Опишите основные модели планирования пространства имен.
8. Что такое учетная запись пользователя? Приведите примеры атрибутов учетной записи пользователя.
9. Чем отличаются организационные подразделения и группы безопасности?
10. Назовите основные элементы объектов групповых политик.

ТЕМА 9. БЕЗОПАСНОСТЬ ACTIVEDIRECTORY. ПРОТОКОЛЫ KERBEROS И IPSECURITY

План

1. Протокол аутентификации Kerberos. Основные термины и понятия.

2. Основные этапы аутентификации.

3. Протокол IPsec.

Данная тема рассчитана на три лекции (лекции 18–20).

9.1. Протокол аутентификации Kerberos. Основные термины и понятия

Протокол аутентификации Kerberos разработан в начале 80-х годов в Массачусетском технологическом институте (Massachusetts Institute of Technology, MIT). Описан в RFC 1510.

В WindowstServer 2003 используется модифицированная пятая версия протокола – Kerberostv5. Для шифрования применяется алгоритм DES (Data Encryption Standard – стандарт шифрования данных). Преимуществом протокола Kerberos по сравнению с протоколом NTLM является то, что в процессе аутентификации сервер не только удостоверяет подлинность клиента, но и по требованию клиента подтверждает свою достоверность. Еще одно преимущество состоит в том, что время аутентификации при использовании Kerberos меньше, чем в случае применения NTLM.

Рассмотрим основные термины, используемые при описании протокола Kerberos.

Понятия *аутентификации* и *авторизации* рассматривались ранее в разделе «Доверительные отношения».

Шифрование (encryption) – процесс преобразования данных в такую форму, которая не может быть прочитана без процесса расшифрования. Шифрование осуществляется с применением *шифрующего ключа (encryption key),* расшифрование использует *расшифровывающий ключ (decryption key).*

В симметричных методах шифрования, к которым относится алгоритм DES, шифрующий и расшифровывающий ключи совпадают,

и такой единый ключ называется *секретным ключом (secret key)*. Секретный ключ пользователя получается путем хеширования его пароля.

Хеширование (hashing) обозначает такое преобразование исходной последовательности данных, результат которого – *хеш* (hash), в отличие от результата шифрования, не может быть преобразован обратно в исходную последовательность. Это преобразование может осуществляться с помощью некоторого ключа. Хеширование часто применяют для проверки знания участниками соединения общего секретного ключа. При этом источник вычисляет хеш некоторого блока данных с использованием секретного ключа и отправляет эти данные совместно с хешем. Приемник также вычисляет хеш блока данных, и при условии совпадения ключей значения хешей должны быть равны.

Сеанс (session) – это период непрерывного соединения между двумя узлами (например, клиентом и сервером). В начале сеанса требуется пройти процедуру аутентификации. Соединение в течение сеанса осуществляется с использованием сеансового ключа.

Сеансовый ключ (session key) – секретный ключ, служащий для шифрования всех сообщений между участниками сеанса. Очевидно, он должен быть известен всем участникам сеанса.

В протоколе Kerberos существует три основных участника сеансов: клиент, сервер и посредник.

Клиент – компьютер (пользователь, программа), желающий получить доступ к ресурсам сервера. Предварительно клиент должен пройти процедуры аутентификации и авторизации, используя свое удостоверение.

Сервер – компьютер (программа), предоставляющий ресурсы авторизованным клиентам.

Посредник – это специальный физически защищенный сервер, на котором работают две службы: *центр распространения ключей* (Key Distribution Center, KDC) и *служба предоставления билетов* (Ticket Granting Service, TGS). В сетях Active Directory этим сервером является контроллер домена.

Центр распространения ключей KDC хранит секретные ключи всех клиентов и серверов и по запросу аутентифицированного клиента выдает ему удостоверение.

Служба предоставления билетов TGS выдает сеансовые билеты, позволяющие пользователям проверять подлинность серверов.

Удостоверения (credentials) – специальные сетевые пакеты, используемые для взаимной идентификации клиента и сервера. Удосто-

верения бывают двух видов: *билеты (tickets)* и *аутентификаторы (authenticators)*.

Билет (ticket) – специальный пакет, удостоверяющий подлинность своего владельца. В состав билета входят имя владельца, сеансовый ключ и другие параметры. Период действия билета ограничен параметром, который называется *время жизни (lifetime)*. По умолчанию время жизни равно пяти минутам.

Существует два типа билетов: *билеты TGT (Ticket-Granting Ticket – билеты на выдачу билетов)* и *сеансовые билеты (session ticket)*.

Билет TGT содержит учетные данные, выдаваемые пользователю центром распределения ключей KDC при входе пользователя в систему.

Сеансовый билет требуется для установления сеанса соединения клиента с сервером.

Аутентификатор (authenticator) – это пакет, доказывающий, что клиент действительно является обладателем секретного ключа.

Приведенные выше термины сведены в схему на рис. 9.1.

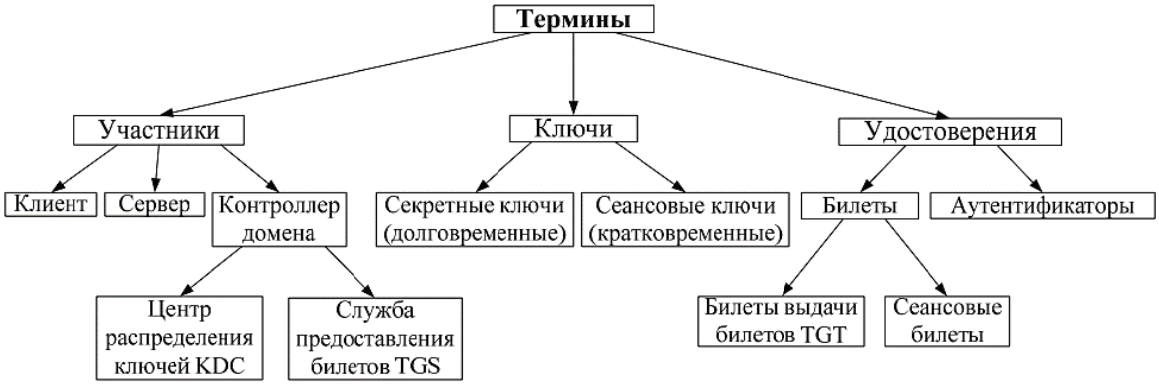


Рис. 9.1. Термины, используемые для описания протокола Kerberos

Для дальнейшего изложения введем обозначения, используемые в протоколе Kerberos.

Обозначение	Комментарий
A_C	Аутентификатор клиента
A_S	Аутентификатор сервера
K_C	Секретный ключ клиента
K_S	Секретный ключ сервера
$\{X\}K$	Сообщение X, зашифрованное ключом K
$\{A_C\}K_C$	Аутентификатор клиента, зашифрованный секретным ключом
$K_{A,B}$	Сеансовый ключ для соединения узлов A и B

$K_{C,TGS}$	Сеансовый ключ для соединения клиента и службы TGS
TGT	Билет TGT
$T_{C,S}$	Сеансовый билет для соединения клиента и сервера
N	Имя клиента
S	Имя сервера
t	Момент времени отправки сообщения

9.2. Основные этапы аутентификации

Клиенту для получения доступа к ресурсам сервера предварительно требуется пройти проверку подлинности, т. е. аутентифицироваться. Процедура аутентификации состоит из трех основных этапов (рис. 9.2):

1) регистрация клиента;
2) получение сеансового билета;
3) доступ к серверу.

Рис. 9.2. Основные этапы аутентификации

9.2.1. Этап регистрации клиента

При входе в систему под управлением Windows Server пользователь вводит имя своей учетной записи, пароль и указывает домен (рис. 9.3).

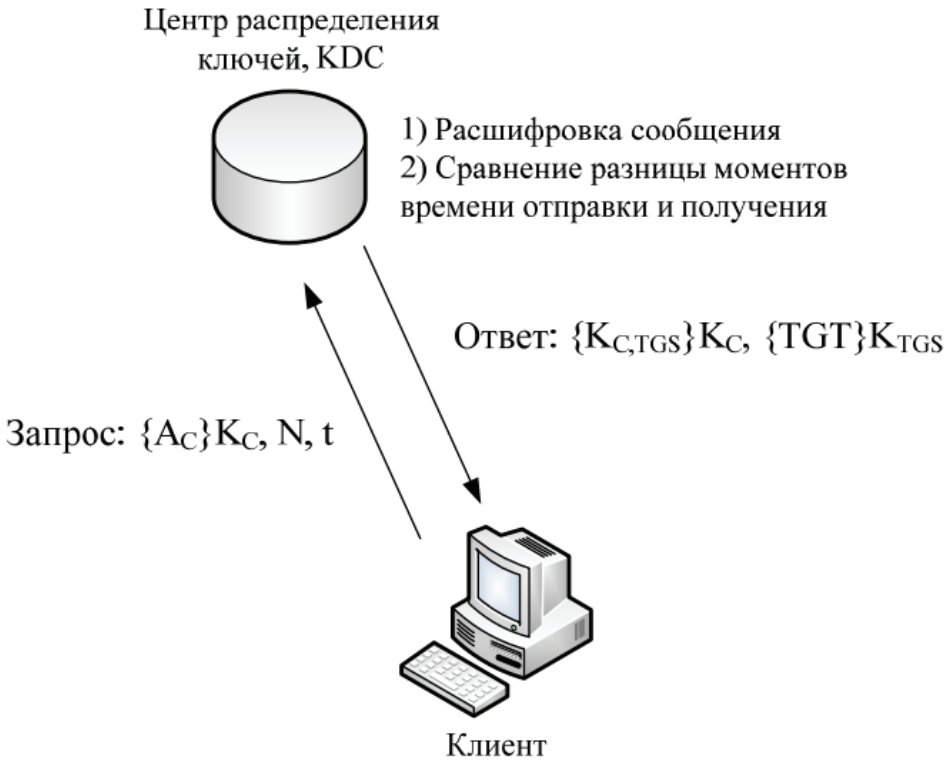


Рис. 9.3. Этап регистрации клиента

Пароль при помощи хеширования преобразуется в секретный ключ клиента K_C. Точно такой же ключ хранится в центре распределения ключей KDC и сопоставлен с данным пользователем. Клиент создает аутентификатор $\{A_C\}K_C$, зашифрованный с использованием ключа K_C, и отсылает его центру распределения ключей (рис. 9.3). Аутентификатор содержит информацию об имени клиента N и время отправки аутентификатора t.

Используя свою копию ключа K_C, центр распределения ключей пытается расшифровать полученное сообщение. В случае успеха вычисляется разница между временем создания аутентификатора и временем его получения. Если разница не превышает пяти минут, то клиент считается аутентифицированным и ему высылается следующая информация:

- $\{K_{CT}G_S\}K_C$ – сеансовый ключ KQ_{TGS} для связи клиента и службы TGS, зашифрованный ключом K_C;

- $\{TGT\}K_{TGS}$– билет на выдачу билетов TGT, зашифрованный ключом K_{TGS}, известным только службе TGS.

Сеансовый ключ K_{CTGS} клиент в состоянии расшифровать, используя свой ключ КС, а расшифровка билета TGT клиентом невозможна, так как ключ K_{TGS} ему неизвестен. Билет TGT в зашифрованном виде сохраняется в кэш-память клиента и при необходимости извлекается оттуда.

В дальнейшем клиент будет использовать полученную информацию для запроса полномочий на доступ к конкретному серверу .

В том случае, если аутентификатор не удалось расшифровать или разница по времени превышает пять минут, клиент считается не прошедшим аутентификацию у службы TGS.

Проверка разницы моментов времени осуществляется в целях защиты от перехвата аутентификатора и его несанкционированного использования.

Так как аутентификаторы, генерируемые клиентом, не повторяются (для их создания применяется значение текущего момента времени), то перехваченный идентификатор может быть использован только в течение пяти минут. Однако центр распределения ключей ведет учет всех аутентификаторов, полученных за последние пять минут, и в случае совпадения аутентификатор отклоняется. Отметим, что для правильного функционирования протокола Kerberos часы всех участников соединения должны быть синхронизированы с точностью до минуты.

