Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Д. М. Романенко

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебно-методическое пособие по выполнению тестовых заданий для студентов специальности 1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» заочной формы обучения

Минск 2014

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

Рассмотрено и рекомендовано редакционно-издательским советом
Белорусского государственного технологического университета

Рецензенты:
доцент, кандидат технических наук, доцент кафедры
информационных технологий автоматизированных систем БГУИР
О. В. Герман;
доцент, кандидат технических наук, заведующий кафедрой
полиграфического оборудования и систем обработки информации БГТУ
М. С. Шмаков

Романенко, Д. М.

Р31 Администрирование информационных систем : учеб.-метод. пособие по выполнению тестовых заданий для студентов специальности 1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» заочной формы обучения / Д. М. Романенко. – Минск : БГТУ, 2014. – 94 с.

В учебно-методическом пособии даны понятие информационной системы, основные задачи сетевого администрирования. Описаны особенности построения и использования RAID-массивов. Рассмотрены правила IP (статической, динамической) и символьной (DNS, NetBios) адресации, методы обеспечения безопасности. Изложены протоколы и принципы маршрутизации. Приведены рекомендации для подготовки студентов к тестовым заданиям. Подробно на примерах с иллюстрациями рассмотрены решения наиболее важных для администратора задач.

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, что вызывает необходимость освоения вычислительной техники будущим инженером-программистом в объеме, позволяющем использовать ее на должном уровне при решении конкретных практических задач.

Цель дисциплины «Администрирование информационных систем» – обучение студентов общим методам создания, настройки и администрирования сетей, а также персональных компьютеров. Задача дисциплины – изучение студентами сетевых операционных систем на базе платформ Windows, а также методов управления информационными системами.

Курс состоит из лекционной части, лабораторного практикума и контрольных заданий в виде тестов.

Основная задача пособия – дать студентам общие систематизированные сведения об организации и структуре важной отрасли, которая затрагивает профессиональные, бытовые, познавательно-развлекательные сферы жизнедеятельности человека, которая интенсивно меняется, развивается. В пособии в доступной форме даны базовые понятия, цели, задачи сетевого администрирования, принципы и правила настройки и использования стека протоколов TCP/IP, методы настройки различных типов адресации в IP-сетях. Предполагается рассмотрение эффективных решений задач управления пользователями и ресурсами сети, а также приобретения необходимых знаний и навыков в области безопасности функционирования распределенной информационной системы.

Учебный материал структурирован в виде шести разделов, поделенных на подразделы. Важные моменты, требующие решения практических задач, подробно рассмотрены на примерах. Вопросы, представленные в конце каждой главы, лишь определяют круг проблем, охватываемых тестом. В тестах может быть использована как другая формулировка вопросов, так и некоторые новые вопросы. Также вопросы, базирующиеся на приведенных примерах, в тесте будут носить как теоретический характер (по теоретической составляющей, представленной в данном пособии), так и практический характер. Настоящее пособие призвано помочь студентам в овладении соответствующими знаниями и в подготовке к тесту по курсу «Администрирование информационных систем».

Для успешного освоения курса желательны базовые знания по основам компьютерных сетей, хотя все необходимые сведения приводятся в данном учебно-методическом пособии.

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

1.1. Цели и задачи администрирования информационных систем

Информационная система (ИС) – 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 является обеспечение надежности, бесперебойности, безопасной работы всей системы с требуемым уровнем производительности. Информационная система обязательно будет базироваться на компьютерной сети, при этом сети бывают условно 3 видов:

- локальные сети (LAN, Local Area Network);
- глобальные сети (WAN, Wide Area Network);
- городские сети (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 компонентов, 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 компонентов.

Активное сетевое оборудование включает в себя такие виды устройств, как повторители (репитеры), мосты, концентраторы, коммутаторы, маршрутизаторы. В корпоративной сети может быть ис-

пользован 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 сетевых 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.

Перечислим 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, 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 замена сетевого принтера с соответствующей настройкой рабочих мест.

3. *Установка и настройка сетевых протоколов*. Данная задача включает в себя выполнение таких работ – 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 Active Directory), 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 повышения эффективности работы сети*. В задачу 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 сетевых узлов включает в себя наблюдение за функционированием сетевых узлов и корректностью выполнения возложенных на данные узлы функций. *Мониторинг сетевого трафика*. Мониторинг сетевого трафика позволяет обнаружить и ликвидировать 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.).

8. *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 наибольшее распространение получили две сетевые модели, которые являются стандартами для описания межсетевого взаимодействия: модель OSI и модель TCP/IP. Обе модели 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 [2, глава 3 и 4].

1.3. Серверные ОС и инструменты администрирования

Ключевую роль в выполнении задач администрирования играет правильная настройка центрального элемента информационной системы – серверной операционной системы (ОС) Windows Server. Такая ОС предоставляет системному администратору широкий набор инструментов для решения задач управления. Основными из этих инструментов являются следующие:

• 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 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 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 альтернативы 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 имеет стандартного графического аналога (эти утилиты рассматриваются во второй лекции).

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

Выводы

Таким образом, в первом разделе описаны основные цели сетевого администрирования, рассмотрены типовые задачи, с решением которых постоянно сталкиваются администраторы. Приведены основные методы решения представленных типовых задач. Необходимо отметить, что данный раздел, посвященный проблематике сетевого администрирования, носит вводный характер. Теоретические и практические составляющие основных методов организации и управления информационными системами будут рассмотрены далее.

КОНРОЛЬНЫЕ 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.

5. Какова основная цель сетевого администрирования?

6. Назовите основные инструменты администрирования. Приведите примеры.

7. Назовите основные виды задач сетевого администрирования.

8. Приведите примеры конкретных задач на каждый вид.

ТЕМА 2. RAID-МАССИВЫ

2.1. 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 основным целям добавилась цель, связанная с сохранением данных в случае отказа части оборудования. Именно эти цели сделали RAID-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: *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 вообще весь массив является одним большим диском.

Зеркалирование – технология, позволяющая повысить надежность системы (рис. 2.1). 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 стопроцентную 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. 2.2). 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. Такое дорогое решение используется только во внешних RAID-массивах, предназначенных для ответственных приложений.

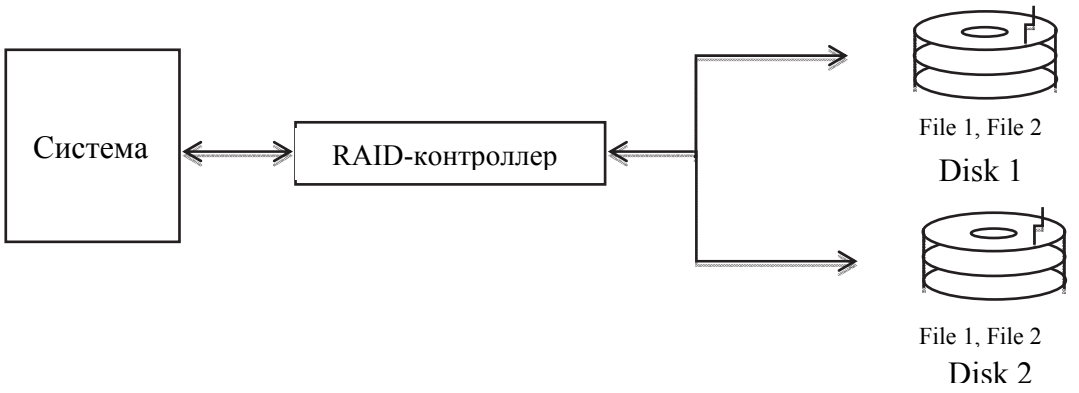

Рис. 2.1. Принципиальная схема технологии «зеркалирования»

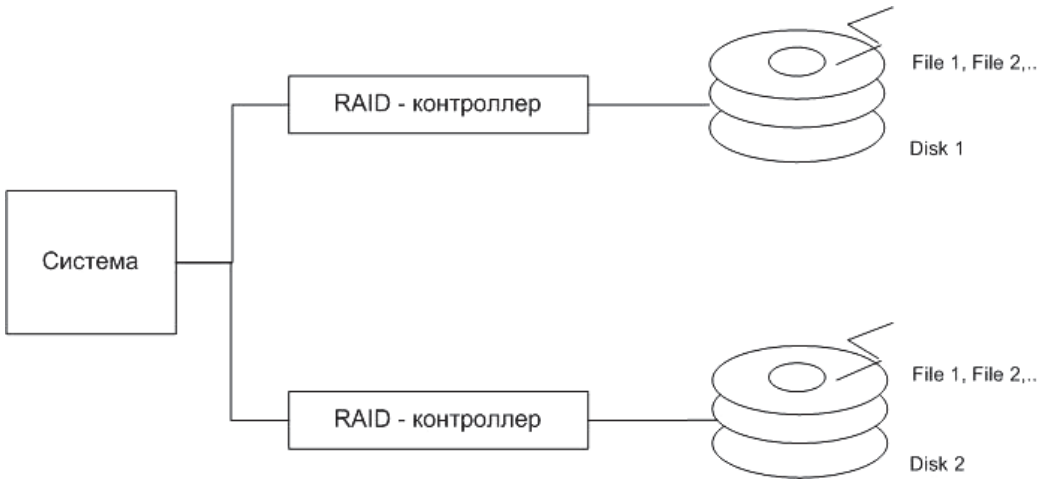

Рис. 2.2. Принципиальная схема технологии «дуплекс»

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. 2.3).

Данная технология позволяет увеличить линейную скорость записи чтения. Основной проблемой является ненадежность – 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.

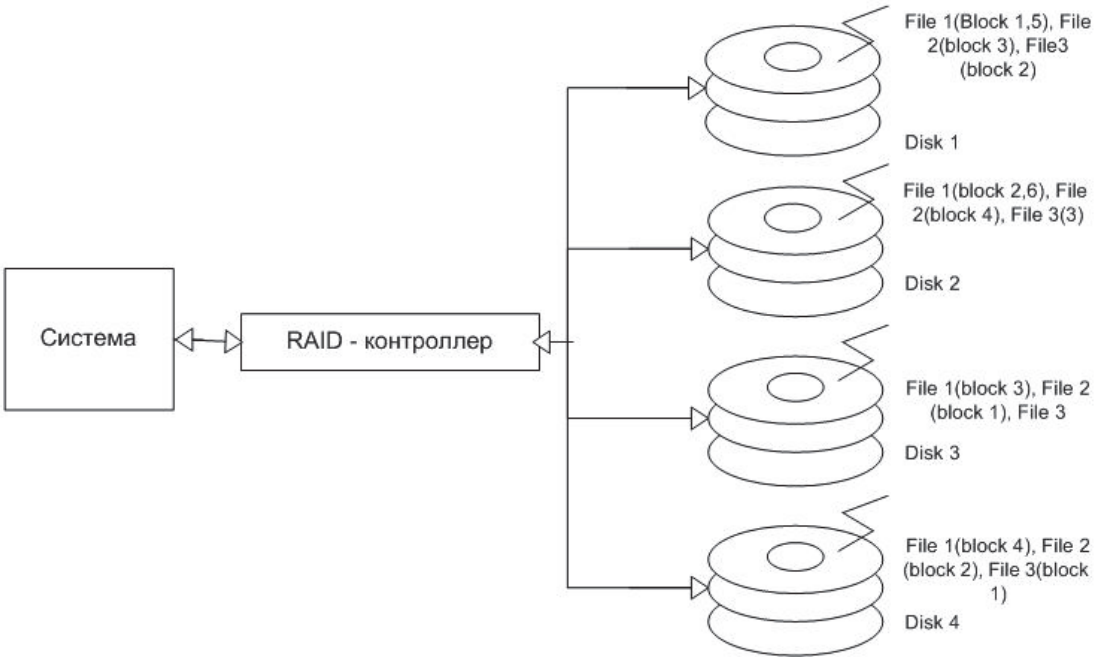

Рис. 2.3. Принципиальная схема технологии «чередование»

Согласно данной технологии используется n+1 накопитель, при этом на n накопителей записывается информация в виде отдельных блоков (как в чередовании). На n+1диске 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, однако он одинаков при любом количестве дисков и составляет емкость одного диска, то есть при 5 дисках в массиве «теряется» всего 20% емкости.

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

2.2. 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. 2.4).

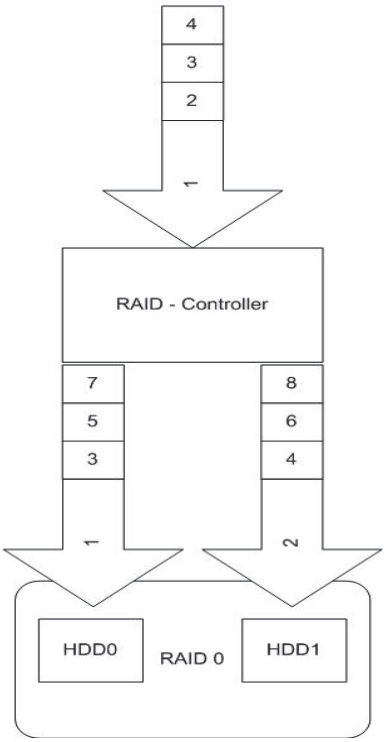

Рис. 2.4. Принципиальная схема RAID 0

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 использовать дешевый RAID-контроллер или даже его программную реализацию (рис. 2.5).