9.2.2. Этап получения сеансового билета

Когда клиенту требуется получить доступ к ресурсам некоторого сервера, он обращается к службе предоставления билетов TGS с запросом о выдаче сеансового билета для соединения с данным сервером. В запрос включается следующая информация (рис 9.4):

- $\{A_C\}K_{CTGS}$ – аутентификатор клиента A_C, зашифрованный с помощью ключа K_{CTGS};
- $\{TGT\}K_{TGS}$ – билет на выдачу билетов TGT, зашифрованный ключом K_{TGS};
- S – информация о сервере, с которым требуется установить соединение;
- t – время отправки запроса.

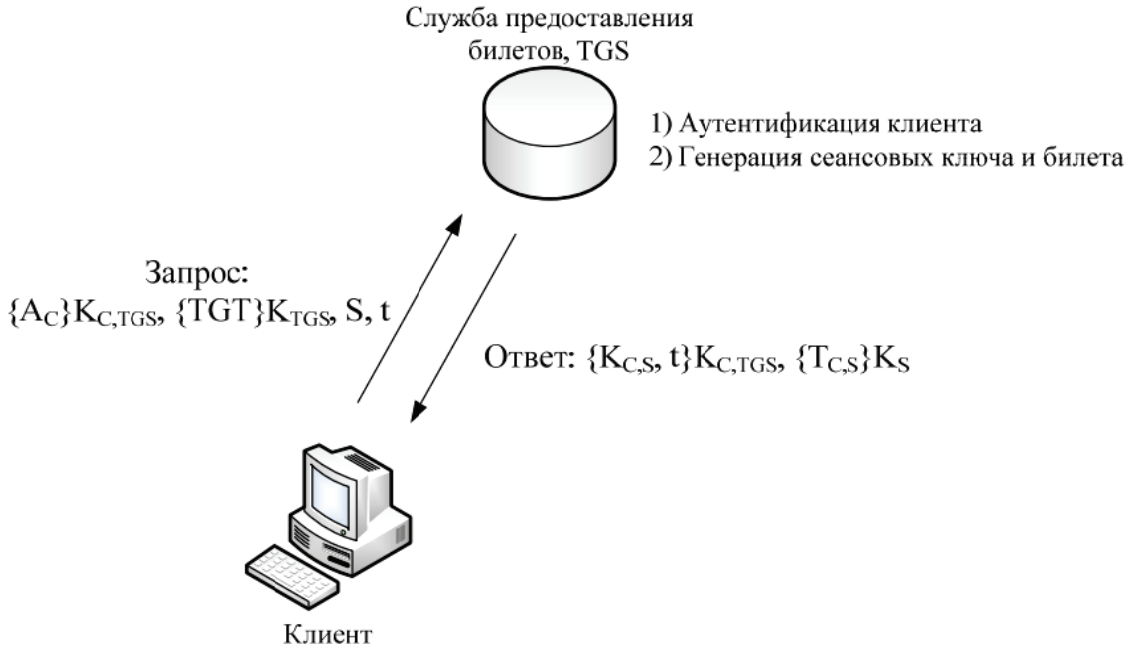


Рис. 9.4. Этап получения сеансового билета

Аутентификатор клиента позволяет службе TGS удостовериться, что клиент является тем, за кого себя выдает. Использование билетов TGT экономит время: служба предоставления ключей TGS не обращается к базе данных центра распределения ключей KDC.

В случае успешной аутентификации на запрос клиента служба TGS отвечает следующей информацией:

- {KCS, t} KC,TGS – сеансовый ключ KCS для связи клиента с сервером, а также время создания ключа; оба параметра зашифрованы ключом $K_{C,TGS}$.
- {TC,S}KS – сеансовый билет TC,S, зашифрованный при помощи ключа KS, известного только службе TGS и серверу. Сеансовый билет предназначен только для сервера, клиент не в состоянии его прочитать.

Сеансовый ключ KC,s генерируется случайным образом, поэтому при каждом новом запросе (даже для связи с одним и тем же сервером) клиент будет получать новые сеансовые ключи. Клиент может расшифровать сеансовый ключ, так как он зашифрован ключом K_{CTGS}, известным клиенту.

Сеансовый билет T_{CS} содержит следующие данные:

- имя сервера;
- имя клиента;
- сеансовый ключ;

- время начала действия билета;
- время окончания действия билета;
- список возможных сетевых адресов клиента.

Последний элемент является необязательным и применяется для дополнительной защиты – в этом случае клиенты не могут соединяться с сервером с адресов, не перечисленных в списке.

Сеансовые билеты, полученные клиентом для разных серверов, сохраняются в кэш-памяти. Таким образом, если клиенту требуется получить доступ к какому-либо серверу, сначала осуществляется поиск в кэш-памяти сеансовых билетов для этого сервера. При отсутствии таковых клиент извлекает билет TGT из кэш-памяти и обращается с запросом к службе TGS.

9.2.3. Этап доступа к серверу

Получив сеансовый билет $T_{C,s}$ и сеансовый ключ $K_{C,s}$, клиент может проходить процедуру аутентификации на требуемом сервере и в случае успешного прохождения начинать обмен данными. Запрос на аутентификацию включает следующие параметры (рис. 9.5):

- $\{A_C\}K_{CS}$– аутентификатор A_C, зашифрованный ключом K_{CS}. Содержит информацию об имени клиента, времени отправления, а также ключ KC,s;
- $\{T_{C,s}\}K_s$– сеансовый билет, зашифрованный ключом K_s.

Подлинность клиента удостоверятся следующим образом. В аутентификатор АСклиент записывает ключ KC,s. Сервер, расшифровав сеансовый билет TC,s с помощью своего секретного ключа Ks, извлекает из него ключ KC,s и сравнивает с ключом, полученным из аутентификатора. Если ключи совпадают, клиент является подлинным, так как он не мог изменить содержимое сеансового билета T_{CS}.

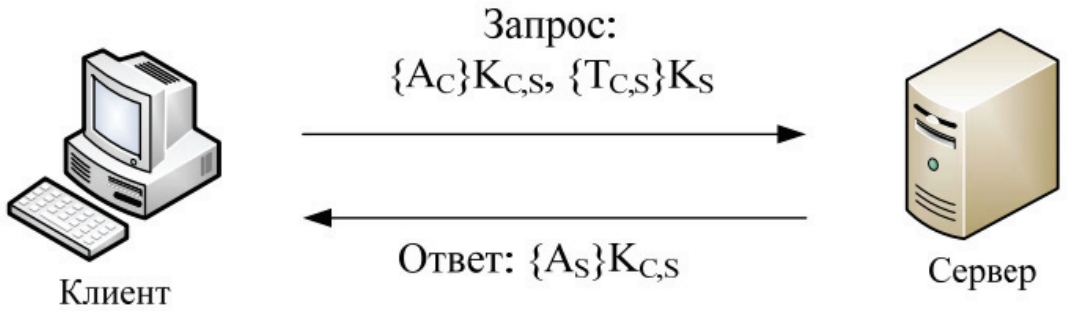


Рис 9.5. Этап доступа к клиенту

Если клиенту требуется подтверждение подлинности сервера, тот отправляет ответ, который содержит аутентификатор сервера A_s, включающий параметр времени отправления из аутентификатора клиента A_C. Без знания секретного ключа K_s извлечь данный параметр из запроса клиента невозможно. Следовательно, если время отправления запроса сервер передал верно, он считается аутентифицированным.

9.3. Протокол IPsec

Протокол Kerberos применяется для аутентификации участников соединения. Но и после этапа аутентификации данные, передаваемые по сети, следует защищать. Стандартные протоколы стека TCP/IP, такие как IP, TCP, UDP, не обладают встроенными средствами защиты. На эту проблему в 1994 году обратил внимание Совет по архитектуре Интернета (Internet Architecture Board, IAB), издав RFC 1636 (*Report of IAB Workshop on Security in the Internet Architectures* («Отчет семинара IAB по безопасности в архитектуре Интернета»)). Инициированная этим сообщением работа привела к появлению протокола *IPsec* (IP security – безопасность IP), описанного в нескольких стандартах RFC (в частности, в RFC 2401-2412). Новая технология безопасности является необходимой частью протокола IPv6, а также может применяться в сетях IPv4.

Протокол IPsec действует на сетевом уровне модели OSI и может применяться независимо от протоколов верхнего уровня, т. е. прикладной протокол может использовать IPsec, считая, что работает с обычным протоколом IP. При этом данные протоколов верхних уровней упаковываются в пакеты IPsec, которые, в свою очередь, помещаются в пакеты протокола IP.

9.3.1. Функции протокола IPsec

Протокол IPsec обеспечивает наличие следующих функций:

- аутентификация – приемник пакетов в состоянии проверить подлинность их источника;
- целостность – осуществляется контроль того, что данные дойдут до получателя в неизменном виде;
- конфиденциальность – шифрование данных обеспечивает их недоступность для несанкционированного просмотра;

- распределение секретных ключей – для правильной работы протокола IPsec необходимо автоматически обеспечивать источник и приемник пакетов секретными ключами для шифрования и расшифрования данных.

Для реализации представленных функций используются три основных протокола:

- AH (Authentication Header – заголовок аутентификации) обеспечивает целостность и аутентичность;
- ESP (Encapsulating Security Payload – инкапсуляция зашифрованных данных) предоставляет функции целостности, аутентичности и конфиденциальности;
- IKE (Internet Key Exchange – обмен ключами Интернета) генерирует и распределяет секретные ключи.

Можно заметить, что протокол ESP имеет схожие функции с протоколом AH. Пересечение функций вызвано тем, что на применение протоколов шифрования во многих странах накладываются определенные ограничения. В связи с этим оба протокола могут применяться независимо, хотя наивысший уровень защиты достигается при их совместном использовании.

На рис. 9.6 представлена структура протокола IPsec и взаимосвязь основных протоколов, входящих в его состав.

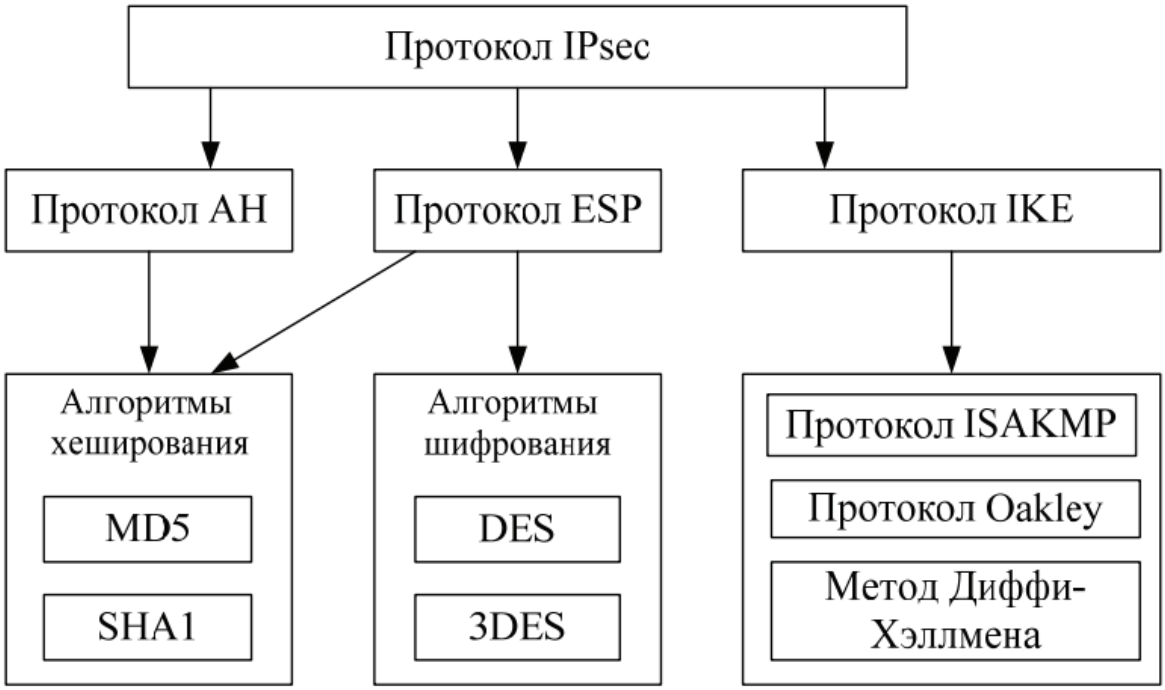


Рис. 9.6. Структура протокола IPsec

9.3.2. Протоколы AH и ESP

Протокол AH (описан в RFC 2402) снабжает пакет IPsec своим незашифрованным заголовком, который обеспечивает:

- аутентификацию исходных данных;
- целостность данных;
- защиту от дублирования уже полученных данных.

Первые две функции протокола AH реализуются путем применения алгоритмов хеширования MD5 (Message Digest 5 – алгоритм формирования профиля сообщения), разработанного Рональдом Ривестом (описан в RFC 2403) или SHA1 (Secure Hash Algorithm 1– алгоритм безопасного хеша), который разработан Национальным институтом стандартов и технологий NIST (National Institute of Standards and Technology). Является более стойким по сравнению с MD5. Описан в RFC 2404. Процедура хеширования осуществляется источником с помощью секретного ключа, который был выдан источнику и приемнику пакета с использованием протокола IKE. Полученное значение хеша помещается в специальное поле заголовка AH. Приемник также осуществляет процедуру хеширования, применяя тот же секретный ключ. В том случае, если вычисленный хеш совпадает с хешем, извлеченным из пакета, данные считаются аутентифицированными и целостными. Иначе пакет в процессе передачи подвергся каким-либо изменениям и не является правильным.

Функция защиты от дублирования уже полученных пакетов осуществляется с помощью поля номера пакета в заголовке AH. В это поле приемник заносит значение счетчика, увеличивающееся при отправке каждого пакета на единицу. Приемник отслеживает номера получаемых пакетов, и, если такой номер совпадает с недавно полученным, пакет отбрасывается.

Протокол ESP (описан в RFC 2406) решает задачи, подобные протоколу AH: обеспечение аутентификации и целостности исходных данных, а также защиту от дублирования пакетов. Кроме того, протокол ESP предоставляет средства обеспечения конфиденциальности данных при помощи алгоритмов шифрования.

Задачи аутентификации, целостности и защиты от дублирования решаются теми же методами, что и в протоколе AH. Передаваемый пакет, за исключением нескольких служебных полей, шифруется с применением алгоритмов шифрования DES и 3DES (DES с тремя ключами).

9.3.3. Протокол IKE

Управление секретными ключами в протоколе IPsec осуществляется при помощи протокола IKE (описан в RFC 2409). Данный протокол основан на двух протоколах: ISAKMP (Internet Security Association and Key Management Protocol – протокол межсетевой ассоциации защиты и управления ключами) и протоколе определения ключей Оакли (Oakley Key Determination Protocol).

Протокол IKE устанавливает соединение между двумя узлами сети, называемое *безопасной ассоциацией* (Security Association, SA). Безопасная ассоциация обеспечивает передачу защищенных данных только в одну сторону, поэтому для установки двустороннего соединения требуется определить две безопасные ассоциации. Для аутентификации узлов безопасной ассоциации, согласования между ними методов хеширования и шифрования IKE использует протокол ISAKMP (описан в RFC 2408).

Для генерации и обмена секретными ключами IKE использует протокол определения ключей Оакли (описан в RFC 2412), разработанный на основе метода обмена ключами Диффи-Хэллмена (Diffie-Hellman). В этом методе секретный ключ генерируется на двух узлах путем обмена двумя числами через открытую сеть. При этом перехват чисел не даст информации о ключах.

Контрольные вопросы

1. Безопасность каких основных процессов следует обеспечивать в сетях передачи данных?
2. Что такое сеанс?
3. Что такое хеширование?
4. Каковы функции центра распределения ключей?
5. В чем отличие билетов TGT от сеансовых билетов?
6. Опишите этап регистрации клиента.
7. Расскажите об этапе получения сеансового билета.
8. Опишите этап доступа к серверу.
9. Назовите основные функции протокола IPsec.
10. Для чего используются протоколы AH и ESP?
11. Для чего используются протоколы IKE?

ТЕМА 10. МАРШРУТИЗАЦИЯ В КОМПЬЮТЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ. СЛУЖБА RRAS

План

1. Понятие маршрутизации. Служба RRAS.

2. Алгоритмы маршрутизации.

3. Адресация в компьютерных системах с маршрутизацией.

4. Методы обмена информацией.

5. Протоколы маршрутизации.

Данная тема рассчитана на две лекции (лекции 21–22).

10.1. Понятие маршрутизации. Служба RRAS

Служба маршрутизации и удаленного доступа является интересной, но сложной технологией для многих администраторов. RRAS (Routing and Remote Access Service) позволяет удаленным клиентам «проходить» физические границы сетевого окружения, чтобы подсоединяться к сети и использовать ее ресурсы.

RRAS содержит много возможностей, включая поддержку разделяемого использования Интернет-соединения, коммутируемое соединение с сервером, маршрутизацию информации из одной сети в другую, защиту данных путем использования виртуальной частной сети (VPN) и многое другое.

Сетевая среда часто бывает сегментирована по различным причинам, включая следующие факторы:

- количество доступных IP-адресов в сетевой среде TCP/IP;
- разделение функций администрирования и управления;
- соображения безопасности;
- владение сетью.

Многие маршрутизаторы могут маршрутизировать TCP/IP, IPX и AppleTalk. Но поскольку работа Windows Server и Интернета основывается на TCP/IP, основное внимание будет уделяться маршрутизации TCP/IP. При использовании TCP/IP адрес сети определяется IP-адресом в сочетании с маской подсети. Адрес сети идентифицирует сеть, где находится данное устройство.

Между разъединенными сетями может требоваться обмен информацией, и тогда на помощь приходит маршрутизация. *Маршрутизация* – это процесс передачи информации через межсетевую границу. Точка отправки называется *источником (отправителем),* а точка приема называется *пунктом назначения (получателем)*. Промежуточное устройство (обычно маршрутизатор, иногда это несколько устройств) отвечает за передачу информации из одной сети в другую, пока эта информация не дойдет до указанного получателя. Например, когда компьютер одной сети отправляет информацию компьютеру, который находится в другой сети, он направляет эту информацию маршрутизатору. Маршрутизатор рассматривает этот пакет и использует адрес получателя в заголовке пакета для передачи информации в соответствующую сеть.

Выполнять функции маршрутизатора также может компьютер под управлением серверной ОС с настроенной службой маршрутизации. Он выполняет данные функции, определяя оптимальный маршрут с помощью алгоритмов маршрутизации.

В ОС типа Windows Server для корректного функционирования маршрутизации создаются таблицы маршрутизации. Таблицу маршрутизации можно просмотреть путем выполнения команда route print (рис. 10.1).

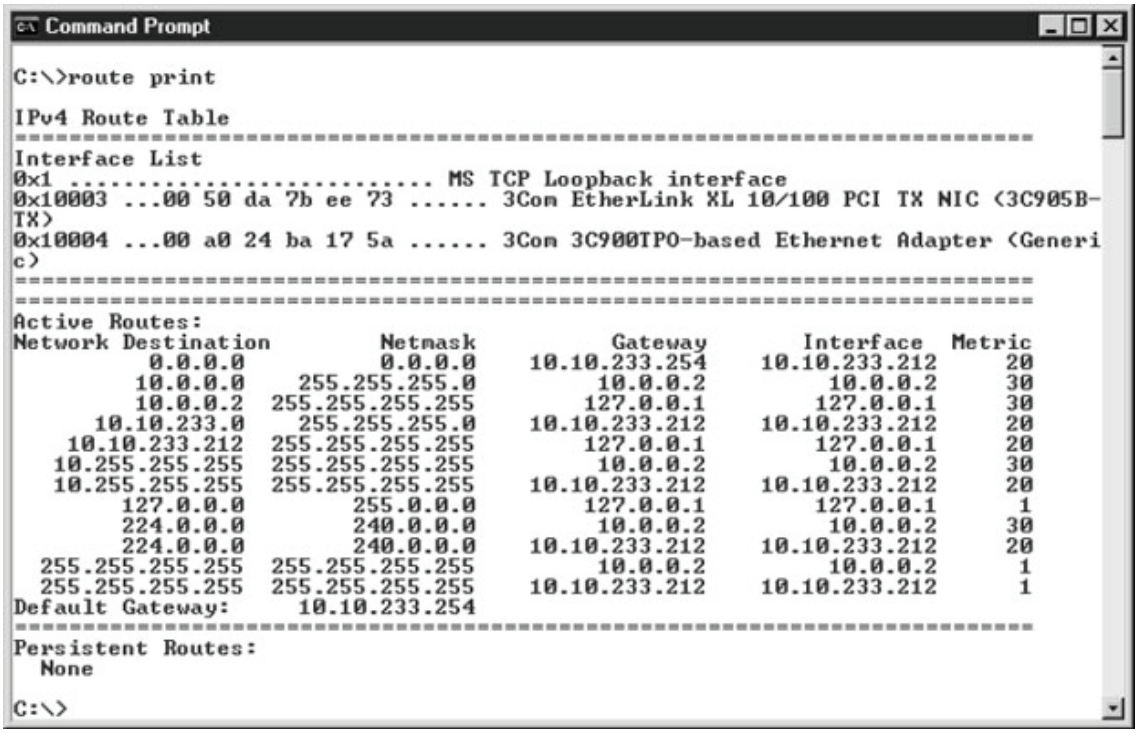


Рис. 10.1. Таблица маршрутизации

Маршрутизаторы могут также использовать *сообщения об изменениях маршрутной информации* для взаимодействия с другими маршрутизаторами, что позволяет им сравнивать и обновлять свои таблицы маршрутизации, вводя в них информацию о маршрутах в другие сети.

10.2. Алгоритмы маршрутизации

Маршрутизаторы, а также компьютеры Windows Server, сконфигурированные как маршрутизаторы, обычно используют алгоритмы статической или динамической маршрутизации, но поддерживается также маршрутизация с коммутируемым соединением по требованию. Все эти алгоритмы маршрутизации в основном отвечают одной цели, хотя они имеют различные механизмы.

Статическая маршрутизация

Задает единственный путь, который должен использоваться для передачи информации между двумя точками. Администратор должен задавать и конфигурировать статические маршруты в таблицах маршрутизации, и они не изменяются, пока это не сделает администратор. Сетевые среды со статической маршрутизацией организуются достаточно просто и особенно подходят для небольших окружений, где возможны лишь небольшие изменения в топологии маршрутизации.

Основным недостатком сетевых сред со статической маршрутизацией является то, что они не адаптируются к изменению состояния сети.

Например, в случае отключения маршрутизатора или канала, статический маршрут не позволяет перенаправлять пакеты на другие маршрутизаторы, чтобы передать их нужным получателям. Кроме того, при добавлении или удалении какой-либо сети в вашем окружении, администратор должен задавать возможные сценарии маршрутизации и конфигурировать их соответствующим образом. Поэтому сетевые среды со статической маршрутизацией (в особенности те, что подвержены частым изменениям) не подходят для более крупных сетей. Достаточно оценить расходы на администрирование, чтобы понять, что статическая маршрутизация подходит только для небольших сетевых окружений.

В качестве эмпирического правила используйте статические маршруты только при следующих условиях:

- сетевые окружения с небольшим числом сетей;
- соединения, которые не предполагается изменять в ближайшем будущем (например, маршрутизатор, который используется как последнее средство, когда информацию нельзя маршрутизировать иным способом).

Авто-статическая маршрутизация

Маршрутизаторы на базе Windows Server, которые используют статические маршруты, могут иметь собственные таблицы маршрутизации, обновляемые вручную или автоматически. Автоматически обновляемые статические маршруты называют *авто-статической* маршрутизацией. Соответствующие обновления можно конфигурировать с помощью интерфейса RRAS или с помощью утилиты NETSH. Это позволяет обновлять информацию маршрутизаторов Windows Server только в определенные периоды времени, что дает экономию затрат на соединения и использование меньшей доли пропускной способности каналов.

Если указывается, что нужно обновить статические маршруты, маршрутизатор отправляет запрос через активное соединение, чтобы обновить все маршрутизаторы данного соединения. Получивший запрос маршрутизатор удаляет все существующие авто-статические маршруты и затем вводит новые записи авто-статической маршрутизации в виде статических (постоянных) маршрутов.