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.

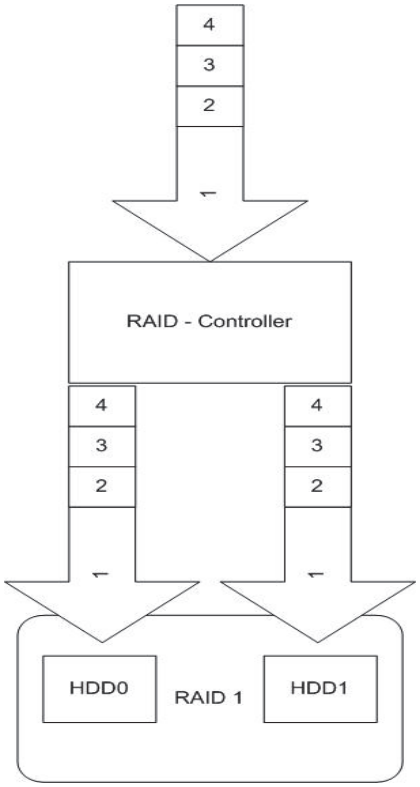

Рис. 2.5. Принципиальная схема RAID 1

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 это практического применения он не нашел.

RAID 3 и **RAID 4.** Данные хранятся на одном диске, применяется соединение чередования и четности (рис. 2.6).

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. RAID 4 отличается только размером блока данных.

RAID 5. Это наиболее распространенный массив, характеризующийся применением идей чередования и четности, но контрольные суммы хранятся не на одном диске, а разбрасывается по всем. Глав-

ный принцип распределения электроблоков заключается в следующем – они не должны располагаться на том же диске, с которого была закодирована информация (рис. 2.7).

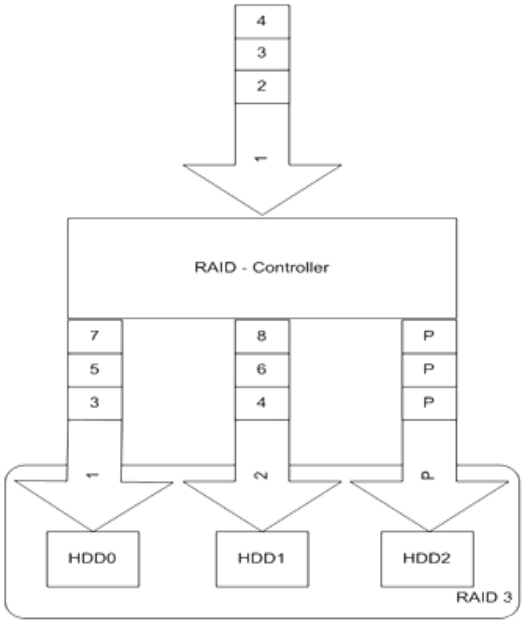

Рис. 2.6. Принципиальная схема RAID 3 и RAID 4

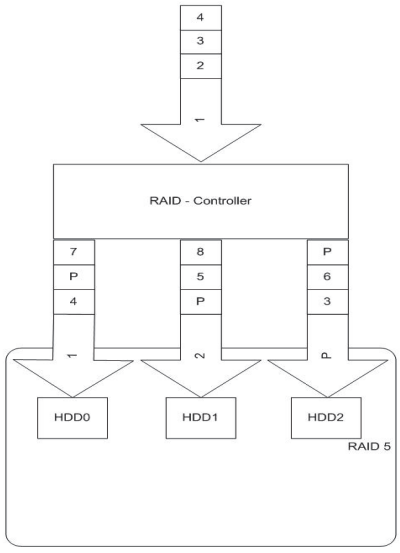

Рис. 2.7. Принципиальная схема RAID 5

Характеризуется высокой скоростью записи с достаточно высокой надежностью, при этом информационная емкость RAID 5 рассматривается как количество дисков минус единица и 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). Если до полного восстановления массива произойдет выход из строя или возникнет невосстановимая ошибка чтения хотя бы на еще 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 RAID 5 данная конфигурация дисков является расточительной, эффективнее использовать RAID 6. Целесообразность использования spare-диска проявляется в системе из нескольких томов RAID 5, в которой spare-диск проинициализирован для каждого из томов RAID 5 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.

2.3. Составные RAID-массивы

У основных уровней RAID есть свои достоинства и недостатки. Для объединения достоинств различных RAID-массивов и нивелиро-

вания 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 строится 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 RAID X-массивов объединены в RAID-массивы уровня Y.

RAID 1+0 (0+1). RAID 1+0 часто называют зеркалом страйпов, 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. Основным недостатком обоих массивов является достаточно низкий процент использования емкости накопителя.

Пример 2.1. Рассмотрим пример построения RAID-массива уровня 1+0. Допустим, данный RAID-массив необходимо построить, используя 4 жестких диска.

Этап 1. Формируем два сета по два жестких диска в каждом. В рамках одного сета пара жестких дисков связывается с использованием RAID 1 – с зеркалированием информации, тем самым повышая надежность хранения (как показано на рис. 2.8).

Рис. 2.8. Формирование двух сетов

Этап 2. В силу того, что вторым, применяемым в рамках составного RAID 1+0, будет RAID 0, то в разные сеты должна отправляться различная информация. Например, в первый сет – 1-й, 3-й, 5-й и т. д. блоки, а во второй сет соответственно – 2-й, 4-й, 6-й и т. д (как показано на рис. 2.9).

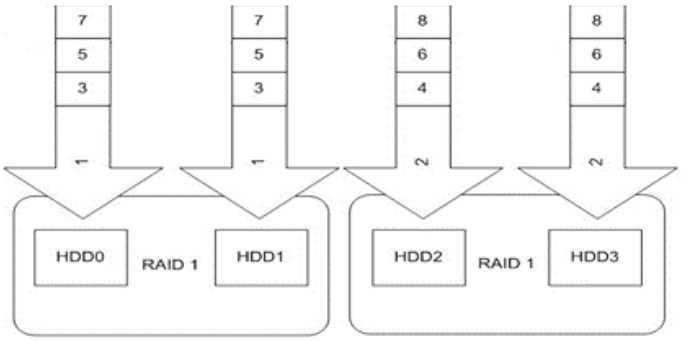

Рис. 2.9. Распределение информации по сетам

Этап 3. Два сета жестких дисков (фактически на RAID-массиве уровня 1) далее объединяются в RAID 0, что позволяет также добиться достаточно высокой скорости чтения/записи информации при неплохом уровне надежности. Таким образом, в итоге получается составной RAID-массив уровня 1+0 (рис. 2.10).

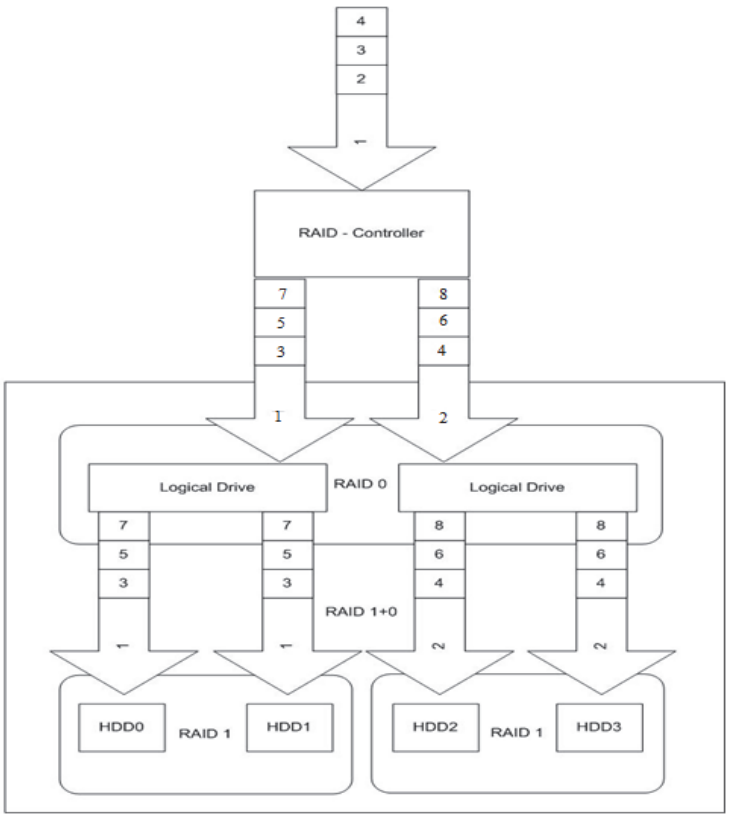

Рис. 2.10. Принципиальная схема RAID 1+0

Надежность и скорость RAID 1+0 полностью зависят от числа сетов, входящих в составной массив, а также от числа жестких дисков

в сетах. Говорят, что при построении RAID-массивов на 10 дисках система может оставаться работоспособной при отказе 5 жестких дисков.

Рассмотрим несколько вариантов построения RAID 1+0 с использованием 6 жестких дисков.

Пример 2.2. При таком числе жестких дисков двумя «крайними» (с ярко выраженными свойствами либо скорости, либо надежности) вариантами организации будут:

1. Составной массив состоит из 2-х сетов по 3 жестких диска в каждом (рис. 2.11). Массив в данном случае ориентирован на достижение максимума надежности (в каждом сете информация зеркалируется на 3 жестких диска) при условии обеспечения неплохого уровня быстродействия – информация разбивается на блоки и с применением принципа «чередования» пишется в 2 сета.

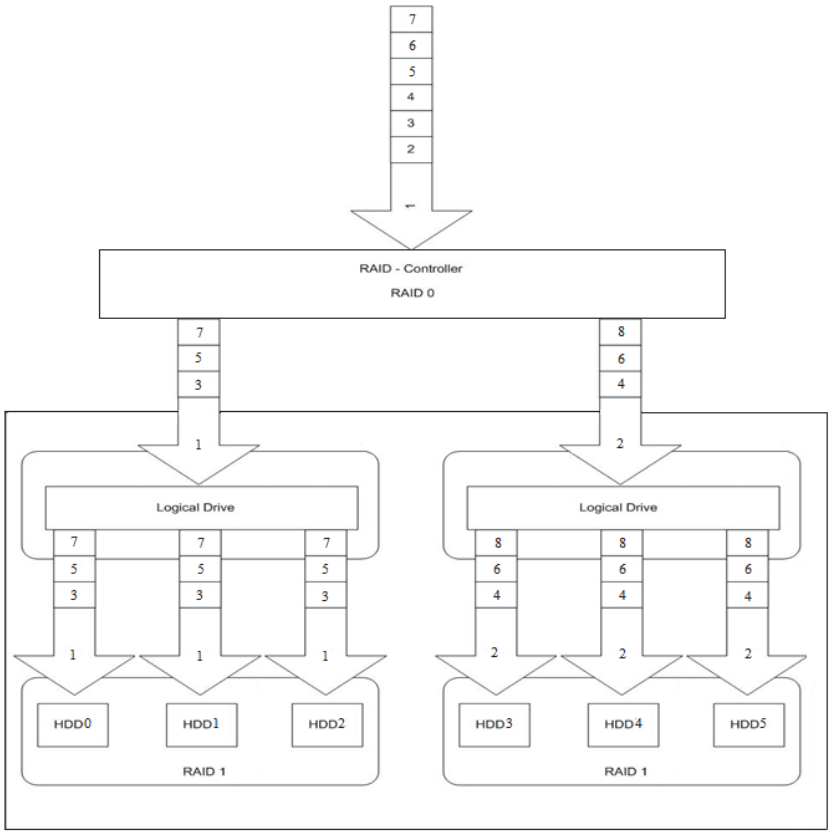

Рис. 2.11. RAID 1+0 (схема: 2 сета по 3 HDD в каждом)

2. Составной массив включает в себя 3 сета по 2 жестких диска в каждом (рис. 2.12). Массив в данном случае ориентирован на достижение максимальной скорости записи/чтения (информация разбивается на блоки и с применением принципа «чередования» записывается

в 3 сета) при условии обеспечения также неплохого уровня надежности – информация зеркалируется на два жестких диска в каждом сете).

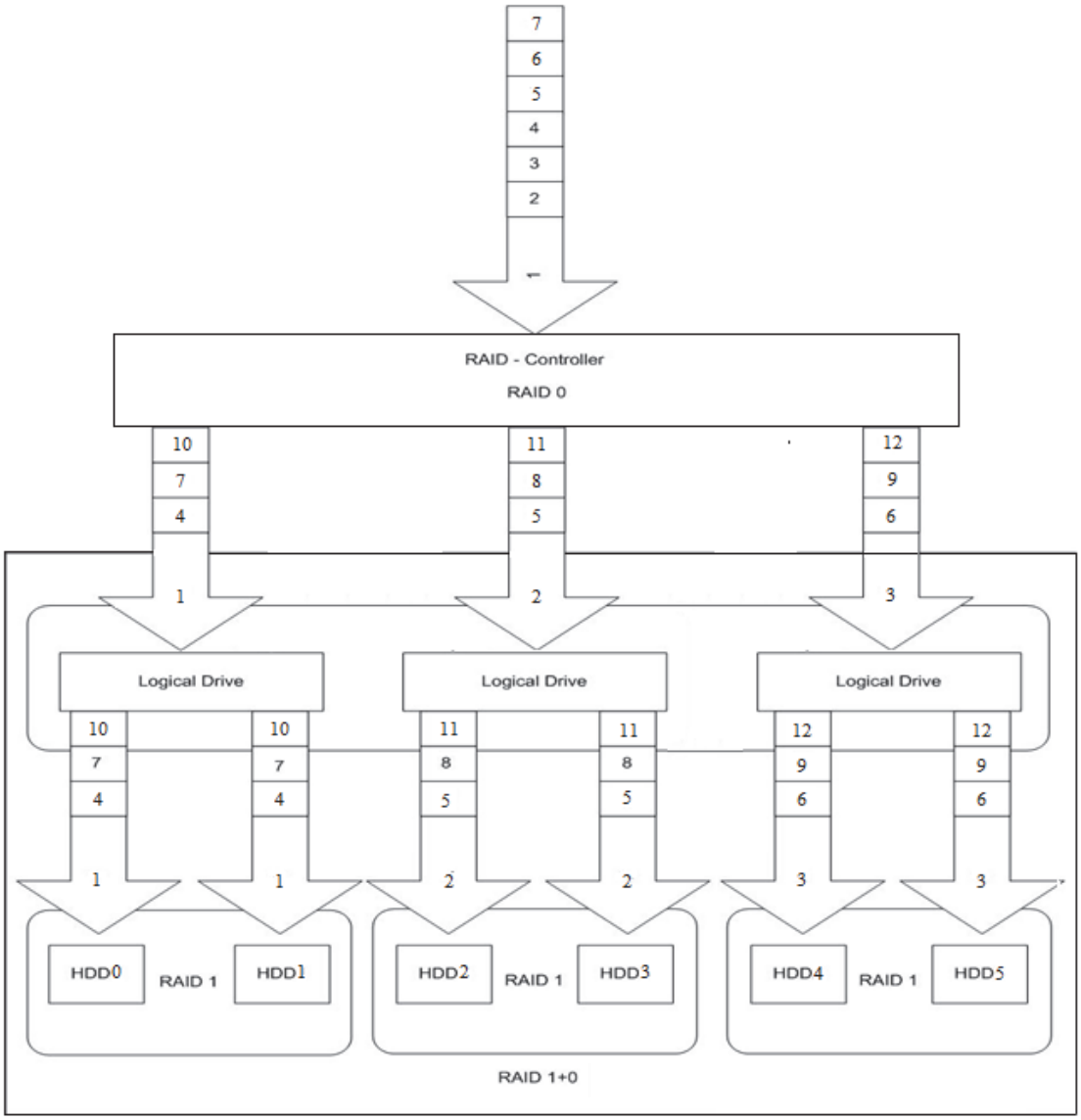

Рис. 2.12. RAID 1+0 (схема: 3 сетов по 2 HDD в каждом)

Очевидно, что схема составного RAID 1+0 (3 сета по 2 жестких диска), представленная на рис. 2.12, превосходит схему составного RAID 1+0 (2 сета по 3 жестких диска), представленную на рис. 2.11, по скорости, так как чередование осуществляется на 3 сета, однако уступает по надежности. Так RAID 1+0 (3 сета по 2 жестких диска) может сохранить работоспособность при выходе 3-х жестких дисков (рис. 2.13), а RAID 1+0 (2 сета по 3 жестких диска) – 4-х жестких дисков (рис. 2.14.). На рис 2.13 и 2.14 перечеркнуты жесткие диски, кото-

рые, например, могут выйти из строя, при этом хранящаяся на RAID 1+0 информация не будет утеряна.

Отметим, что независимо от числа сетов и жестких дисков в них эффективность использования дискового пространства для RAID 1+0 будет составлять 50%.

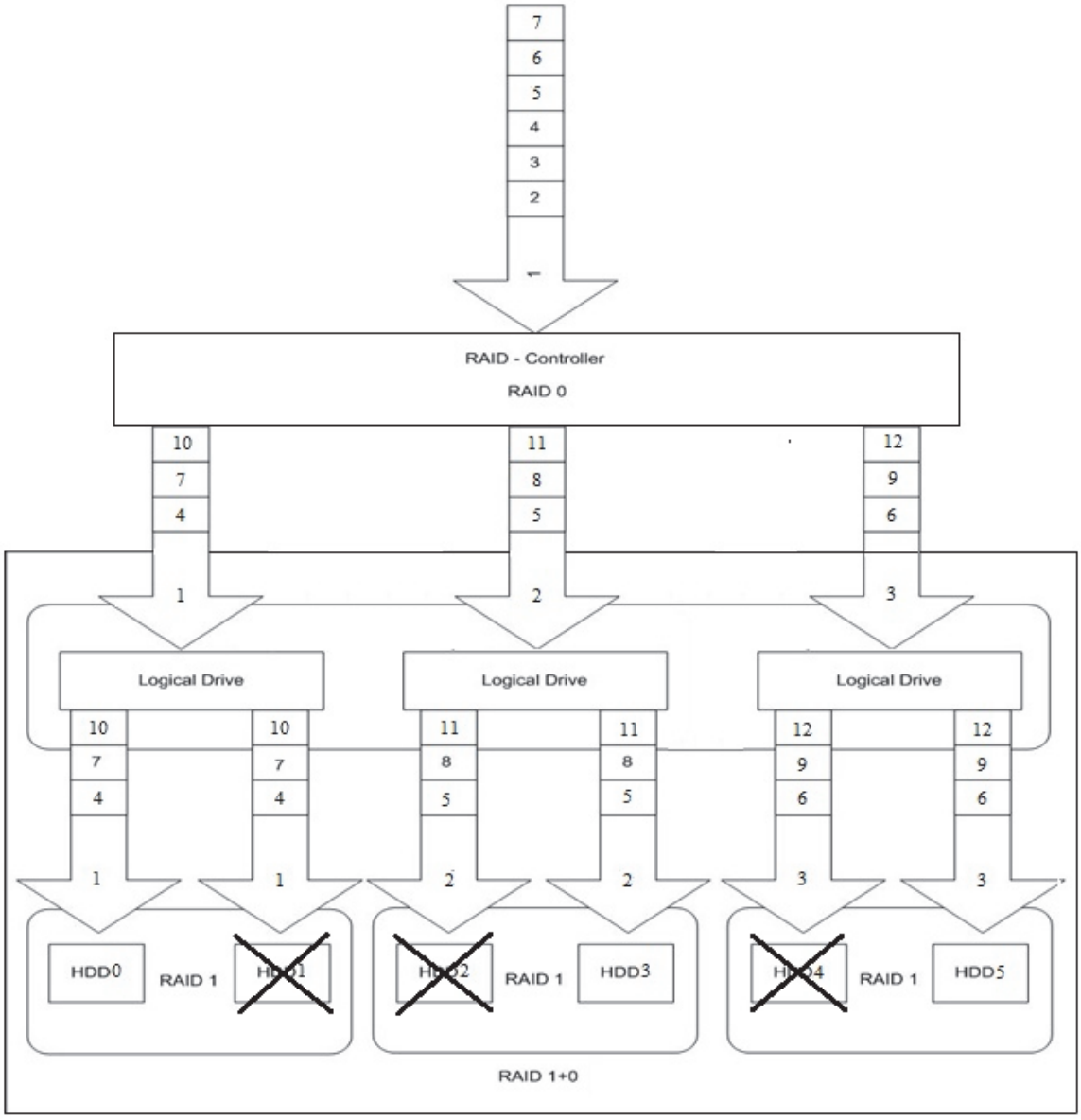

Рис. 2.13. Работоспособность RAID 1+0 (схема: 3 сетов по 2 HDD в каждом) при выходе из строя нескольких жестких дисков

Таким образом, комбинируя RAID 1+0 по числу сетов и жестких дисков, можно добиться оптимального сочетания надежности хранения информации и скорости чтения/записи.

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

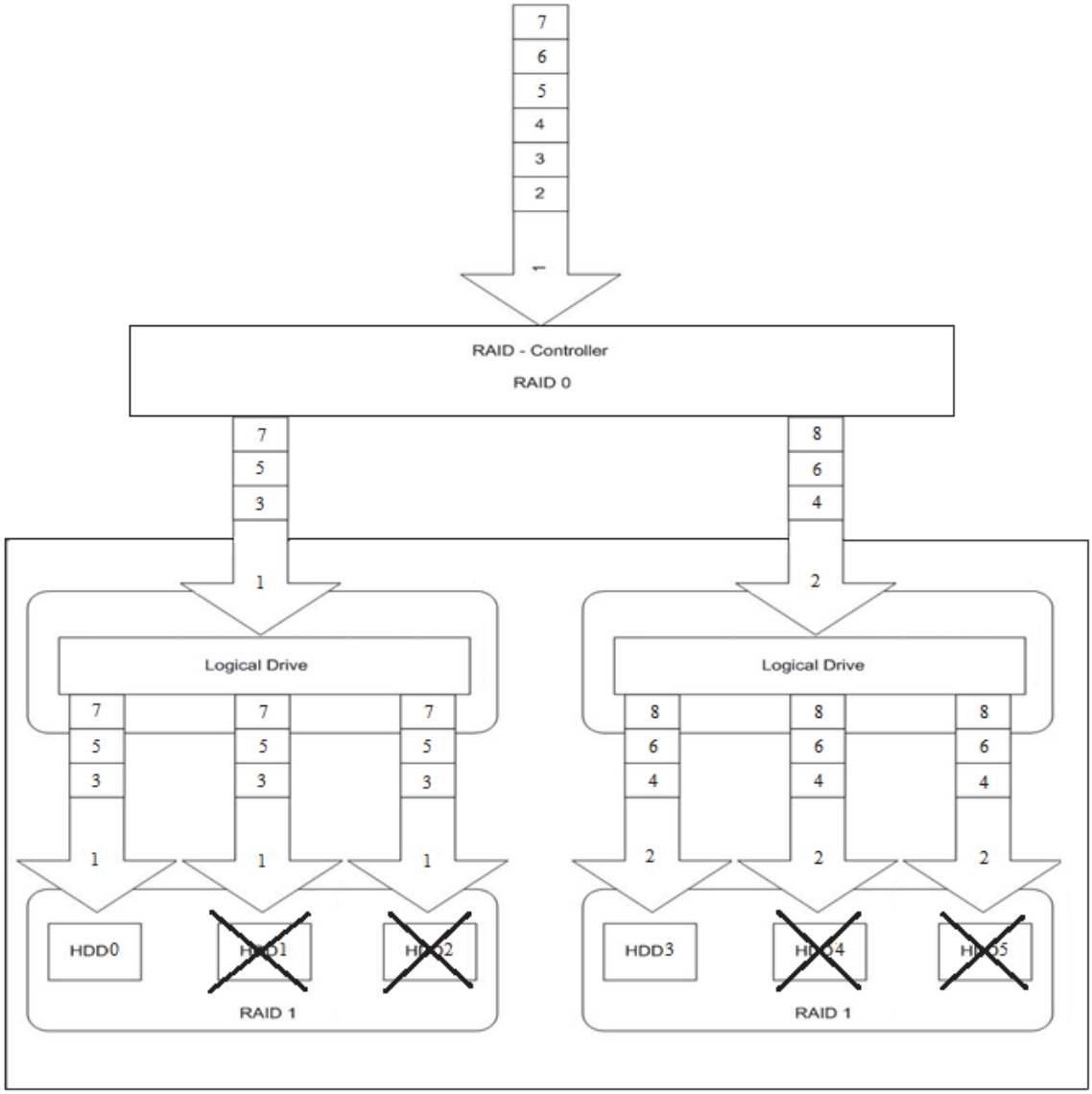

Рис. 2.14. Работоспособность RAID 1+0 (схема: 3 сетов по 2 HDD в каждом) при выходе из строя нескольких жестких дисков

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 пишется 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 заключается в высоком проценте использования емкости дисков, а также в достаточно высокой скорости чтения данных. Основные недостатки – сложность реализации RAID-контроллера и цена.

RAID 5+0 (0+5) (рис. 2.15). RAID 0+5 представляет собой набор страйпов, на основе которого построен RAID 5. 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: 2 RAID-массива уровня 5 объединяются в страйп, что позволяет получить высокую производительность при работе с файлами малого размера, поэтому RAID 5+0 часто используется на Web–серверах.