Авто-статические обновления поддерживаются только в тех случаях, когда вы используете протокол RIP для IP.

Динамическая маршрутизация

Как можно понять из названия, алгоритмы динамической маршрутизации адаптируются к изменениям сетевой среды без ручного вмешательства. Вносимые изменения почти мгновенно отражаются в информации маршрутизатора. Условия, при которых целесообразно использовать динамическую маршрутизацию:

- происходит отключение маршрутизатора или канала, что требует изменения маршрута для передаваемой информации;
- в интерсети добавляется или удаляется маршрутизатор;
- большое сетевое окружение, где имеется много сценариев маршрутизации;
- большое сетевое окружение, в котором часто происходят изменения сетевой топологии.

Алгоритмы динамической маршрутизации могут адаптироваться к изменению состояний в реальном масштабе времени путем взаимодействия с другими маршрутизаторами. Когда маршрутизатор полу-

чает уведомление, что в сети произошло какое-либо изменение, он пересчитывает маршруты и уведомляет другие маршрутизаторы. Это позволяет всем маршрутизаторам сети получать информацию о топологии всей сети даже в те моменты, когда она изменяется. В настоящее время большинство маршрутизаторов используют алгоритмы динамической маршрутизации и хорошо адаптируются к сети любого размера.

Маршрутизация с коммутируемым соединением по требованию (demand/dial routing)

Большинство протоколов маршрутизации (RIP, OSPF и т. д.), которые используются для взаимодействия с другими маршрутизаторами, периодически отправляют маршрутную информацию, чтобы адаптироваться к динамическим изменениям состояния сети. Это требуется для того, чтобы информация передавалась по маршрутам с наименьшей «стоимостью». Однако существуют ситуации, когда периодические обновления маршрутизаторов весьма нежелательны. В таких ситуациях можно использовать маршрутизацию с коммутируемым соединением по требованию. При такой маршрутизации активизация канала происходит только при необходимости передачи информации другой стороне соединения, что дает экономию затрат на соединения. Для поддержки обновлений маршрутизаторов используется статическая или авто-статическая маршрутизация.

10.3. Адресация в компьютерных системах с маршрутизацией

При работе со службой RRAS, поддерживающей TCP/IP, вы должны предоставлять клиентам IP-адреса. Есть два варианта получения необходимой информации TCP/IP: клиенты могут получать IP-адрес непосредственно из статического пула адресов сервера RRAS или могут использовать DHCP.

RRAS можно конфигурировать для назначения IP-адреса из пула, содержащего один или несколько статических IP-адресов. Для каждого следующего клиента, присоединяющегося с помощью RRAS, назначается следующий доступный IP-адрес из этого пула, и этот адрес становится свободным только когда происходит отсоединение клиента.

Рекомендуется использовать саму службу RRAS для назначения IP-адресов клиентам только в небольших по размеру сетях. На это

имеются две причины. Во-первых, клиенту предоставляется ограниченное количество информации. Вы должны вручную сконфигурировать иные параметры TCP/IP на стороне клиента, чтобы он мог эффективно взаимодействовать с другими машинами и использовать ресурсы вашей сети. Во-вторых, при изменении информации TCP/IP (например, добавление или изменение сервера WINS или сервера DNS), информация не передается клиенту.

Данные проблемы можно решить, используя скрипты, которые изменяют настройку конфигурации TCP/IP при каждом входе клиента. Но эффективней использовать DHCP.

Существует пять механизмов разрешения имен, которые могут использоваться клиентами для поиска и использования ресурсов:

- сервер DNS;
- сервер WINS;
- файл HOSTS;
- файл LMHOSTS;
- широковещательные сообщения.

10.4. Методы обмена информацией

RRAS – это одна служба, она действует как клиент/серверная служба. Клиентом является любой объект, который для доступа к ресурсам подсоединяется с помощью службы сервера RRAS, в то время как серверная сторона обеспечивает эти соединения.

Служба маршрутизации и удаленного доступа может использовать различные методы обмена информацией:

- аналоговые соединения;
- ISDN (Integrated Services Digital Network);
- X.25;
- ADSL (Asymmetric Digital Subscriber Line);
- ATM (Asynchronous Transfer Mode);
- ICS (Internet Connection Sharing).

Аналоговое соединение – это стандартная телефонная служба, предназначенная, прежде всего для голосовой связи, но в настоящее время она также широко используются для передачи данных сети и цифровой информации. Очень медленное соединение (56 Кбит/с) используется только при отсутствии других вариантов подключения.

ISDN (цифровая сеть связи с комплексными услугами) – это цифровая версия телефонных сетей. Обычно в линии ISDN данные пере-

даются со скоростью от 64 до 128 Кбит/с в зависимости от количества используемых каналов ISDN использует для передачи данных два B-канала, имеет также третий канал (D-канал), который используется не для передачи данных, а для слежения и управления двумя B-каналами.

X.25 – это глобальный стандарт, разработанный для передачи данных через телефонную сеть или через сети с коммутацией пакетов. Соединения X.25 очень похожи на телефонные или коммутируемые соединения. Хост-компьютер вызывает другой компьютер с запросом соединения, и если он принят, можно передавать информацию. В Windows Server 2003 и 2008 соединения X.25 поддерживаются путем использования сборщиков/разборщиков пакетов (PAD) и карт X.25. Клиенты RRAS могут также использовать модем для коммутируемого соединения с сервером Windows Server 2003 с помощью X.25, но для приема этого вызова на сервере должна использоваться карта X.25.

ADSL – это новый метод доступа, обеспечивающий высокоскоростной обмен данными с помощью тех же телефонных линий, что используются в обычных телефонных сетях. Хотя ADSL предполагает высокую пропускную способность, «восходящий» поток (от клиента к службе) передается медленнее, чем «нисходящий» (от службы к клиенту). Обычно «восходящий» поток передается со скоростью от 64 до 256 Кбит/с, а «нисходящий» – до 1.544 Мбит/с для передачи данных. Существуют также разновидности ADSL, такие как HDSL и VDSL, предполагающие значительно более высокие скорости передачи восходящего и нисходящего потоков. При подключении сетевого адаптера ADSL в Windows Server 2003, он будет представлен либо как интерфейс Ethernet либо как интерфейс коммутируемого соединения. Конфигурация соединения зависит от интерфейса, который оно использует. Например, если адаптер ADSL представлен как интерфейс Ethernet, то данное соединение действует в точности так, как если бы это было соединение Ethernet, но если он представлен в виде интерфейса коммутируемого соединения, то ADSL использует ATM.

ATM (Асинхронный режим передачи) – это набор технологий, которые обеспечивают надежные высококачественные сетевые услуги для аудио, видео и данных. В ATM используется технология VLSI (интеграция сверхвысокого уровня) для сегментирования данных, передаваемых с высокой скоростью, в пакеты, которые называются *ячейками*. Каждая ячейка имеет размер 53 байта и состоит из 5 байтов заголовочной информации и 48 байтов данных.Технология ATM *ориентирована на соединения*, а это означа-

ет, что путь от источника к месту назначения создается до того, как начинается передача данных через соединение.

10.5. Протоколы маршрутизации

Протоколы маршрутизации являются протоколами особого типа – они следят за топологией всей сети в маршрутизируемой сетевой среде, динамически поддерживают информацию о других маршрутизаторах в данной сети и используют эту информацию, чтобы определить наилучший маршрут для передаваемых данных.

Протоколы маршрутизации используют алгоритмы, которые влияют на маршрутизатор, а также на маршрутизацию информации из одной сети в следующую сеть. Каждый протокол маршрутизации имеет характеристики, которые отличают его от других протоколов, но все они должны обеспечивать такие качества как простота, оптимальность, стабильность и гибкость.

В настоящее время используются многие протоколы маршрутизации, включая следующие:

- BGP (Border Gateway Protocol – пограничный межсетевой протокол);
- EIGRP (Enhanced Interior Gateway Routing Protocol – расширенный внутренний шлюзовый протокол);
- EGP (Exterior Gateway Protocol – внешний шлюзовый протокол);
- IGRP (Interior Gateway Routing Protocol – внутренний шлюзовый протокол);
- OSPF (Open Shortest Path First – открытие в первую очередь кратчайших маршрутов);
- RIP (Routing Information Protocol – протокол маршрутной информации);

В маршрутизаторы RRAS под управлением Windows Server 2003 включена встроенная поддержка для OSPF и RIP. Кроме того, RRAS может поддерживать протоколы маршрутизации от сторонних компаний (такие как BGP, EIGRP, EGP и IGRP) с помощью своих интерфейсов прикладного программирования (API).

Протокол RIP

RIP – это протокол дистанционно-векторной маршрутизации, который относительно прост для использования и конфигурирования. Протоколы дистанционно-векторной маршрутизации отправляют обновления информации маршрутизаторов только соседним маршрутизаторам.

Существуют две версии RIP. Для обновления информации маршрутизаторов в версии 1 используются широковещательные (broadcast) сообщения, а в версии 2 – групповые (multicast) сообщения. RIP динамически поддерживает маршрутную информацию путем отправки сообщений с обновлением маршрутной информации другим маршрутизаторам с помощью RIP в интерсети каждые 30 секунд.

В дополнение к сообщениям обновления информации маршрутизаторов, RIP поддерживает несколько других механизмов, повышающих надежность и поддерживающих таблицы маршрутизации на уровне текущих изменений, включая следующее.

1. Ограничение по количеству сегментов (транзитных участков). Путь между двумя соседними маршрутизаторами считается сегментом (транзитным участком). Это средство ограничивает путь к точке назначения 15 сегментами. Точки назначения, для достижения которых требуется более 15 сегментов, считаются недостижимыми.

2. Блокировки (Hold-down). Блокировки гарантируют, что старые, недоступные маршруты не будут использоваться в таблице маршрутизации.

3. Ограничение «горизонта» (Split horizons). Это средство препятствует избыточности сообщений с обновлением маршрутной информации (циклов маршрутизации), направляемых маршрутизатору, из которого было отправлено данное сообщение.

4. Подавление возвратных обновлений (Poison reverse updates). Это средство препятствует зацикливанию маршрутов в интерсети.

Протокол OSPF

Это эффективный, но довольно сложный протокол маршрутизации, поддерживающий как состояния связи, так и таблицы маршрутизации. Как протокол маршрутизации каналов, OSPF сохраняет информацию, получаемую от всех остальных маршрутизаторов в сети, и использует эту информацию для расчета кратчайшего пути до каждого маршрутизатора.

OSPF был разработан с целью выхода за ограничения других протоколов маршрутизации, таких как RIP. Он поддерживает масштабирование без существенного увеличения объема служебной информации. Протокол маршрутизации OSPF описан в документах RFC 1247 (для OSPF) и 2328 (для OSPF версии 2).

Ключевым отличием протокола OSPF является то, что он может действовать иерархически (рис. 10.2).

Каждый маршрутизатор OSPF отправляет и получает маршрутную информацию от всех остальных маршрутизаторов в автономной

системе (AS), где имеется набор маршрутизаторов (сетей), которые применяют общую стратегию маршрутизации. Каждая AS может быть разбита на меньшие группы, которые называются областями и являются, по сути, непрерывными сетями в соответствии с их сетевыми интерфейсами.

Некоторые маршрутизаторы могут иметь несколько интерфейсов и могут быть связаны более чем с одной областью. Эти маршрутизаторы соответственно называются маршрутизаторами границ областей (ABR).Они занимаются передачей маршрутной информации из одной области в другую, что позволяет снизить трафик маршрутизации в автономной области (AS).

Для AS, содержащей несколько областей, требуется магистраль для структурной и управляющей поддержки. Эта магистраль соединяет маршрутизаторы границ областей (ABR) и любые сети, которые не полностью содержатся в любой конкретной области.