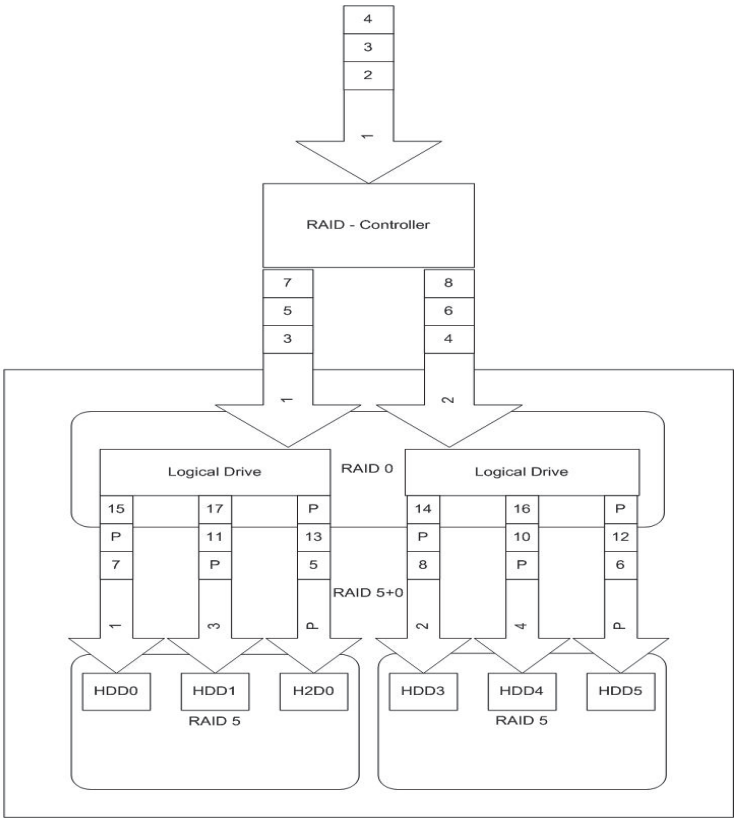

Рис. 2.15. Принципиальная схема RAID 5+0

RAID 1+5 (5+1). Этот уровень построен на сочетании зеркалирования и дуплекса и чередовании с четностью. Основная цель – построение массива значительно повышающего надежность. Массив 1+5 продолжает работать при отказе 3 накопителей, а 5+1 – даже при по-

тере 5 из 8 накопителей. Основные недостатки – 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, который не нашел особого применения. Так как достаточно эффективным является RAID 0+5, RAID 6+0 особого 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. 2.16).

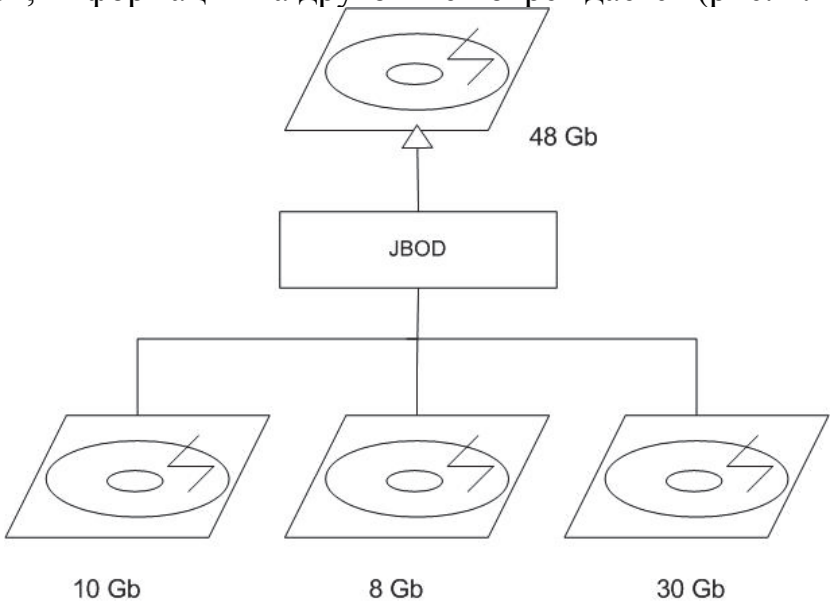

Рис. 2.16. Принципиальная схема RAID JBOD

MATRIX RAID. Эта технология реализуется фирмой Intel, начиная с чипсетов ICH6R. Данная технология позволяет на некотором количестве дисков организовать один или несколько массивов уровня RAID 1, или 5, или 0. Не повышает ни быстродействия, ни надежно-

сти, но позволяет использовать все доступное пространство жестких дисков. В случае выхода из строя одного из жестких дисков, информация на другом не повреждается.

Выводы

Для успешного выполнения тестовых заданий по данной главе важно знать не только теоретические основы построения RAID-массивов, основные принципы, структуру одиночных RAID-массивов, но и понимать принципы построения составных, уметь их анализировать и сравнивать с точки зрения эффективности использования дискового пространства, надежности и скорости. Также необходимо уметь выбирать оптимальную структуру (схему) RAID-массива в соответствии с поставленными требованиями. Вопросы в тесте по материалу данного раздела будут носить прежде всего практический характер.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите основные технологии, используемые при построении RAID-массивов. Приведите примеры.
2. Опишите RAID 0. Назовите основные достоинства и недостатки.
3. Опишите RAID 1. Назовите основные достоинства и недостатки.
4. Опишите RAID 2. Назовите основные достоинства и недостатки.
5. Опишите RAID 3 и 4. Назовите основные достоинства и недостатки.
6. Опишите RAID 5. Назовите основные достоинства и недостатки.
7. Опишите RAID 0+1 и 1+0. Назовите основные достоинства и недостатки.
8. Нарисуйте схему RAID 1+0 для 10 жестких дисков с максимальной надежностью.
9. Нарисуйте схему RAID 1+0 для 10 жестких дисков с максимальной скоростью записи/чтения.
10. Опишите RAID 3+0 и 0+3. Назовите основные достоинства и недостатки.
11. Опишите RAID 5+0 и 0+5. Назовите основные достоинства инедостатки.
12. Опишите RAID 5+1 и 1+5. Назовите основные достоинства и недостатки.

13. Опишите RAID 1+1+0. Назовите основные достоинства и недостатки.

14. Что такое JBOD RAID? Каково его назначение?

15. Выберите RAID-массив, состоящий из 6 жестких дисков, таким образом, чтобы достигнуть максимальной надежности при условии более 50% использования дискового пространства.

16. Выберите RAID-массив, состоящий из 6 жестких дисков таким образом, чтобы достигнуть максимальной скорости при условии не менее 50% использования дискового пространства.

ТЕМА 3. IP-АДРЕСАЦИЯ И МАРШРУТИЗАЦИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

3.1. Протокол IPv4

3.1.1. 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}. 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 – несуществу-ющий, поскольку 256 выходит за пределы допустимого диапазона: от 0 до 255.

IP-адрес состоит из двух логических частей – ***номера подсети*** (ID подсети) 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 единицы. 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, оставшиеся биты обозначают ID хоста.

3.1.2. 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 3.1. 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 A*. 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,
Subnet mask: 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 по-другому, 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 3.2. 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 = AND 11000000.10101000.01011001.00010000
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 масок подсети, имеющей следующий вид:

11111111.11110111.00000000.00001000 (255.247.0.8),
11111111.01111111.00000000.00000000 (255.127.0.0),
11111111.11110000.10000000.00000000 (255.240.128.0),

так как последовательности единиц и нулей не являются непрерывными на протяжении всех 32 бит.

С помощью масок администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей.

Пример 3.3. Допустим, организации выделена сеть класса *B*: 160.95.0.0 (рис. 3.1).

В такой сети может находиться до 65 534 узлов. Однако организации требуется 3 независимые сети с числом узлов в каждой не более 254. В этой ситуации можно применить деление на подсети с помо-

щью масок. Например, при использовании маски 255.255.255.0 третий байт адреса будет определять номер внутренней подсети, а четвертый байт – номер узла (рис. 3.2).

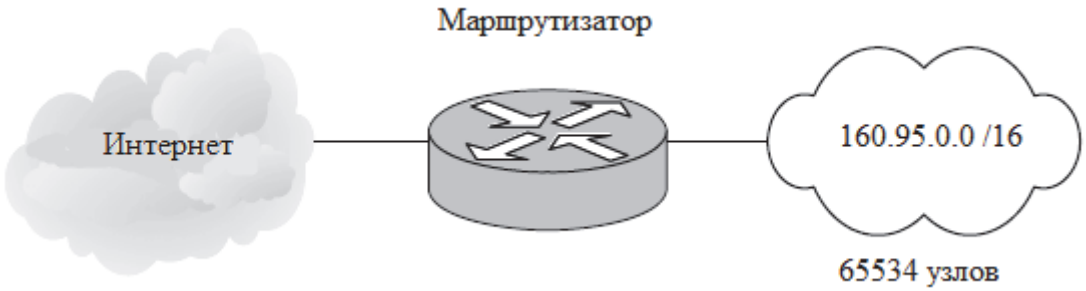

Рис. 3.1. Сеть класса B до деления на подсети

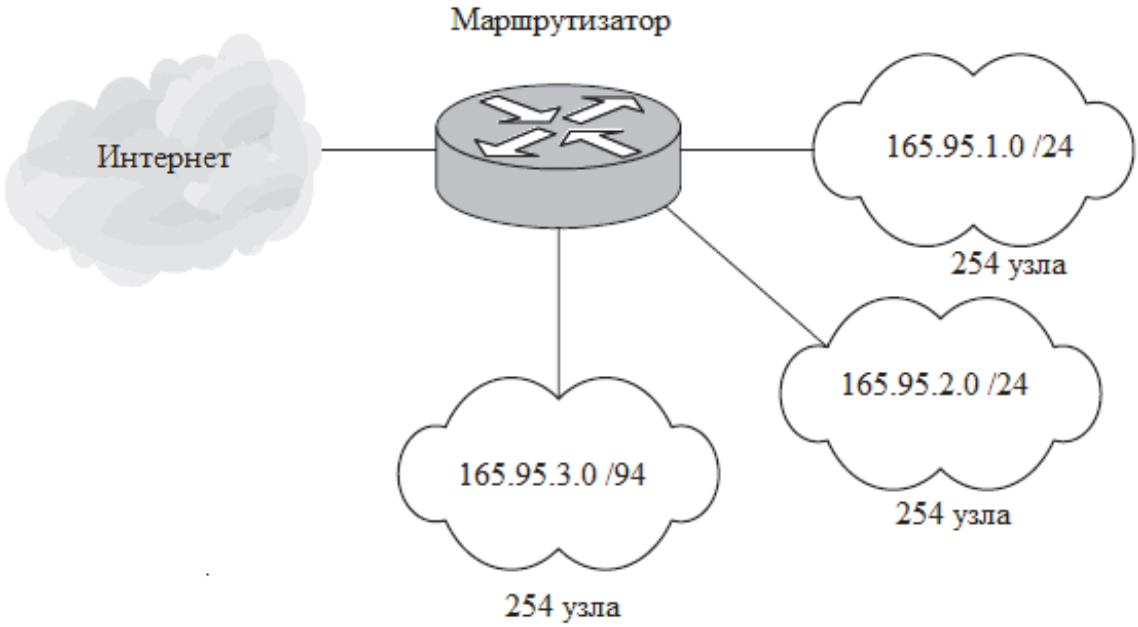

Рис. 3.2. Сеть класса B после деления на подсети

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.

3.1.3. Особые IP-адреса

Некоторые IP-адреса являются особыми, они не должны применяться для идентификации обычных сетей.

1. Если первый октет ID сети начинается с 127, такой адрес считается адресом машины-источника пакета. В этом случае пакет не выходит в сеть, а возвращается на компьютер-отправитель. Такие адреса называются loopback («петля», «замыкание на себя») и используются для проверки функционирования стека TCP/IP без реальной отправки пакета по сети.

2. Если все биты IP-адреса равны нулю, адрес обозначает узел-отправитель и используется в некоторых сообщениях ICMP (например, на начальном этапе получения IP-адреса от DHCP-сервера).

3. Если все биты IP-адреса равны 1, адрес называется ограниченным широковещательным (limited broadcast). Пакеты, направленные по такому адресу, рассылаются всем узлам той подсети, в которой находится отправитель пакета. В данном случае пакет с таким адресом не выходит за пределы своей сети, т. к. маршрутизатор, связывающий сети, его дальше «не пропустит». Такой особый IP-адрес (limited broadcast) может иметь также другой вид – все биты Network ID будут равны 0, а все биты Host ID – равны 1. Пакет с таким адресом также «не выйдет» за пределы сети отправителя, так как нули в идентификаторе сети указывают именно на сеть, где находится отправитель пакета.

4. Если все биты ID хоста равны 1, а биты Network ID идентифицируют определенную сеть (но не ту, где находится отправитель пакета), то адрес называется широковещательным (broadcast); пакеты, имеющие широковещательный адрес, доставляются всем узлам подсети назначения. В случае, если сеть отправителя и сеть назначения совпадают, то адрес принято называть также broadcast, хотя фактически пакет не выйдет за пределы сети отправителя.

5. Если все биты ID хоста равны 0, адрес считается идентификатором подсети (subnet ID).

Наличие особых IP-адресов объясняет, почему из диапазона доступных адресов исключаются два адреса – это случаи, когда все биты ID хоста равны 1 или 0. Например, в сети класса C не 256, а 254 узлов.

Рассмотрим пример адресов, которые будут являться особыми.

Пример 3.4. Пусть IP-адрес, по которому должен быть отправлен пакет, равен 172.16.255.255, а маска – 255.255.0.0. Адрес отправителя 172.17.1.1/16.

Согласно правилам, описанным выше, Network ID в данном случае будет 172.16.0.0 (определенная подсеть, но не та, где находится отправитель пакета). Host ID будет равен 0.0.255.255, т. е. все биты

в двоичной форме равны 1. Это значит: такой адрес считает особым адресом типа broadcast, и пакет будет отправлен всем узлам в сеть с Network ID 172.16.0.0.

Пример 3.5. Пусть IP-адрес, по которому должен быть отправлен пакет, равен 37.255.255.255, а маска – 224.0.0.0.

Если провести расчеты, как это сделано в примере 3.2 (определить Network ID и Host ID), то получится, что идентификатор сети равен 0.0.0.0, что означает подсеть, где находится отправитель пакета, а вот идентификатор хоста – 37.255.255.255, что в двоичной форме будет 00011111.11111111.11111111.11111111, т. е все биты Host ID равны 1 (Network ID – первые 3 бита адреса, Host ID – оставшиеся 29 бит). Это означает: адрес относится к категории limited broadcast, и пакет с таким адресом будет отправлен всем узлам в сети отправителя.

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

3.2. 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 также 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.

3.2.1. 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 мультикаст-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 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).

3. Одиночный адрес может приписываться нескольким физическим интерфейсам, если приложение рассматривает эти несколько интерфейсов как единое целое при представлении его на уровне Интернет.

4. 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.

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

3.2.2. 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 представленных выше адресов:
1080::8:800:200C:417A
ff01::43
::1
::

3. Альтернативная форма записи. В данный форме записи ее младшая часть имеет стандартный вид для IPv4, т. е. представляется 4-мя восьмибитными 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. 3.1).

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 отличны от них.

Таблица 3.1

Префиксы IPv6-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
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

3.2.3. Unicast-адреса

Существует несколько форм представления unicast-адресов в IPv6:

• глобальные unicast-адреса провайдера;
• географические unicast-адреса;
• IPX иерархический адрес;
• 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. 3.3):

N бит	128 бит
Префикс субсети	Интерфейс ID

Рис. 3.3. Общий вид unicast-адресов

Для локальной системы сети, где применимы MAC-адреса, используются IP-адреса следующего типа (рис. 3.4):

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

Рис. 3.4. 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 имеют сложную иерархическую структуру). В примере, представленном на рис. 3.5, 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. 3.5. Unicast-адреса для сложноструктурированных сетей

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

Также есть так называемый **не специфицированный адрес**, который состоит из всех нулей (0:0:0:0:0:0:0:0). Он не может присваи-

ваться какому-либо узлу. Как правило, он 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. Используемый 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 используется IPv6 unicast-адреса, которые в младших 32 битах содержит IPv4-адреса. Первые из них называются **IPv4-compatible IPv6-address** (рис. 3.6).

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

Рис. 3.6. 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. 3.7.

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

Рис. 3.7. Unicast-адрес с вложенным IPv4-адресом

Глобальные unicast-адреса (провайдеров)

Данный адрес имеет следующий формат (рис. 3.8):

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

Рис. 3.8. Глобальные unicast-адреса

ID регистора определяет организацию регистра, который задает провайдерскую часть адреса. Использующийся термин «префикс регистрации» относится к старшей части адреса, включая поле ID регистора.

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, включая ID клиента.

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

Локальные unicast-адреса

Существует 2 типа таких адресов, первые – локальные адреса сети, вторые – локальные адреса каналов (предназначены для работы с каналом, а сети – с определенной сетью) (рис. 3.9).

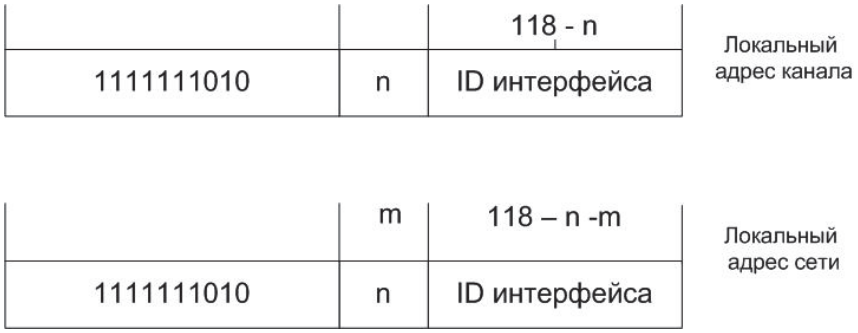

Рис. 3.9. Локальные unicast-адреса

3.2.4. 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 следующий (рис.3.10).

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.

Рис. 3.10. Структура anycast-адреса

3.2.5. Multicast -адреса

Являясь идентификатором ID группы узлов, узел может принадлежать любому числу multicast-групп. Общий вид multicast-адреса представлен на рис. 3.11.

Рис. 3.11. Структура multicast-адреса

Старшие 3 бита флага зарезервированы и пока используются в виде нулей. 4 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. 3.2).

Таблица 3.2

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

Окончание табл. 3.2

Значение поля scope	Зона действия
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, например:

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 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 определенных временным локальным multicast-адресом 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:

- 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.

- Адреса всех маршрутизаторов:

FF01::2;
FF02::2.

Идентифицирует группу всех IPv6-маршрутизаторов в пределах области 1 – локальные узлы, области 2 – локальные связи.

- DHCP Server/ relay agent:

FF02::C.

Идентифицирует группу всех IPv6 DHCP серверов и ретранслирующих агентов в пределах области 2 – локальный канал.

- Адрес активного узла:

FF02::1:xxxx:xxxx.

3.2.6. Необходимые адреса узлов

Хост должен распознавать следующие адреса, 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.

3.3. Понятие маршрутизации. Таблицы маршрутизации

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 основываются на TCP/IP, основное внимание уделяется 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 – *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.

Компьютер также может выполнять функции маршрутизатора 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. 3.12).

Согласно примеру, представленному на рис. 3.12 (результат использования команды Route Print), можно увидеть, что таблицы разделены на 5 колонок. Первой идет колонка сетей. В ней представлены все сетевые сегменты, к которым подключен маршрутизатор.

Колонка Netmask показывает маску подсети, но не сетевого интерфейса, к которому подключен сегмент, а самого сегмента. Это позволяет маршрутизатору определить класс адреса для сети места назначения.

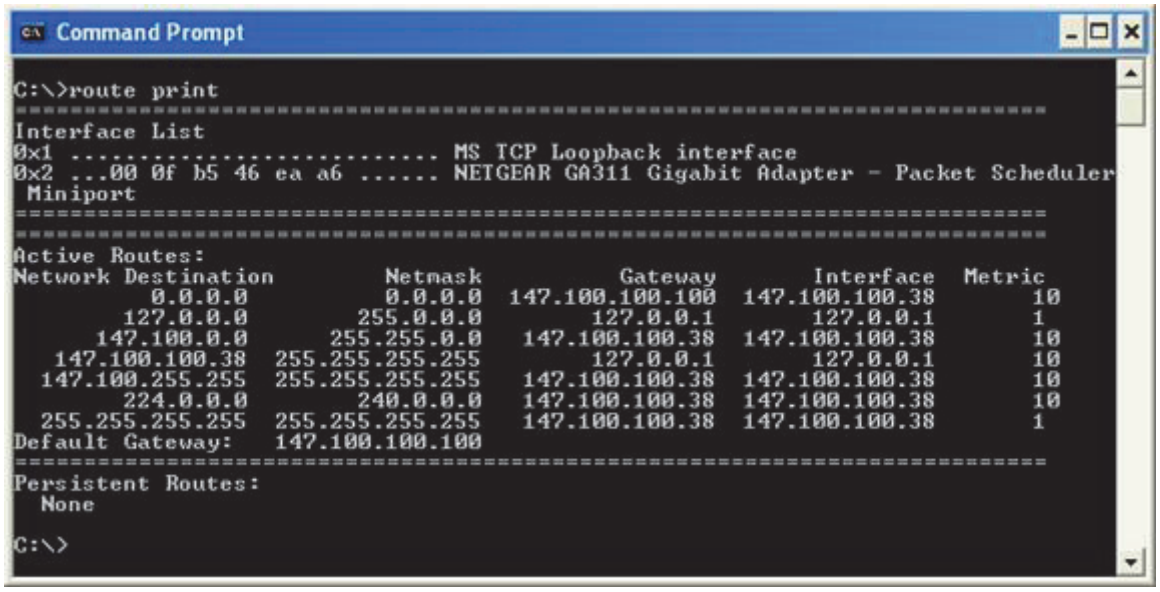

Рис. 3.12. Пример таблицы маршрутизации

Третьей является колонка шлюза. После того как маршрутизатор определил сеть назначения, в которую необходимо отправить пакет, он сверяется со списком шлюза. Данный список «говорит» маршрутизатору, через какой IP-адрес необходимо отправлять пакет в сеть назначения.

Колонка интерфейса предоставляет информацию о сетевом адаптере, подключенном к сети назначения. Точнее будет сказать, что данная колонка предоставляет информацию о IP-адресе сетевого адаптера, который соединяет маршрутизатор с сетью назначения.

Последней идет метрическая колонка. Маршрутизация работает по следующему принципу: существует несколько маршрутов отправки пакетов, при этом имеет смысл отправить его по самому короткому пути. Именно в таких ситуациях и нужны метрики. Windows не задействует метрики, пока есть только один маршрут достижения места назначения. В противном случае Windows проверяет метрики для определения кратчайшего пути.

Существует множество других вариантов использования команды ROUTE. Ее синтаксис следующий:

route [-*f*] [-*p*] [command [destination] []

Переключатель *-f* является необязательным. Он указывает Windows на необходимость очистить таблицы маршрутизации от пунктов шлюза. Если данный переключатель используется совместно с другими командами, то пункты шлюза будут удалены перед выполнением других инструкций, содержащихся в команде.

Переключатель *-p* делает определенный маршрут постоянным. Обычно при перезагрузке сервера любые определенные через команду ROUTE маршруты удаляются. Переключатель *-p* указывает на необходимость сохранять данный маршрут даже при перезагрузке системы.

Командная часть в синтаксисе ROUTE достаточно проста. Она может состоять из 4 вариантов: PRINT, ADD, DELETE и CHANGE. Пример команды ROUTE PRINT приведен выше (рис. 3.12), но и у нее могут быть варианты. Например, можно использовать специальные символы в команде. Если нужно напечатать маршруты для подсети 192.x.x.x, можно воспользоваться командой ROUTE PRINT 192*.

Команда ROUTE DELETE работает так же, как и ROUTE Print. Просто введится ROUTE DELETE, а следом место назначения или шлюз, который необходимо удалить из таблицы маршрутизации. Например, при желании удалить шлюз 192.0.0.0 введите ROUTE DELETE 192.0.0.0.

Все выше сказанное касается и команд ROUTE CHANGE и ROUTE ADD. При введении данной команды следует определить место назначения, маску подсети и шлюз. Также можно указать метрики и интерфейс. Далее рассмотрим примеры использования команды route.

1. Чтобы добавить маршрут по умолчанию с адресом стандартного шлюза 192.168.12.1, введите команду:

route add 0.0.0.0 mask 0.0.0.0 192.168.12.1.

2. Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1.

3. Чтобы добавить постоянный маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0 и следующим адресом перехода 10.27.0.1, введите команду:

route -p add 10.41.0.0 mask 255.255.0.0 10.27.0.1.

4. Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0, следующим адресом перехода 10.27.0.1 и метрикой стоимости 7, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 metric 7.

5. Чтобы добавить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0, следующим адресом перехода 10.27.0.1 и использованием индекса интерфейса 0x3, введите команду:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 if 0x3.

6. Чтобы удалить маршрут к конечной точке 10.41.0.0 с маской подсети 255.255.0.0, введите команду:

route delete 10.41.0.0 mask 255.255.0.0.

7. Чтобы удалить все маршруты из таблицы IP-маршрутизации, которые начинаются с 10., введите команду:

route delete 10.*.

8. Чтобы изменить следующий адрес перехода для маршрута с конечной точкой 10.41.0.0 и маской подсети 255.255.0.0 с 10.27.0.1 на 10.27.0.25, введите команду:

route change 10.41.0.0 mask 255.255.0.0 10.27.0.25 .