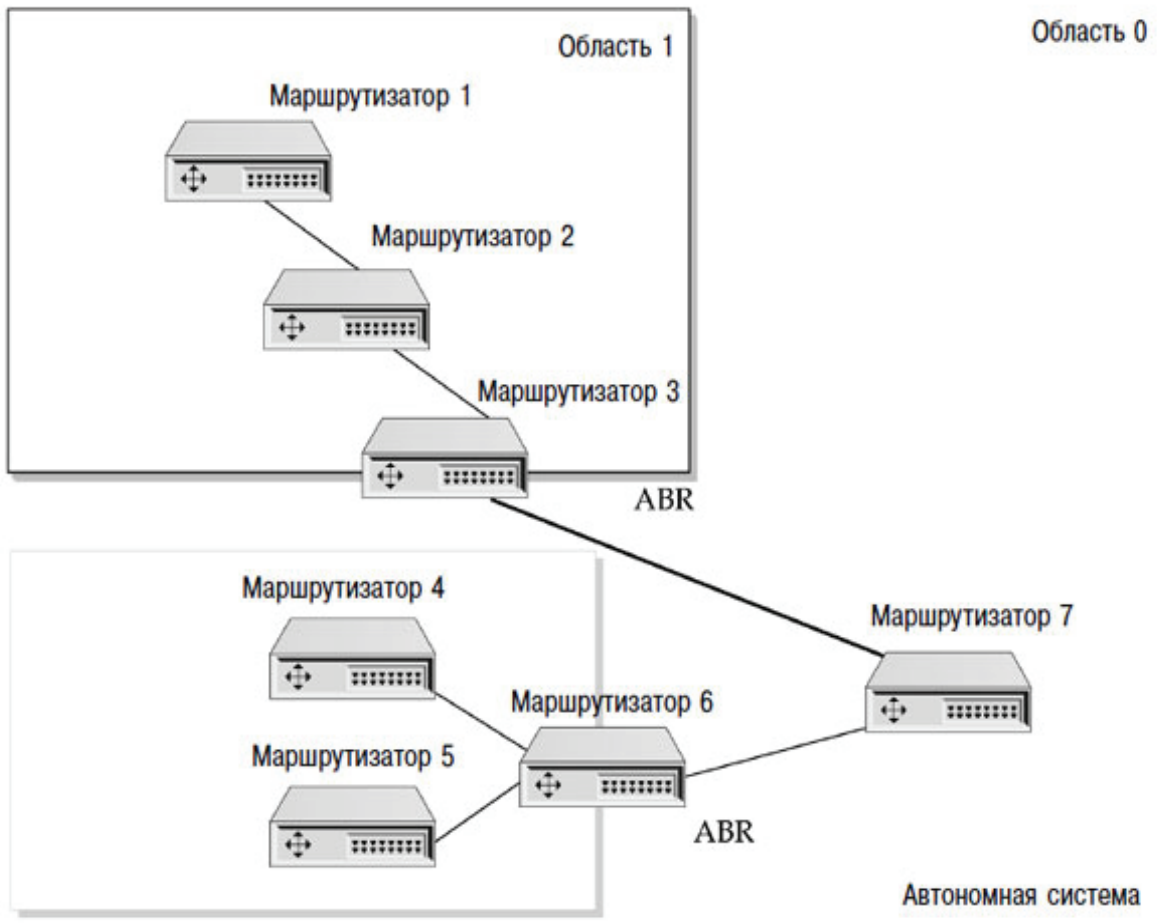


Рис. 10.2. Пример иерархической системы на базе протокола OSPF

Контрольные вопросы

1. Что такое удаленный доступ?
2. Назовите виды удаленного доступа.
3. Опишите алгоритм статической маршрутизации, ее достоинства и недостатки.
4. Опишите алгоритм авто-статической маршрутизации, ее достоинства и недостатки.
5. Опишите алгоритм динамической маршрутизации, ее достоинства и недостатки.
6. Опишите алгоритм маршрутизации с коммутируемым соединением по требованию, ее достоинства и недостатки.
7. Опишите основные используемые методы обмена информацией.
8. Какие вы знаете протоколы маршрутизации?
9. Приведите основные принципы функционирования протокола OSPF.
10. Приведите основные принципы функционирования протокола RIP.

ТЕМА 11. УДАЛЕННЫЙ ДОСТУП В ИНФОРМАЦИОННЫХ СИСТЕМАХ. VIRTUAL PRIVATE NETWORK

План

1. Протоколы удаленного доступа.

2. Протоколы аутентификации удаленных клиентов.

3. Общая характеристика виртуальных частных сетей.

4. Протоколы виртуальных частных сетей.

Данная тема рассчитана на одну лекцию (лекция 23).

Возможность использования удаленными пользователями ресурсов локальной сети называется **удаленным доступом** (remote access). Различают два основных вида удаленного доступа:

– *соединение по коммутируемой линии* (dial-up connection);

– *соединение с использованием виртуальных частных сетей* (Virtual Private Networks, VPN).

Оба вида соединений работают по модели «клиент-сервер».

Клиент удаленного доступа – это компьютер, который имеет возможность подключаться к удаленному компьютеру и работать с его ресурсами или с ресурсами удаленной сети так же, как с ресурсами своей локальной сети. Единственное отличие удаленной работы от локальной с точки зрения клиента – более низкая скорость соединения.

Сервер удаленного доступа(Remote Access Server, RAS) – это компьютер, способный принимать входящие запросы от клиентов удаленного доступа и предоставлять им собственные ресурсы или ресурсы своей локальной сети.

Компьютер с установленной операционной системой Windows-Server 2003 может исполнять роль как клиента удаленного доступа, так и сервера.

В последнем случае на нем должна быть запущена *Служба маршрутизации и удаленного доступа* (Routing and Remote Access Service, RRAS).

11.1. Протоколы удаленного доступа

Подключение клиента к серверу удаленного доступа состоит из следующих основных этапов:

– установка соединения;

– аутентификация и авторизация клиента удаленного доступа;

– сервер удаленного доступа выступает в роли маршрутизатора, предоставляя доступ клиенту к ресурсам локальной сети, серверам баз данных, электронной почте, файловым серверам, принтерам и т. д.

Схема подключения представлена на рис. 11.1.

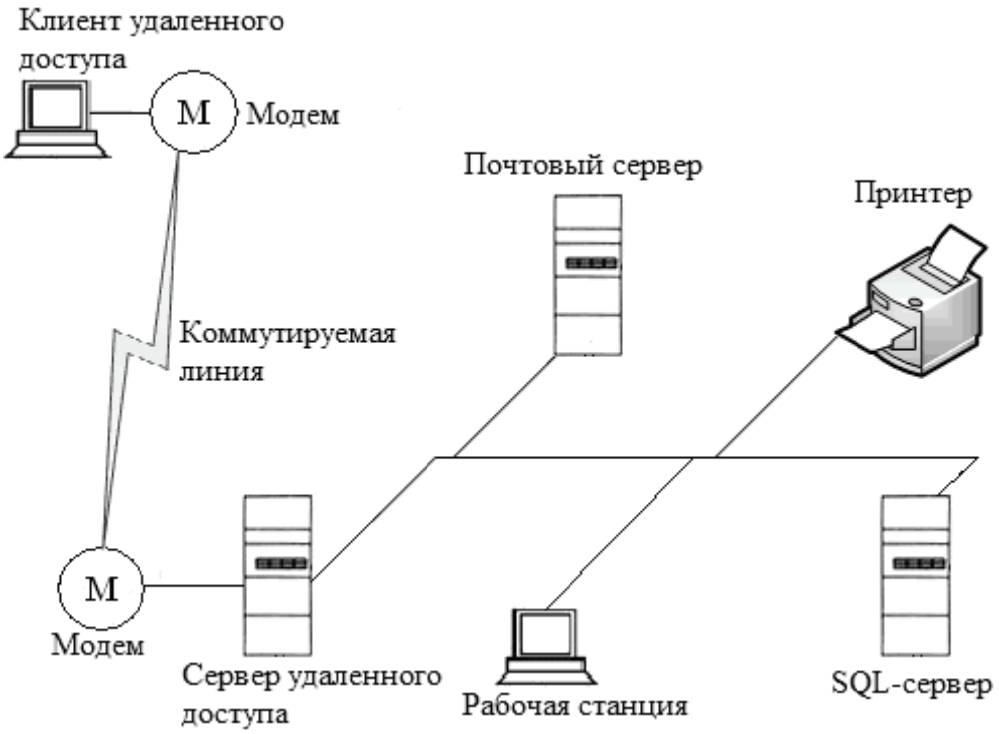


Рис. 11.1. Схема подключения удаленного доступа по коммутируемым линиям

Для соединений удаленного доступа было разработано несколько специальных протоколов. Например, операционные системы Windows Server поддерживают следующие протоколы удаленного доступа:

– *протокол SLIP* (Serial Line Internet Protocol);

– *протоколPPP* (Point-to-Point Protocol).

Протокол SLIP является одним из старейших протоколов удаленного доступа и предлагает передачу TCP/IP-пакетов без обеспечения безопасности данных и контроля целостности. Протокол описан

в RFC 1055. В Windows Server 2003 поддержка протокола SLIP реализована только на уровне клиента.

Протокол PPP предназначен для коммутируемых соединений типа «точка–точка». Это означает, что в протоколе отсутствуют средства адресации, поэтому в процессе связи могут принимать участие только два компьютера – клиент и сервер.

Протокол PPP, в отличие от SLIP, обеспечивает функции безопасности и контроля ошибок. Описание протокола содержится в RFC 1332, 1661 и 1662.

Соединение «точка – точка» устанавливается в четыре этапа.

1. *Настройка параметров канального уровня*. Клиент и сервер согласовывают максимальный размер кадра, возможность сжатия, протокол аутентификации и некоторые другие параметры.

2. *Аутентификация клиента*. Сервер осуществляет аутентификацию и авторизацию клиента на основе протокола, выбранного на предыдущем этапе.

3. *Обратный вызов* (callback). В целях безопасности может использоваться процедура обратного вызова, когда сервер разрывает соединение с клиентом и сам вызывает его по определенному телефонному номеру.

4. *Настройка протоколов верхних уровней*. Сервер отправляет клиенту список протоколов верхних уровней, отвечающих за передачу данных, шифрование и сжатие. Клиент выбирает один из подходящих протоколов списка.

Протокол PPTP является расширением протокола PPP за счет улучшения его возможностей, таких как безопасность и поддержка нескольких протоколов. PPTP может инкапсулировать протоколы TCP/IP и NetBEUI внутри дейтаграмм PPP. Дейтаграммы связаны с методами доставки без установления соединения, которые не обязательно позволяют достичь нужного получателя. В реальности PPTP – это набор стандартизованных протоколов, его соединение обычно шифруется с помощью 40-битных схем шифрования RC4 или DES.

11.2. Протоколы аутентификации удаленных клиентов

Разработано несколько протоколов, используемых для аутентификации удаленных клиентов.

1. *PAP* (Password Authentication Protocol) – протокол аутентификации по паролю (описан в RFC 1334). Самый простой протокол ау-

тентификации, в котором имя пользователя и пароль передаются открытым, незашифрованным способом. В Windows Server протокол PAP применяется только в том случае, если клиент удаленного доступа не поддерживает больше никаких протоколов.

2. *CHAP* (Challenge Handshake Authentication Protocol) – протокол аутентификации с предварительным согласованием вызова (описан в RFC 1994). В этом протоколе клиент посылает серверу пароль в виде специальной хеш-последовательности, созданной с использованием *алгоритма MD-5*. Сервер принимает *хеш (свертка) пароля клиента*, вычисляет хеш по хранимому у себя паролю и сравнивает обе последовательности. В случае совпадения соединение устанавливается, иначе происходит разрыв. Недостатком является отсутствие взаимной аутентификации, т. е. сервер аутентифицирует клиента, а клиент не получает информации о подлинности сервера.

3. *MS-CHAP* (Microsoft Challenge Handshake Authentication Protocol) – реализация протокола CHAP, разработанного Microsoft (описан в RFC 2433). Действует по принципу протокола CHAP, за исключением того, что для хеширования используется *алгоритм MD-4*, а не MD-5.