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

3.4. Алгоритмы маршрутизации

Маршрутизаторы, а также компьютеры Windows Server, сконфигурированные как маршрутизаторы, обычно используют алгоритмы статической или динамической маршрутизации, где поддерживается 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 в реальном масштабе времени к изменению состояний путем взаимодействия с другими маршрутизаторами. Когда маршрутизатор получает уведомление, что в сети произошло какое-либо изменение, он перерассчитывает 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 используют 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 периодиче-

ские 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.

Выводы

В данном разделе рассмотрены вопросы, связанные с решением задач сетевой адресации в распределенных информационных системах, описаны как правила использования IP-адресов 4 и 6 версий, так и приведены примеры, затрагивающие определение типа адреса, идентификатора сети, идентификатора узла. Рассмотрен пример структуризации сети в рамках выделенного адресного пространства, представлен пример таблицы маршрутизации и правила работы с ней (добавление, изменение, удаление маршрутов). Вопросы в тесте по материалу данного раздела будут носить как теоретический, так и практический характер.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каково назначение IP-адреса?
2. Какова структура IPv4-адреса?
3. Опишите понятия NETWORK ID и HOST ID в IPv4.
4. Использование масок для определения NETWORK ID и HOST ID.
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.

17. Поясните каждый маршрут, представленный на рисунке таблицы маршрутизации.

18. Опишите алгоритм статической маршрутизации, ее достоинства и недостатки.

19. Опишите алгоритм авто-статической маршрутизации, ее достоинства и недостатки.

20. Опишите алгоритм динамической маршрутизации, ее достоинства и недостатки.

21. Опишите алгоритм маршрутизации с коммутируемым соединением по требованию, ее достоинства и недостатки.

22. Какие вы знаете протоколы маршрутизации?

ТЕМА 4. РАСПРЕДЕЛЕНИЕ IP-АДРЕСОВ. ПРОТОКОЛ DHCP

4.1. 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, – это 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 десятка 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 больших 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 неограниченно) и устанавливается администратором.

Выделяют 3 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-сервера 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 диапазон IP-адресов отдельных стандартных областей не пересекались.

Суперобласти. С помощью их 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 для 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 использования два DHCP-сервера в физическом сегменте для управления различными логическими сетями.

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

Во всех TCP/IP сетях 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 в области можно добавлять исключения.

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

4.2. 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;
• LeaseTime – 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 этой группы отдельных параметров. Отнести клиента к определенному классу можно, применив утилиту IPconfig с ключом /setclassid.

4.3. Принцип работы DHCP

Процесс функционирования служб DHCP заключается в обмене сообщениями между сервером и клиентом. Список используемых сообщений представлен в табл. 4.1.

Таблица 4.1

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

Окончание табл. 4.1

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. 4.1. 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. 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 (например, 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.

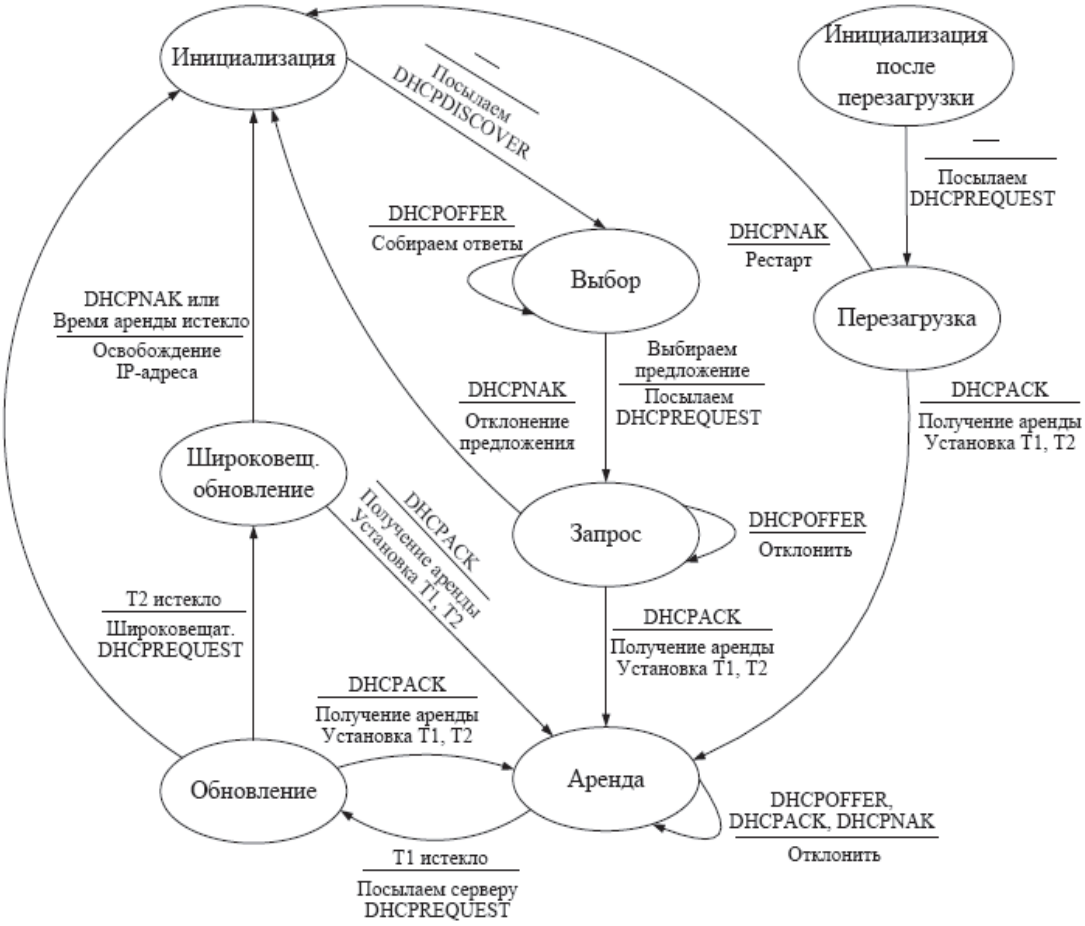

Рис. 4.1. Диаграмма переходов, иллюстрирующая принципы работы протокола DHCP

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 случае 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».

4.4. Адреса для динамической конфигурации

При настройке областей действия перед администратором встает вопрос: 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.

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.

4.5. 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 супербластей (рис. 4.2).

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 % использования адресов;

• доступно – количество и % доступных адресов.

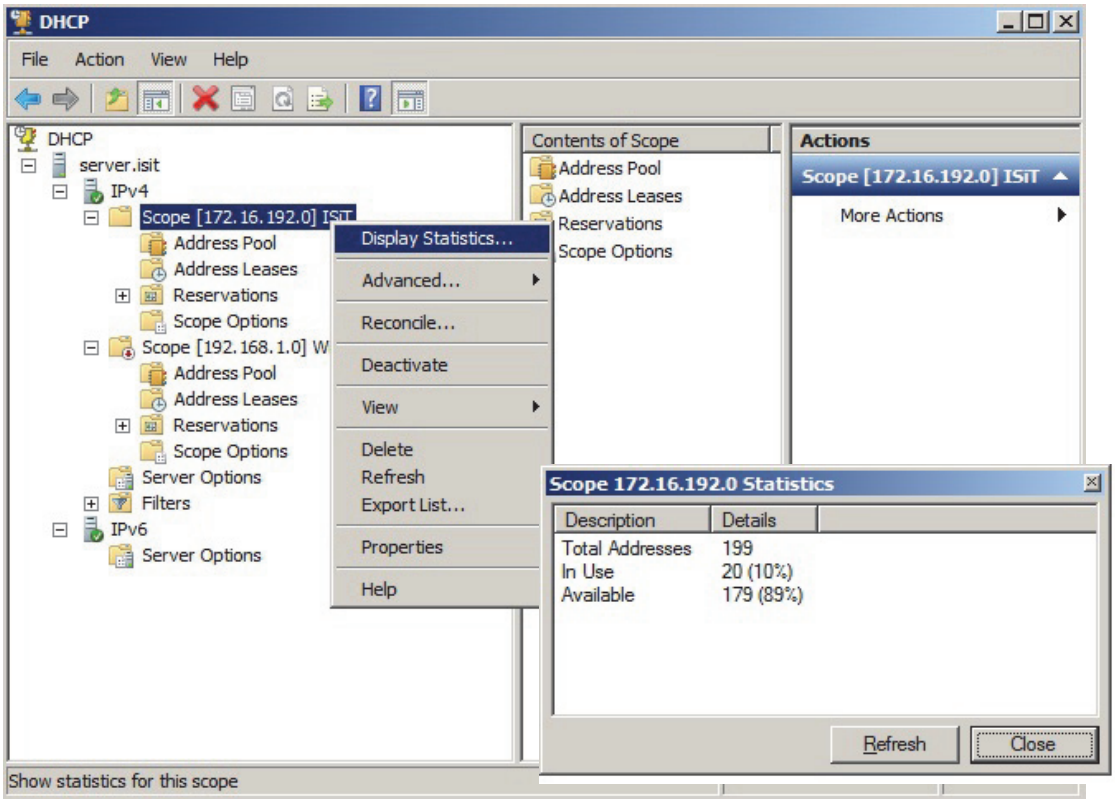

Рис. 4.2. Пример просмотра статистики DHCP-сервера

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 вы можете выяснить точную причину сбоя в файле журнала. Далее рассмотрим 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;

15 – в аренде отказано;

20 – клиенту выделен BOOTP-адрес.

Имеются также отдельные коды для событий, связанных с динамическим обновлением DNS:

30 – запрос динамического обновления DNS;

31 – сбой динамического обновления DNS;

32 – успех динамического обновления DNS.

Также есть отдельные коды (с 50 по 64), связанные с событиями авторизации сервера (представлены в табл. 4.2).

Таблица 4.2

Коды событий авторизации в журналах DHCP-сервера

Код события	Описание
50	Недостижимый домен DHCP-сервер не может обнаружить соответствующий домен для настроенной в Active Directory установки.
51	Авторизация пользователя завершена успешно DHCP-сервер был авторизован для запуска в сети.
52	Произведено обновление операционной системы до Windows Server 2003 DHCP-сервер был обновлен до Windows Server 2003 и, следовательно, возможность выявления неавторизованного DHCP-сервера (используется для определения, был ли авторизован сервер в Active Directory) была отключена.

Окончание табл. 4.2

53	Кэширование данных авторизации DHCP-сервер был авторизован для использования кэшированных ранее сведений. Служба Active Directory не была видна во время запуска сервера в сети.
54	Авторизация пользователя завершена неудачно DHCP-сервер не был авторизован для запуска в сети. После возникновения события сервер, скорее всего, был остановлен.
55	Авторизация (обслуживание) DHCP-сервер был успешно авторизован для запуска в сети.
56	Во время авторизации произошла ошибка. Обслуживание остановлено DHCP-сервер не был авторизован для запуска в сети, и его работа была завершена операционной системой. Необходимо сначала авторизовать сервер в каталоге, а затем снова запустить его. Дополнительные сведения см. в разделе Авторизация сервера DHCP в Active Directory.
57	В домене найден сервер Существует другой DHCP-сервер, авторизованный в том же домене.
58	Сервер не может найти домен DHCP-сервер не может обнаружить указанный домен.
59	Сбой сети Сбой сети не дал серверу возможности определить, был ли он авторизован.
60	Нет контроллеров домена, поддерживающих службы каталога Контроллер домена Windows Server 2003 не обнаружен. Чтобы определить, был ли сервер авторизован, необходим контроллер домена, поддерживающий Active Directory.
61	Найден сервер, принадлежащий к домену служб каталога В сети найден другой DHCP-сервер, принадлежащий к домену Active Directory.
62	Был найден другой сервер В сети был найден другой DHCP-сервер.
63	Перезапуск случайной проверки DHCP-сервер повторно пытается определить, был ли он авторизован для запуска и предоставления обслуживания в сети.
64	Отсутствуют интерфейсы, поддерживающие DHCP Привязки службы DHCP-сервера или сетевые подключения настроены таким образом, что они не позволяют обеспечивать обслуживание.

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. 4.3.

Таблица 4.3

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-клиента

Пример 4.1. Ниже представлен фрагмент журнала, созданного службой DHCP-сервера (рис. 4.3).

Определим суть записей, сделанных в журнал, представленный на рис. 4.3.

В начале журнала приводится коды событий и их значение. Далее уже записывается информации в соответствии с правилами о происходящих событиях с обязательным указанием даты и времени. Так, например, первая строка данного журнала говорит о том, что запущена служба DHCP и соответственно начато ведение журнала (код события 00) в 3 часа 12 минут 36 секунд 07 мая 2014 года (00,05/07/14,03:12:36,Started,,,,). Следующая запись свидетельствует о том, что двумя секундами позже авторизация службы (сервера) прошла успешно (55,05/07/14,03:12:38,Authorized(servicing),,,,). Далее была осуществлена очистка базы данных IP-адресов (24,05/07/14,04:12:46,Database Cleanup Begin,,,,0).

Последняя запись означает, что произошел свой динамического обновления DNS (31,05/07/14,04:12:46, DNS Update Failed,172.16.192.102,НатальяПацей-ПК,-1,).

Рассмотрим другой пример.

Пример 4.2. Пусть строки журнала имеют следующий вид.

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

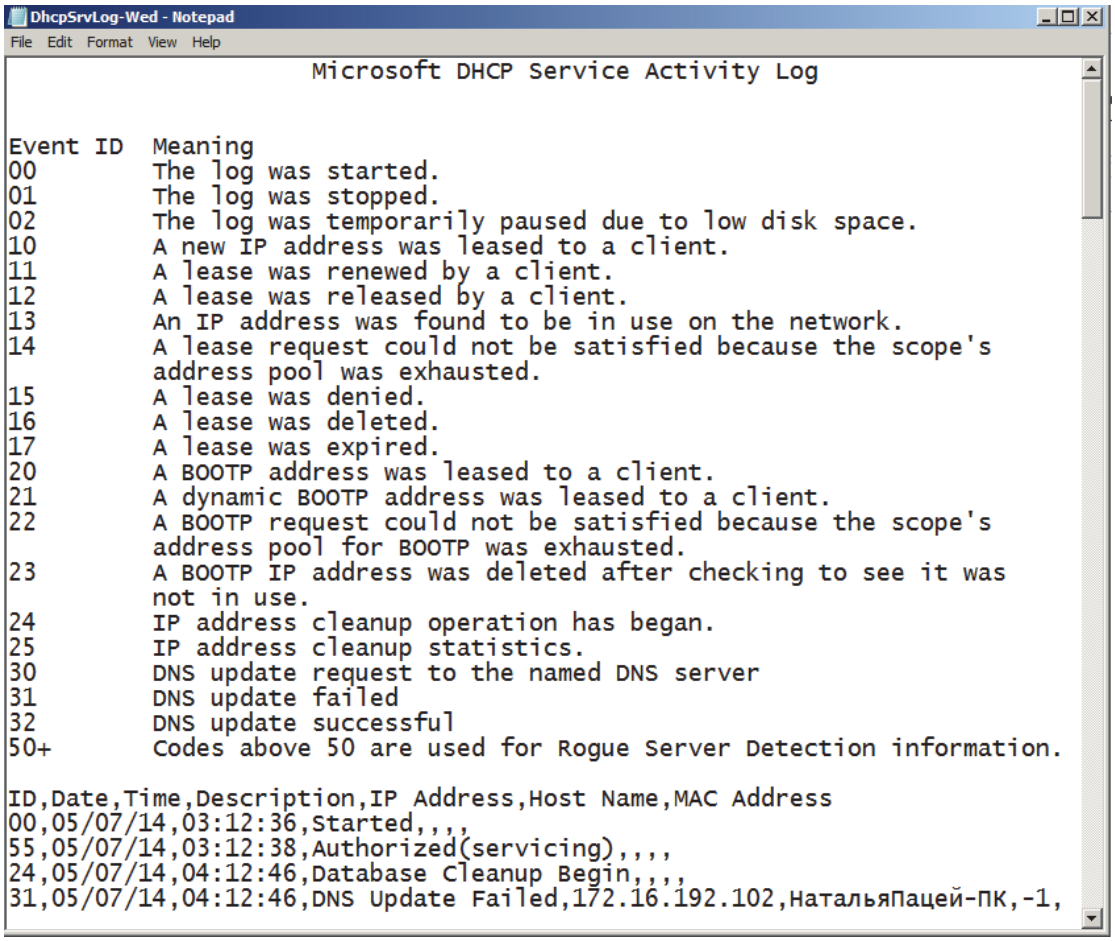

Рис. 4.3. Фрагмент журнала службы DHCP

В данном примере DHCP-сервер был запущен (*00,08/22/02,12:43:06,Запущена,,,)*, но не был авторизован при первоначальном запуске (*60,08/22/02,12:43:21,Нет контроллеров домена, поддерживающих службы каталога,,TEST,*) и впоследствии был остановлен (*63,08/22/02,12:43:28, Перезапуск случайной проверки,,, 01,08/22/02,13:11:13,Остановлена,,,*). После авторизации в Active Directory сервер был запущен (*00,08/22/02,12:43:06,Запущена,,,*) и начал обслуживать DHCP-клиентов (*55,08/22/02,12:43:54, Авторизовано (обслуживается),, TEST,*).

4.6. 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\dhcp и во вложенные папки во временную папку на сервере получателе.

4) Запустите редактор реестра regedit32.exe и раскройте HKEY-LOCALMACHINE\SYSTEM\CurrentControlSet\Services\DHCPServer. 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 перезагрузить 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 папку %systemroot%\system32\dhcp для замены существующей БД DHCP.

10) Запустите regedit32.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.

Выводы

В данном разделе рассмотрены вопросы, связанные с решением задач динамической адресации в информационных системах, описаны основные принципы и правила работы DHCP-сервера, подробно рассмотрен процесс получения клиентом IP-адреса, приведены примеры, касающиеся работы с журналами DHCP, и правила переноса базы данных DHCP-сервера. Вопросы в тесте по материалу данного раздела будут носить как теоретический, так и практический характер.

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

5.1. Система доменных имен

Символьный адрес (имя) не является обязательным, но он упрощает работу пользователей в сети. Существует 2 типа символьных имен, которые используются в IP-сетях:

- DNS-имя (RFC 1034, 1035);
- NetBIOS-имена.

Фрагмент системы доменных имен пространства интернет представлен на рис. 5.1.

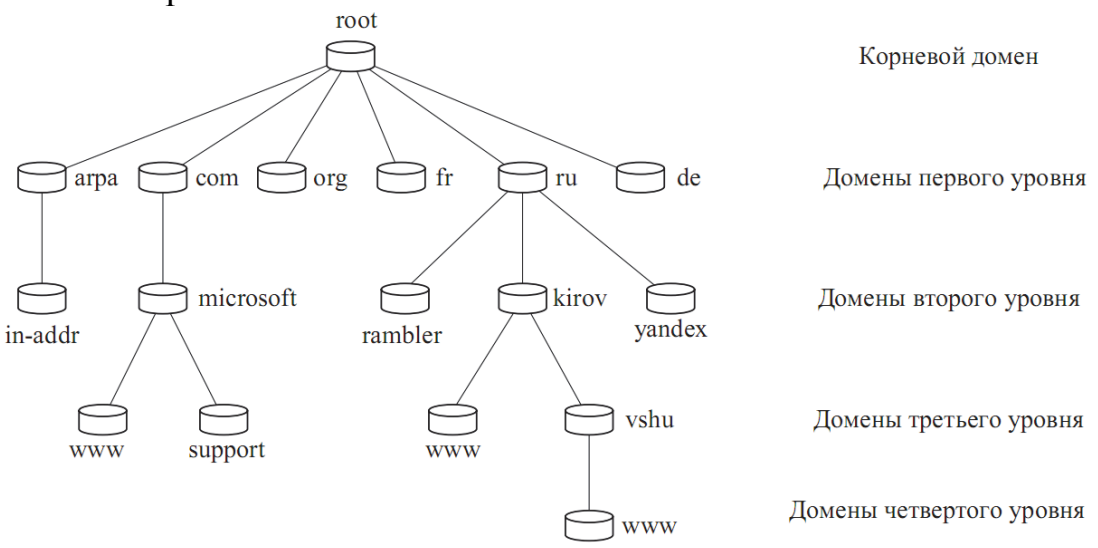

Рис. 5.1. Система доменных имен

Корневой домен как реальный узел не существует – он исполняет роль вершины дерева. Его потомки (поддомены) – это домены 1-ого (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-ого уровня включает в себя только домены 2-ого. Записи об отдельных хостах могут содержаться в доменах, начиная со 2-ого. Созданием и управлением домена 1-ого уровня занимается международная организация ICANN. Домены 2-ого уровня,

находящиеся в географических доменах, распределяются специальными национальными организациями. Управлением доменами 3-его и следующего уровней занимаются владельцы соответствующих доменов 2-ого 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-ого 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, так же как 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 зону (primary zone), остальные серверы 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. 5.2.

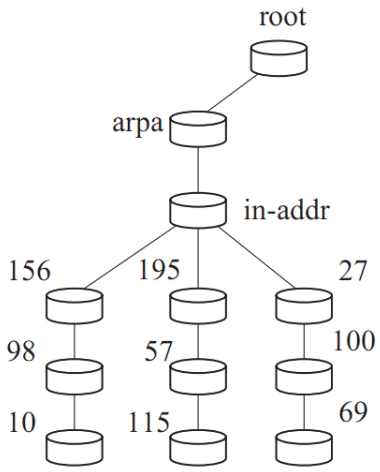

Рис. 5.2. Зона обратного просмотра доменных имен

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

5.2. Процесс разрешения имен

В процессе разрешения участвуют DNS-клиент и DNS-сервер. Системный компонент DNS-клиента, называющийся DNS-распознавателем, отправляет запросы на DNS-серверы. Бывает 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. 5.3 проиллюстрирован процесс разрешения доменного имени с помощью рекурсивного запроса.

Сначала DNS-клиент осуществляет поиск в собственном локальном кэше DNS-имен. Это память для временного хранения ранее разрешенных запросов. В эту же память переносится содержимое файла HOSTS (каталог windows/system32/drivers/etc). Утилита IPconfig с ключом /displaydns отображает 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.

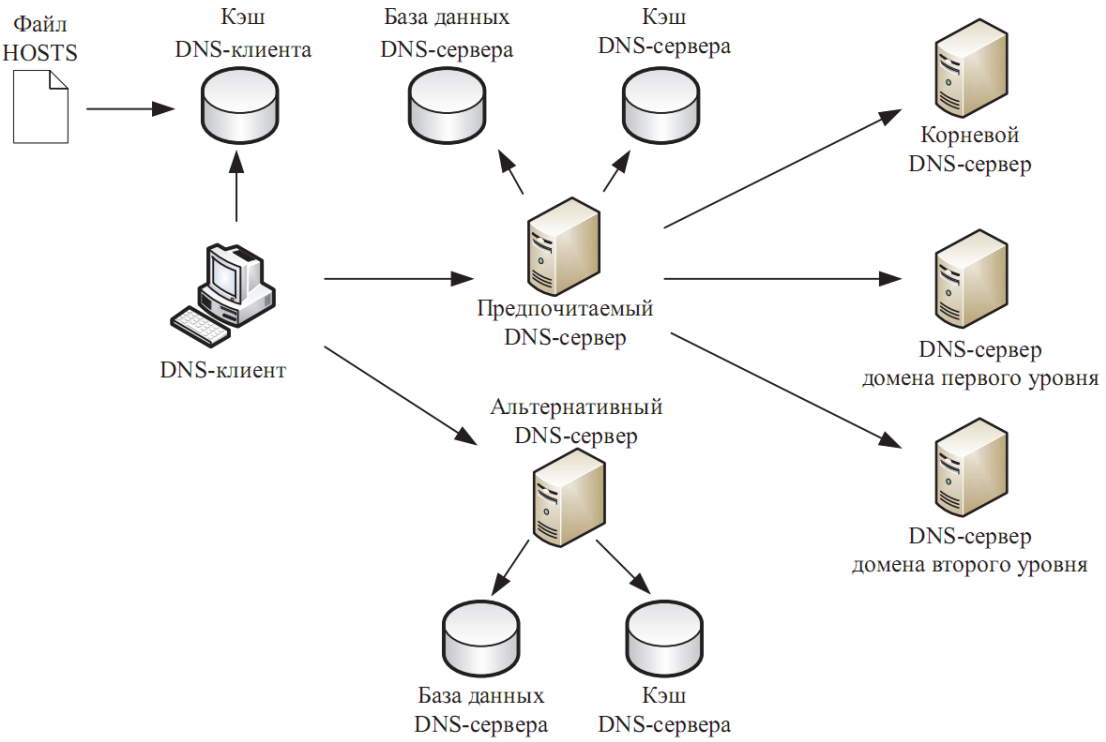

Рис. 5.3. Процесс рекурсивного разрешения имен

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.

5.3. База данных DNS

Единичный информационный объект базы данных DNS называют запись о ресурсах (resource record). 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 БД DNS в виде отдельной записи. Компоненты ресурсных записей представлены в табл. 5.1, а типы ресурсных записей – в табл. 5.2.

Таблица 5.1

Компоненты ресурсных записей

Тип записи	Назначение записи
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 и 32-битные, так и адреса в соответствии с протоколом X25 и номера телефонов, т. е. можно сделать вывод, что поле class указывает, как использовать информацию хранящуюся в данном ресурсе.
TTL	Задает временной интервал продолжительности нахождения данной записи в кэше. При просмотре этой записи в кэше данный интервал уменьшается, а если становится меньше или равен 0, то запись удаляется из кэша.
RDLENSTH	Длина поля данных.
RDATA	Данные ресурса. Максимальная длина поля составляет 65 535 байт, формат представленных данных определяется полями типа и класса.