4. *MS-CHAPv2* – вторая версия протокола MS-CHAP (описан в RFC 2759), где так же как и в MS-CHAP, применяется *алгоритм хеширования MD-4*, но отличием является требование взаимной аутентификации. Между клиентом и сервером происходит обмен следующими сообщениями:

– сервер отправляет клиенту сообщение, содержащее некоторую последовательность символов, называемую ***строкой вызова***;

– клиент отправляет серверу хеш-последовательность, полученную на основе строки вызова и пароля пользователя, а также свою строку вызова для сервера;

– сервер вычисляет хеш по своей строке вызова и пользовательскому паролю, сравнивает его с полученным хешем от клиента и в случае успеха отправляет хеш, вычисленный на основе своей строки вызова, строки вызова от клиента, имени и пароля пользователя;

– клиент, получая сообщение сервера, вычисляет хеш на основе тех же данных, и в случае совпадения вычисленного хеша с полученным от сервера, процесс взаимной аутентификации считается законченным успешно.

5. *EAP* (Extensible Authentication Protocol) – расширяемый протокол аутентификации (описан в RFC 2284).

Протокол EAP – это расширение PPP, которое позволяет согласовывать произвольный метод аутентификации между удаленным клиентом и сервером. После создания соответствующего канала клиент и сервер согласовывают, какой тип механизма аутентификации EAP будет использоваться: EAP-MD5, CHAP, EAP-TLS, смарт-карты и т. д. Согласно названию протокола в него может быть добавлено любое количество методов аутентификации.

Автоматически он предоставляет следующие два метода EAP.

- EAP-MD5 CHAP. EAP-Message Digest 5 CHAP – это обязательный метод EAP, который поддерживает много одинаковых атрибутов с методом CHAP, но, кроме того, поддерживает отправку вызовов и ответов в виде сообщений EAP.
- EAP-TLS (EAP-Transport Level Security). Этот метод безопасности транспортного уровня осуществляет аутентификацию с помощью сертификатов. Данный метод является обязательным, если вы используете смарт-карты. EAP-TLS является в настоящее время наиболее сильным типом аутентификации, и для него требуется, чтобы сервер RRAS был членом домена. Он обеспечивает взаимную аутентификацию (аутентифицируются как клиент, так и сервер), шифрование, а также обмен секретными личными ключами

6. В ОС Windows Server применяются следующие типы аутентификации EAP: EAP-MD5, CHAP, EAP-TLS (Transport Level Security, безопасность на транспортном уровне), PEAP (Protected EAP, защищенный EAP).

11.3. Общая характеристика виртуальных частных сетей

В последние годы стоимость использования каналов связи Интернет стала уменьшаться и скоро стала ниже, чем цена использования коммутируемых линий. Однако при установлении соединения через Интернет серьезной проблемой является обеспечение безопасности, так как сеть является открытой, и злоумышленники могут перехватывать пакеты с конфиденциальной информацией. Решением этой проблемы стала технология виртуальных частных сетей.

Виртуальные частные сети (Virtual Private Network, VPN) – это защищенное соединение двух узлов через открытые сети. При этом организуется виртуальный канал, обеспечивающий безопасную передачу информации, а узлы, связанные VPN, могут работать так, как будто соединены напрямую.

Компьютер, инициирующий VPN-соединение, называется **VPN-клиентом**. Компьютер, с которым устанавливается соединение, называется **VPN-сервером**.

VPN-магистраль – это последовательность каналов связи открытой сети, через которые проходят пакеты виртуальной частной сети.

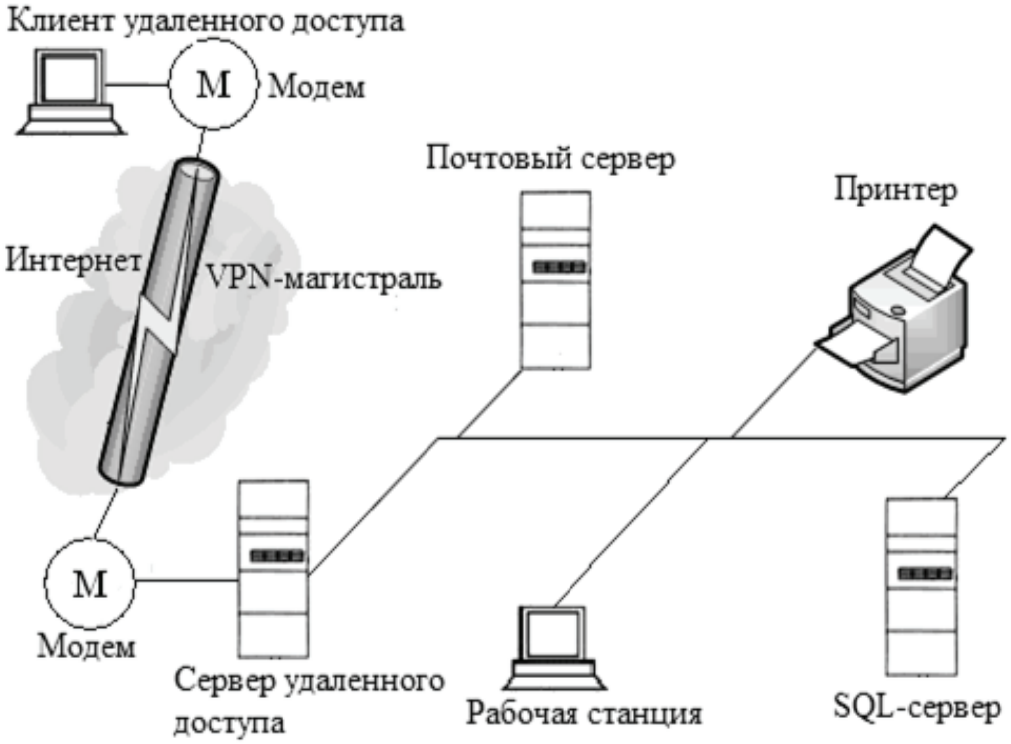
Существует два типа VPN-соединений:

– *соединение с удаленными пользователями* (Remote Access VPN Connection);

– *соединение маршрутизаторов* (Router-to-Router VPN Connection).

Соединение с удаленными пользователями осуществляется в том случае, если одиночный клиент подключается к локальной сети организации через VPN (рис. 11.2). Другие компьютеры, подключенные к VPN-клиенту, не могут получить доступ к ресурсам локальной сети.

Рис. 11.2. Схема VPN-соединения с удаленным пользователем



Соединение маршрутизаторов устанавливается между двумя локальными сетями, если узлы обеих сетей нуждаются в доступе к ресурсам друг друга (рис. 11.3). При этом один из маршрутизаторов играет роль VPN-сервера, а другой – VPN-клиента.

Рис. 11.3. Схема VPN-соединения между маршрутизаторами

11.4. Протоколы виртуальных частных сетей

Безопасность передачи IP-пакетов через Интернет в VPN реализуется с помощью туннелирования.

Туннелирование (tunneling) – это процесс включения IP-пакетов в пакеты другого формата, позволяющий передавать зашифрованные данные через открытые сети.

В современных системах Windows Server поддерживаются следующие протоколы туннелирования.

1. *PPTP* (Point-to-Point Tunneling Protocol) – *протокол туннелирования соединений «точка – точка»*, основан на протоколе PPP (описан в RFC 2637). Поддерживает все возможности, предоставляемые PPP, в частности аутентификацию по протоколам PAP, CHAP, MS-CHAP, MS-CHAPv2, EAP. Шифрование данных обеспечивается методом MPPE (Microsoft Point-to-Point Encryption), который применяет алгоритм RSA/RC4. Сжатие данных происходит по протоколу MPPC (Microsoft Point-to-Point Compression), описанному в RFC

2118.Недостатком протокола является относительно низкая скорость передачи данных.

2. *L2TP* (Layer Two Tunneling Protocol – *туннельный протокол канального уровня*) – протокол туннелирования, основанный на протоколе L2F (Layer Two Forwarding), разработанном компанией Cisco, и протоколе PPTP (описан в RFC 2661).

Поддерживает те же протоколы аутентификации, что и PPP. Для шифрования данных используется протокол IPsec. Поддерживает сжатие данных. Имеет более высокую скорость передачи данных, чем PPTP.

Процесс инкапсуляции пакета в соответствии с протоколом L2TP представлен на рис. 11.4.

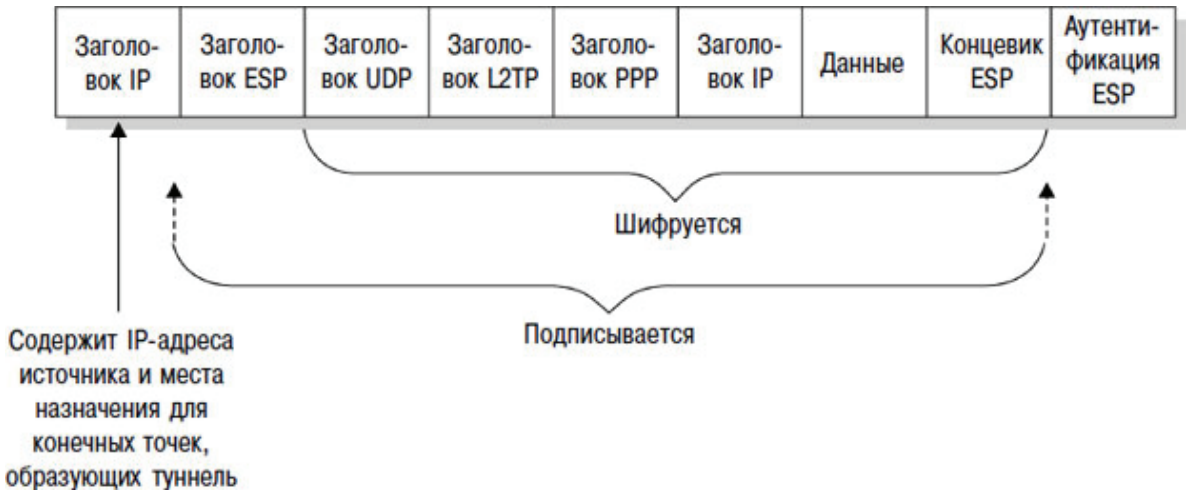


Рис. 11.4. Инкапсуляция пакетов по протоколу L2TP

L2TP часто используется для коммутируемых соединений, поскольку его структура оптимизирована для них. Более того, реализация L2TP в Windows Server предназначена в первую очередь для сетей IP и не поддерживает туннелирование в собственном режиме через сети X.25, Frame Relay или ATM.

Протокол PPTP остается единственным протоколом, который поддерживают старые версии Windows (Windows NT 4.0, Windows 98, Windows Me). Однако существует бесплатный VPN-клиент Microsoft L2TP/IPsec, который позволяет старым операционным системам Windows устанавливать соединение VPN по протоколу L2TP.

Контрольные вопросы

1. Что такое удаленный доступ?
2. Назовите виды удаленного доступа.
3. В чем отличие протоколов удаленного доступа SLIP и PPP?
4. Для чего нужна аутентификация при удаленном доступе?
5. Какие вы знаете алгоритмы аутентификации? В чем заключаются основные различия между ними?
6. Опишите алгоритм работы MS-CHAPv2.
7. Каким образом сети VPN обеспечивают безопасную передачу пакетов?
8. Назовите виды VPN-соединений.
9. Перечислите достоинства и недостатки протоколов PPTP и L2TP

ТЕМА 12. АДМИНИСТРИРОВАНИЕ С ПОМОЩЬЮ ПРОТОКОЛОВ TELNET И SSH

План

1. Протокол TELNET.

2. Протокол SSH.

3. Политика безопасности протокола SSH.

4. Схема работы SSH.

5. Сценарии как средство администрирования ОС Windows.

Данная тема рассчитана на две лекции (лекции 24–25).

12.1. Протокол TELNET

Telnet позволяет пользователю установить TCP-соединение с сервером, а затем передавать коды нажатия клавиш так, как если бы работа проводилась на консоли сервера, а также передавать клиенту текстовый отклик сервера (протокол описан в RFC 854). Служит для выполнения удаленного доступа к компьютеру с помощью командного интерпретатора. Для входа на удаленный компьютер необходима аутентификация (имя, пароль). Telnet предлагает три основные услуги.

1. Определяет сетевой виртуальный терминал (NVT), который обеспечивает стандартный интерфейс к удаленной системе.

2. Включает механизм, который позволяет клиенту и серверу согласовывать опции обмена.

3. Обеспечивает симметрию соединения, позволяя любой программе выступать в качестве клиента.

TELNET позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные сетевые виртуальные терминалы строчного типа, работающие в кодах ASCII, а также обеспечивает возможность согласования более сложных функций.

На прикладном уровне над Telnet находится либо программа поддержки реализации терминала, либо прикладной процесс в обслуживающей машине, к которому осуществляется доступ с терминала. Формат сетевого виртуального терминала достаточно прост: для данных используются семибитовые ASCII коды, для командных последовательностей – восьмибитовые октеты. Если команда TELNET вводится без аргументов,

то компьютер переходит в командный режим. При вводе команды TELNET с аргументами осуществляется связь с удаленным компьютером. После того как связь установлена, начинаются переговоры с удаленной машиной об используемых опциях. Коды опций представлены в табл.12.1. Каждая из договаривающихся сторон может посылать другой один из четырех запросов, представленных в табл. 12.2.

Таблица 12.1

Код опции

Код опции в TELNET	Описание	Номер RFC
0	Двоичный обмен	856
1	Эхо	857
2	Повторное соединение NIC	15391
3	Подавление буферизации ввода	858
4	Диалог о размере сообщения NIC	15393
5	Статус	859
6	Временная метка	860
7	Удаленный доступ и отклик	726
8	Длина выходной строки nic	20196
9	Размер выходной страницы nic	20197
10	Режим вывода символов <возврат каретки>	652
11	Вывод горизонтальной табуляции	653
12	Установка положения табуляции при выводе	654
13	Режим вывода команды смены страницы	655
14	Вывод вертикальной табуляции	656
15	Определение положения вертикальной табуляции	657
16	Режим вывода символа <перевод строки>	658
17	Расширенный набор кодов ASCII	698
18	Возврат (logout)	727
19	Байт-макро	735
20	Терминал ввода данных	732
21	Supdup	736
22	Supdup вывод	747
23	Место отправления	779
24	Тип терминала	930
25	Конец записи	885
26	Tacacs-идентификация пользователя	927
27	Пометка вывода	933
28	Код положения терминала	946
29	Режим	3270, 1041
30	X.3 PAD	1053
31	Размер окна	1073

Некоторые опции могут быть запущены только сервером, некоторые только клиентом, а некоторые обеими сторонами. Сторона может запустить опцию, если она имеет на это право.

Предложение может быть как принято другой стороной, так и отклонено.

После согласования опций TELNET переходит в режим ввода, введенный текст пересылается удаленному компьютеру. Ввод может производиться посимвольно либо построчно. В первом случае каждый символ отправляется немедленно, во втором отклик на каждое нажатие клавиш производится локально, а отправляется после нажатия клавиши Enter.

Таблица 12.2

Запросы, используемые в протоколе Telnet

Запрос	Код	Возможные ответы на запрос
WILL	251	1. Предложение для запуска 2. Принятие запроса на запуск
WONT	252	1. Отказ запросу на запуск 2. Предложение для отключения 3. Принятие запроса на отключение
DO	253	1. Одобрение предложения на запуск 2. Требование на запуск
DONT	254	1. Неодобрения предложения на запуск 2. Одобрение предложения на блокировку 3. Требование на отключение

Когда связь с удаленной машиной установлена, переход в командный режим может быть выполнен нажатием Esc, в этом режиме доступны следующий команды:

- Open имя_ЭВМ – открывает связь с ЭВМ;
- Display – отображает все или часть набора параметров TELNET;
- ? – выдает справку;
- Close – закрывает сессию TELNET.

Значения переменных можно узнать с помощью команды display (табл. 12.3).

Таблица 12.3

Переменные Telnet

Название переменной	Назначение
Echo	Определяет, будет ли отображаться на экране то, что вы вводите с клавиатуры. При значении off ввод не отображается (например, при вводе пароля).
Escape	Задает символ, который используется в качестве escape. Появление этого символа во входном потоке заставляет его и последующие символы интерпретироваться в ЭВМ, где функционирует процесс TELNET как команда.
Interrupt	Специфицирует символ прерывания процесса. Ввод его приводит к остановке процесса пользователя, работающего на удаленной ЭВМ.
Quit	Специфицирует символ, который используется пользователем на его клавиатуре для выполнения команд brake или attention.
Flushoutput	Определяет символ, который служит для прерывания процедуры вывода на удаленной ЭВМ.
EOF	Специфицирует символ, который используется для обозначения конца файла на удаленной машине.

Блок данных процедуры TELNET содержит три байта и называется командой. Формат этого блока показан на рисунке.

Формат блока данных процедуры TELNET

Первый байт в соответствии с таблицей содержит 8 единиц, далее следует байт команды. Третий октет служит для размещения кода опции, он может и отсутствовать.

В табл. 12.4 представлены наименования и коды команд TELNET, которые используются как клиентом, так и сервером в сочетании с префиксным байтом 255.

Таблица 12.4

Наименования и коды команд Telnet

Название	Код (десятичный / шестнадцатеричный)	Описание
SE	240/0xF0	Завершает согласование, начатое командой SB
NOP	241/0xF1	Нет операции
Data Mark	242/0xF2	Синхронизация (Synch) обмена данными. Эта команда всегда сопровождается TCP Urgent notification
Break	243/0xF3	Нажата кнопка «Break» или «Attention»
Interrupt Process	244/0xF4	Приостанавливает, прерывает, аварийно прекращает или завершает процесс
Abort output	245/0xF5	Подавление вывода текущего процесса. Также отправляет сигнал Synch пользователю
Are You There	246/0xF6	Отправляет обратно ответ терминала, состоящий из печатных символов
Erase character	247/0xF7	Получатель должен удалить предыдущий символ, если это возможно
Erase Line	248/0xF8	Стереть последнюю введенную строку, то есть все данные, полученные после последнего перевода строки
Go ahead	249/0xF9	Ожидается передача данных
SB	250/0xFA	Начало согласования опции, требующего передачи параметров
WILL опция	251/0xFB	Указывает на желание исполнять или подтверждает, что сейчас исполняется указанная опция
WON’T опция	252/0xFC	Указывает на отказ начать или продолжить исполнять указанную опцию
DO опция	253/0xFD	Запрос на то, чтобы другая сторона исполнила или подтвердила исполнение указанной опции
DON’T опция	254/0xFE	Требование на то, чтобы другая сторона остановила исполнение или подтвердила то, что указанная опция более не исполняется
IAC	255/0XFF	Байт данных 255

Комбинации горячих клавиш, используемых при работе с протоколом TELNET, представлены в табл. 12.5.

Отметим, что протокол TELNET не использует шифрование и поэтому уязвим для атак при применении в Интернете или локальной сети. Равную функциональность при большей защищенности обеспечивает сетевой протокол SSH.

Таблица 12.5

Коды горячих клавиш

Ctrl+E	ECHO
Ctrl+]	Escape
Ctrl+?	Erace
Ctrl+O	Flushoutput
Ctrl+C	Interrupt
Ctrl+U	Kill
Ctrl+\	Quit
Ctrl+D	EOF

12.2. Протокол SSH

Протокол SSH используется для организации безопасного входа на удаленную машину или систему и безопасной передачи информации. Существует две версии: SSH1 и SSH2. В силу того, что в первой версии было найдено много уязвимостей, дальнейшее ее развитие приостановлено, сейчас развивается SSH2.

Основное различие между двумя протоколами заключаются в том, что в SSH2 все функции разделены между тремя протоколами, в то время как SSH1 представляет собой единый неделимый протокол. Данное отличие делает SSH2 более гибким и более мощным в создании туннелей. Далее подробнее рассмотримSSH2.

Основные протоколы SSH2.

1. Протокол транспортного уровня(SSH-TRANS):

– аутентификация серверов, конфиденциальность и целостность;

– сжатие информации;

– работает с использование соединения TCP/IP, но может быть реализовано и на базе других протоколов с гарантированной доставкой.

2. Протокол аутентификации пользователей:

– для проверки полномочий клиентов;

– работает на основе протокола транспортного уровня.

3. Протокол соединений (SSH-Connect):

– мультиплексирование шифрованного туннеля в несколько логических каналов;

– работает поверх протокола аутентификации пользователей.

Сам по себе протокол транспортного уровня является достаточным для организации безопасного соединения. Он является основойSSH2. Протокол аутентификации пользователя специально отделен от протокола транспортного уровня, т. к. возможны ситуации,

когда аутентификация не требуется. При наличии каналов с высокой пропускной способностью благодаря протоколу соединений есть возможность организовывать многопоточность.

При использовании SSH2 клиент передает один запрос на обслуживание в процессе организации защищенного соединения на транспортном уровне, а другой запрос на обслуживание передается после успешной проверки полномочий клиента. Такое решение предоставляет возможность создания новых протоколов и их совместного использования с перечисленными выше протоколами.

Криптографическая защита протокола SSH не фиксирована, могут быть использованы различные алгоритмы шифрования, также возможен вариант работы без шифрования.

12.3. Политика безопасности протокола SSH

Каждому серверному хосту следует иметь ключ, причем хосты могут иметь множество ключей, созданных с использованием различных алгоритмов. Возможно использование одного ключа множеством хостов. Считается, что если хост имеет ключи, то он должен обеспечивать хотя бы один ключ для каждого из требуемых алгоритмов. Ключ сервера используется в процессе обмена ключами для подтверждения того, что клиент связывается с нужным сервером. В данном случае клиент должен заранее знать открытый ключ сервера. В рамках протокола SSH используются две модели поддержки ключей.

1. У клиента хранится локальная база данных, в которой содержатся имена хостов и их открытые ключи. Недостаток: трудоемкость поддержки ассоциаций между ключами и именами хостов.

2. Ассоциация между ключами и хостами сертифицируется специальным агентством (SCI), клиент знает только корневой ключ данного агентства и благодаря ему может проверять все ключи хостов.

Протокол позволяет отключить проверку ассоциации сервер-ключ при первом подключении к хосту. Это позволяет организовывать соединение с хостом до получения от него ключа или сертификации хоста. При таком соединении обеспечивается защита от пассивного прослушивания, но повышается уязвимость для активного перехвата. По умолчанию такая возможность отключена.

Протокол SSH допускает полное согласование алгоритмов и форматов шифрования, обеспечение целостности, обмена ключами, сжатия данных и т.д.

Алгоритмы шифрования, обеспечивающие целостность сжатия и открытия ключей, могут отличаться для каждого направления передач. В силу того, что основной задачей протокола SSH является повышение уровня безопасности, то все алгоритмы шифрования обеспечивают целостность, и открытие ключей относится к числу известных и проверенных. Используются с криптографией обоснованные размеры ключей, что позволяет рассчитывать на защиту от атак.

Все алгоритмы согласуются даже в тех случаях, когда тот или иной алгоритм не поддерживается. В этой ситуации обеспечивается простой переход к использованию других алгоритмов без изменения базового протокола.

12.4. Схема работы SSH

Для аутентификации сервера используется протокол на основе алгоритмов электронной цифровой подписи RSA, DSA.

Для аутентификации клиента может использоваться RSA, DSA, но допускается также аутентификация при помощи пароля (для обратной совместимости с TELNET) и даже IP-адреса хоста. Наиболее распространена аутентификация по паролю.

Аутентификация по паролю считается безопасной, так как пароль передается по зашифрованному каналу. Аутентификация по IP небезопасна и используется в крайнем случае.

Для создания сеансового ключа используется алгоритм Д. Ф. Хелмана, а для шифрования передаваемых данных используется симметричный алгоритм 3DES, AES.

Целостность данных проверяется алгоритмами HMAC-SHA1, HMAC-MD5. Для сжатия шифруемых данных используется алгоритм LempelZiv. Сжатие включает лишь по запросу клиента.

Передаваемые пароли шифруются ассиметричным шифрованием. Длина ключа для симметричных алгоритмов не менее 128 бит.

Алгоритм SSH предусматривает обязательную смену ключа после передачи определенного количества информации(2 Гбайта).

SSH позволяет защищаться от IP-спуфинга, DNS-спуфинга, от прослушивания паролей и передачи данных, но SSH не позволяет защитить данные при условии, что злоумышленник получил привилегированный доступ к одной из сторон передач.

12.5. Сценарии как средство администрирования ОС Windows

Основное назначение административных сценариев – автоматизация часто повторяющихся задач. Если администратор сталкивается с задачей, которую нужно выполнить более одного раза или же регулярно, имеет смысл доверить ее решение сценарию. Очевидно, что в этом случае все подобные задачи будут решаться быстро и единообразно. Кроме того, написание сценариев позволит создать инструментарий администратора, не предусмотренный в графическом интерфейсе.

Visual Basic Scripting Edition (обычно просто VBScript) – скриптовый язык программирования, интерпретируемый компонентом Windows Script Host. Он широко используется при создании скриптов в операционных системах семейства Microsoft Windows.

VBS-сценарий – это обычный текстовый файл с именем *.VBS, который пишется в приложении типа блокнот, а запускается на исполнение – двойным щелчком мыши или вызовом по имени в консоли.

Сценарии не компилируются, а интерпретируются. То есть для обработки скрипта в системе должен присутствовать интерпретатор языка VBS – Windows Script Host (WSH).

Также для создания сценария может использоваться и Java Script.

Scripting Host

Scripting host (машина сценариев) – это операционная среда сценария. Windows не имеет понятия о VBScript: если в командной строке ввести строчку кода на VBScript, система выдаст сообщение об ошибке. Когда Windows сталкивается с файлом, расширение которого указывает на файл-сценарий, операционная система передает файл машине сценариев для интерпретации. Машина интерпретирует предложенный сценарий, а затем передает сообщения сценария (по сути – запрос на регистрацию данных) в операционную систему Windows для исполнения.

Windows поддерживаетдвемашинысценариев: Microsoft Internet Explorer (IE) и Windows Script Host (WSH). Выбор той или иной машины влияет на используемые в сценарии возможности. Если применяется WSH, как чаще всего и бывает, то в сценарии могут использоваться объекты WSH, но не IE, и наоборот. Машина сценария не обязана понимать содержание всех мыслимых сценариев; воспринимается только сценарий, написанный на языке машины, и тот, который ею поддерживается. Для WSH и IE понятными являются языки VBScript и JScript.

Элементы сценария

Каждая строка сценария – это оператор, который сообщает компьютеру, что следует сделать. Исполняемые операторы обычно имеют форму типа «действие-объект»: описывается само действие и тот объект, над которым действие совершается. Сценарий может содержать условия, при наличии которых указанные операторы должны быть выполнены. Хост сценария интерпретирует строки кода слева направо и сверху вниз так, что можно, например, получив некоторые данные в строке 10, использовать их в 30-й строке. Исключение составляют процедуры. Процедуры (функции и подпрограммы) – это набор операторов, которые выполняются только при явном обращении к ним. В данном случае процедура сразу же начинает выполняться независимо от того, из какого места кода было обращение.

Исполняемые части сценария называются операторами. Неисполняемая часть сценария называется комментарием и должна предваряться апострофом (') или ключевым словом Rem. Например:

```
Rem Это комментарий
```

или

```
' Это комментарий
```

Комментарий может занимать всю строку целиком или быть частью строки, содержащей исполняемый код. Сценарий следует документировать, чтобы не участвующий в его написании человек (или даже незнакомый с лексикой его языка) смог легко понять, для чего сценарий предназначен. Иногда в целях отладки программы в начале исполняемой строки ставят признак комментария.

VBScript понимает четыре типа данных: числа (number); строки (string); дата и время (date and time); булевы данные (boolean). Примеры чисел – 2 или 9458. Строки – это любая комбинация символов, заключенная в двойные кавычки, например "рыба" и "Это строка %@#^>". Дата и время должны находиться внутри символов # и выглядеть соответственно. Так, например, #16 January 1968# и #1/01/02 11:45 PM# – нормальные с точки зрения VBScript данные. Булевы данные – TRUE или FALSE, например x<x+1 = TRUE. Булевы данные часто бывают нужны при тестировании сценария.

VBScript рассматривает перечисленные четыре типа данных как подмножество другого типа данных – variant, который может содержать данные любого вида. Таким образом, VBScript можно не сооб-

щать, с данными какого типа вы работаете, но нужно иметь в виду, что некоторые задачи выполняются в Visual Basic (VB) и VBScript с описанными типами данных по-разному. Группы однотипных данных называются массивами (array).

Для простоты работы с данными VBScript поддерживает еще два типа данных, не имеющих никакого начального значения (null-данные), которые можно присваивать переменным (variable) и константам (constant) сценария. Значения переменных в ходе выполнения программы могут меняться, но их имена при этом остаются прежними. Константы при выполнении сценария имеют только одно значение и изменяться не могут.

Передать данные в сценарий можно двумя способами. Во-первых, их в явном виде прописывают в телепрограммы. Например, ""\\server\sharedfolder"" – обычное использование в сценарии строковых данных для обозначения пути. Во-вторых, передать нужные данные во входном потоке в сценарий. Кроме того, по ходу обработки сценарий может самостоятельно генерировать данные (например, вычислить дату двумя неделями позднее текущей), а затем использовать их.

Манипулировать данными можно с помощью операторов (operator) символов, которые обычно применяются для обозначения математических функций. Какие-то операторы имеют более высокий приоритет, какие-то – более низкий, и это влияет на порядок вычисления выражений (expression). Выражение есть некоторое вычисление, в которое могут быть включены числа, переменные, строки, константы. В выражениях могут использоваться операторы. Например, выражение dInputDate + 2 = dNewDate означает, что к значению переменной dInputDate добавляется 2, и результат вычисления вновь присваивается переменной dNewDate.

Функции и подпрограммы

VBScript имеет набор встроенных функций, которые позволяют выполнять некоторые операции без подробного описания решаемой задачи. С помощью встроенных функций можно манипулировать числами, строками, значениями даты и времени, массивами. В состав VBScript также входят функции преобразования данных одного типа в другой. Например, VBScript обычно исходит из того, что число, допустим, 45, имеет тип «число», но при необходимости его можно рассматривать как данные строкового типа.

VBScript предусматривает создание собственных функций пользователя (user-defined function, UDF) для выполнения каких-то специфических задач. Например:

```
FunctionTestFunct
TestFunct = Sqr(9) + 2
EndFunction
```

Пользовательская функция TestFunct работает со встроенной функцией Sqr для извлечения квадратного корня из девяти и добавления к полученному результату двух. UDF, как и встроенная функция, может использовать аргументы.

TestFunct UDF возвращает результат в основное тело программы. Подпрограмма выполняет некоторые действия, но ничего не возвращает в основной код в качестве результата. Программист может задействовать подпрограмму несколько раз, при необходимости использовать один и тот же участок кода. Подпрограмма

```
Sub AskUserName
WScript.Echo _
"Please type a username."
WScript.Quit
EndSub
```

использует возможности объекта WScript для вывода на экран некоторого сообщения, после чего завершает свою работу. Функции и подпрограммы могут задействовать значения переменных, декларированных внутри основного кода сценария, или же использовать собственные переменные.

Объекты в сценарии

Объект (Object) представляет собой физическую или логическую часть вычислительной среды, например, дисковод или имя учетной записи. Конечно, можно программировать, не прибегая к объектам, но большинство сценариев управления работает с объектами. Если используется WSH, VBScript может обращаться к объектам, изначально присущим WSH, например, представляющим файлы, каталоги, части реестра; VBScript также поддерживает объекты Windows Management Instrumentation (WMI) и Active Directory Service Interfaces (ADSI). Объекты WMI связаны с физическими и логическими частями вычислительной системы: например, адресами IP, файловыми системами, сетевыми адаптерами. ADSI-объекты представляют ресурсы службы каталогов: в частности, Active Directory (AD) или иные поддерживаемые каталоги, скажем Windows NT 4.0 SAM. Статические группы

объектов одинаковой природы называются классами, а группы, описываемые пользователем, библиотеками.

Объекты имеют свойства и методы. Объект определяется его свойствами (т. е. IP Address – это свойство объекта Network Card, а 12.4.21.197 – значение данного свойства). Методы – это действия, которые могут выполняться над объектом (Copy – один из методов объекта File). Не все объекты имеют методы. Свойства и методы используются при написании кода одинаково: сначала следует объект, затем ставится точка, далее название метода или свойства (например, ObjectName.PropertyName). Объекты могут содержать другие объекты. В частности, объект WSH WScript включает подчиненный объект WshArguments, который является набором аргументов, передаваемых при вызове файлу-сценарию. Для выделения первого элемента строки аргументов используется WScript.Arguments(0). Как было видно при обсуждении объектов WSH, формальное имя подчиненных объектов не совпадает с именем, используемым при обращении к ним в сценарии.

Рекомендации по разработке сценариев

1. Строки сценария должны быть короткими – они легче читаются. VBScript допускает использование знака подчеркивания для разрыва строки, а конструкция If ...Then ... Else поможет избавиться от нагромождения логики в одной строке.

2. При работе с Windows Script Host (WSH) следует использовать среду командной строки (command-line environment). WSH может исполняться в двух средах: в среде командной строки и в графической среде (по умолчанию). В первом случае вывод направляется в командное окно, если только программист не перенаправит поток данных в другое место. В графической среде вывод поступает в окна сообщений.

Чаще всего используется командная среда. Некоторые операции в графической среде не работают, и если несколько строк кода генерируют вывод, то для каждой из них понадобится свое окошко сообщений. В результате работа сценария будет приостанавливаться до тех пор, пока оператор не нажмет кнопку ОК. Для исполнения сценария в командной среде следует воспользоваться одним из двух способов: предварять каждую команду сценария вызовом cscript, например.

cscript1.vbs

Существует возможность установить среду командной строки в виде среды по умолчанию.

```
wscript //h:cscript //s
```

3. Присваивайте имена переменным в соответствии с типом представляемых данных (т. е. имена строковых переменных должны начинаться с s, объектных – с o). Подобная практика поможет при отладке кода. В ряде случаев при несоответствии типов данных объявленным переменным, работа сценария будет протекать не так, как ожидается. А если тип данных ассоциируется с названием переменной, ошибки из-за несоответствия типов переменных и данных будут выявляться быстрее.

4. Заранее и в явном виде описывайте переменные. Хотя переменные разрешено описывать неявно (просто присваивая им значения), их применение можно запретить. Для этого используйте утверждение Option Explicit в самом начале файла-сценария. С этого момента любая применяемая в коде программы переменная должна явно описываться оператором Dim, что позволит ограничить число ошибок из-за случайно или неверно набранных переменных.

Контрольные вопросы

1. Опишите назначение протоколов TELNET и SSH.
2. Приведите примеры команд протокола TELNET.
3. Какие запросы используют в протоколе TELNET?
4. Приведите формат блока данных процедуры.
5. В чем различия между протоколами SSH1 и SSH2?
6. Опишите политику безопасности протокола SSH.
7. Опишите схему работы протокола SSH.
8. Каково назначение сценариев?
9. Опишите объекты в сценариях.

ЛИТЕРАТУРА

1. 1. Котельников, Е. В. Сетевое администрирование на основе Microsoft Windows Server 2003 : курс лекций / Е. В. Котельников. – 2007. – 103 с.
2. Урбанович, П. П. Компьютерные сети: учеб. пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 400 с.
3. Бозуэлл, У. Внутренний мир Windows Server 2003, SP1 и R2 / У. Бозуэлл. – М.: Вильямс, 2006. – 1264 с.
4. Станек, У. Справочник администратора. Microsoft Windows Server 2003 / У. Станек. – М.: Русская редакция, 2003. – 640 с.
5. Ханикат, Д. Знакомство с Microsoft Windows Server 2003 / Д. Ханикат. – М.: Русская редакция, 2003. – 464 с.
6. Зубанов, Ф. Н. Active Directory. Подход профессионала / Ф. Н. Зубанов. – М.: Русская редакция, 2003. – 544 с.
7. Полак-Брагинский, А. Администрирование сети на примерах / А. П. Полак-Брагинский. – СПБ.: БВХ-Петербург, 2005. – 320 с.

Учебное издание

Романенко Дмитрий Михайлович

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Тексты лекций

Редактор *К. В. Великода*
Компьютерная верстка *К. В. Великода*
Корректор *К. В. Великода*

Издатель:
УО «Белорусский государственный технологический университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/227 от 20.03.2014.
Ул. Свердлова, 13а, 220006, г. Минск