Таблица 5.2

Типы ресурсных записей

Тип ресурсной записи	Величина	Описание
CNAME	5	Псевдоним ресурса.

Окончание табл. 5.2

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 A 207.46.199.60).
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-ю запись вне данной зоны).
MX	15	Идентификатор почтового ресурса (определяет хост, который служит в данном 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 выше перечисленных 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. 5.3 перечислены некоторые из этих символов, имеющих специальное значение.

Таблица 5.3

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.

5.4. Разрешенные символы в DNS-именах

Изначально имена узлов ограничивались набором символов, указанных в документах RFC 952 и 1123. Это 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 используются расширенный набор символов UTF-8.

5.5. Мониторинги устранения неполадок

Для проверки способности DNS-серверов выполнять разрешение имен используется утилита nslookup. Она может работать в 2-х режимах:

• режим командной строки – обычный режим запуска утилит командной строки. Утилита nslookup 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 также 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.

Для устранения ошибок 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 62 счетчика относятся к производительности DNS, в том числе:

• счетчик общей статистики;

• счетчик TCP и UDP;

• счетчик использования памяти;

• счетчик рекурсивного поиска;

• счетчик зонных передач.

5.6. 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-ого служебного. Если видимых символов меньше 15, то оставшиеся символы заполняются байтом нулей. 16-й символ соответствует службе, работающей на компьютер с данным именем.

Просмотреть список пространства NetBios можно с помощью команды *nbtstat -n*. Рассмотрим пример на рис. 5.4. На рисунке изображен вывод команды «*nbtstat -n*» на сервере *dc1.world.ru*, являющийся списком NetBIOS-имен, сгенерированных данным сервером.

```
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. 5.4. Пример работы утилиты nbtstat

В угловых скобках указан шестнадцатеричный код 16-ого 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 папке, где файл hosts отображает FQDN-имена.

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

В зависимости от типа узла NetBios разрешение имен осуществляется с помощью различной комбинацией 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-адресах с помощью WINS-сервера;
- m-узел (mixed node, смешанный узел) – комбинирует запросы b и p узлов, первоначально узел 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, так как предусматривает использование при необходимости двух вариантов разрешения NetBios имени, при этом основным является метод, создающий минимальный сетевой траффик. Тип узла определяется следующим образом: если в свойствах протокола TCP/IP нет адреса WINS-сервера (рис. 5.6), то данный компьютер считается b-узлом, в противном случае является h-узлом (рис. 5.7). Использование других типов узлов настраивается через реестр Windows.

В больших сетях для распределения нагрузки по регистрации и разрешению имен NetBios имен 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 системой DNS, настраивается репликация.

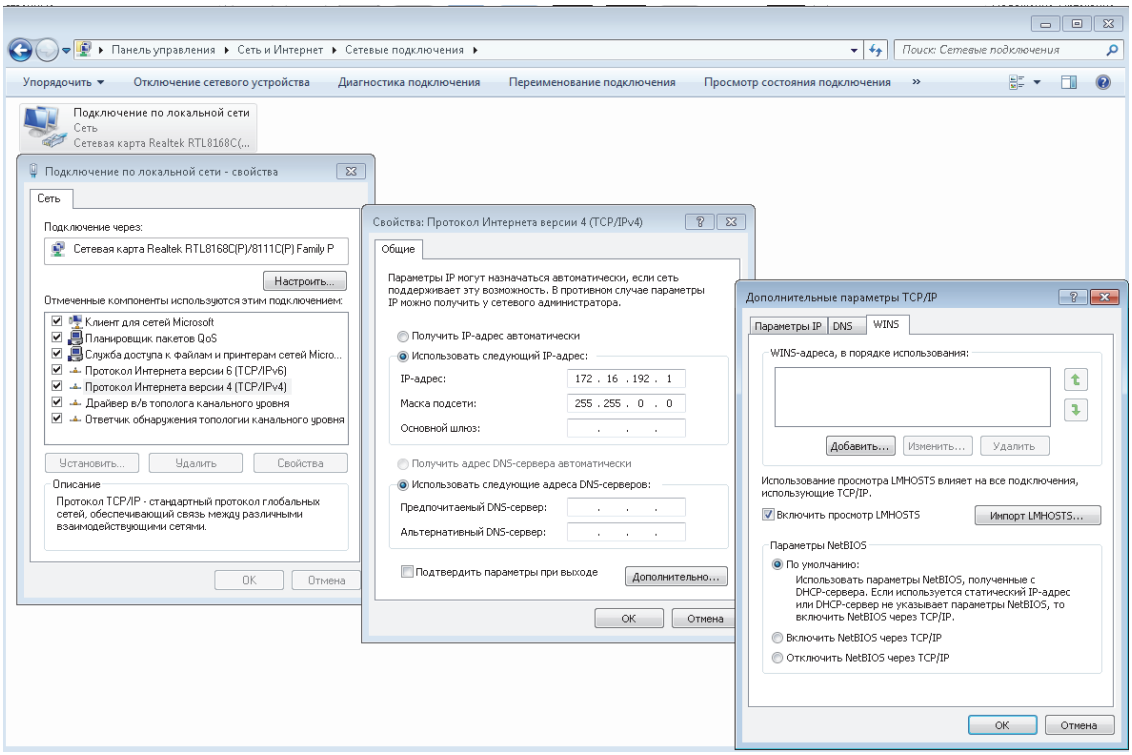

Рис. 5.6. Пример b-узла

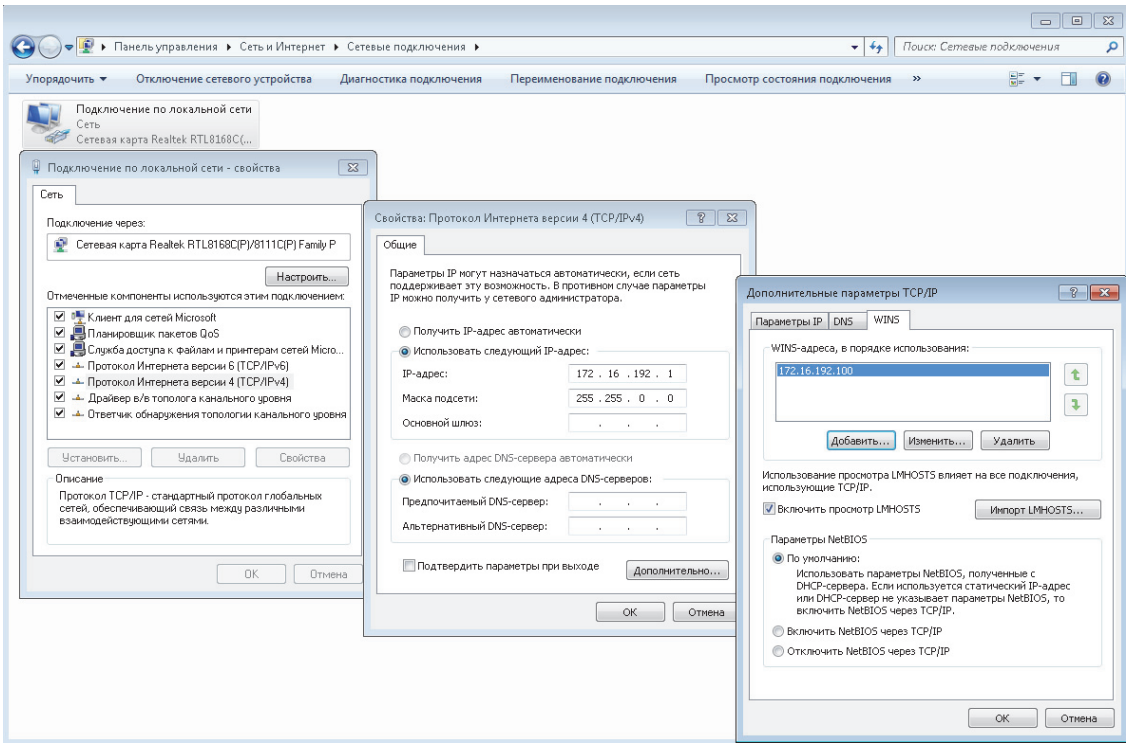

Рис. 5.7. Пример h-узла

Выводы

В данном разделе рассмотрены вопросы, связанные с решением задач символьной адресации (DNS, NetBios) в информационных системах, описаны основные принципы и правила работы DNS-сервера и службы DNS, подробно рассмотрен процесс разрешения клиентом символьного адреса, рассмотрена структура базы данных DNS-сервера. Также приведена структура NetBios имени, способы определения его типа, методы разрешения. Вопросы в тесте по материалу данного раздела будут носить как теоретический, так и практический характер, рассматривая при этом конкретную сетевую ситуацию (проблему), для решения которой требуется понимание всех происходящих процессов.

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 6. БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ. ПРОТОКОЛЫ KERBEROS И IPSECURITY

6.1. 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 Windows Server 2003 используется модифицированная пятая версия протокола – Kerberos v5. 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 – время аутентификации при использовании Kerberos меньше, чем в случае применения NTLM.

Рассмотрим основные термины, используемые при описании протокола Kerberos.

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, 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, использующиеся для взаимной идентификации клиента и сервера. Удостоверения 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 5 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. 6.1.

Для дальнейшего изложения введем обозначения, представленные в табл. 6.1.

Рис. 6.1. Термины, используемые для описания протокола Kerberos

Таблица 6.1

Обозначение, используемые в протоколе Kerberos

Обозначение	Комментарий
A_C	Аутентификатор клиента
A_S	Аутентификатор сервера
K_C	Секретный ключ клиента
KS	Секретный ключ сервера
{X}K	Сообщение X, зашифрованное ключом K
$\{A_C\}K_C$	Аутентификатор клиента, зашифрованный секретным ключом клиента
$K_{A,B}$	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

6.2. 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. 6.2):

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

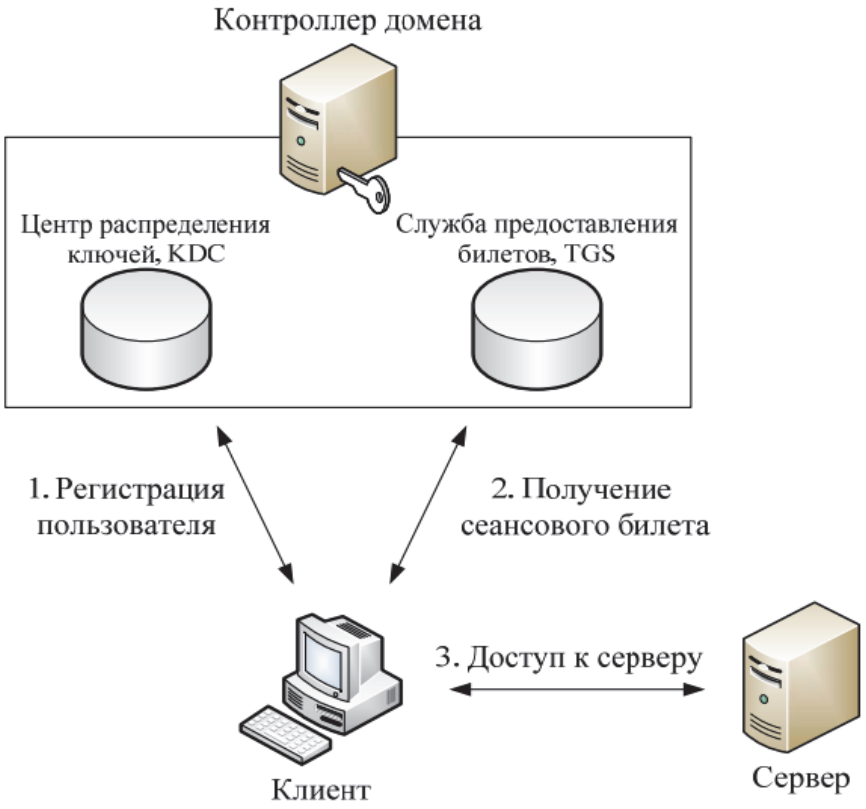

Рис. 6.2. Основные этапы аутентификации

6.2.1. Этап регистрации клиента

При входе в систему под управлением Windows Server пользователь вводит имя своей учетной записи, пароль и указывает домен (рис. 6.3).

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. 6.3). Аутентификатор содержит информацию об имени клиента N и времени отправки аутентификатора t.

Используя свою копию ключа K_C, центр распределения ключей пытается расшифровать полученное сообщение. В случае успеха вы-

числяется разница между временем создания аутентификатора и временем его получения. Если разница не превышает пяти минут, то клиент считается аутентифицированным и ему высылается следующая информация:

- $\{K_{CTGS}\}K_C$ – 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.

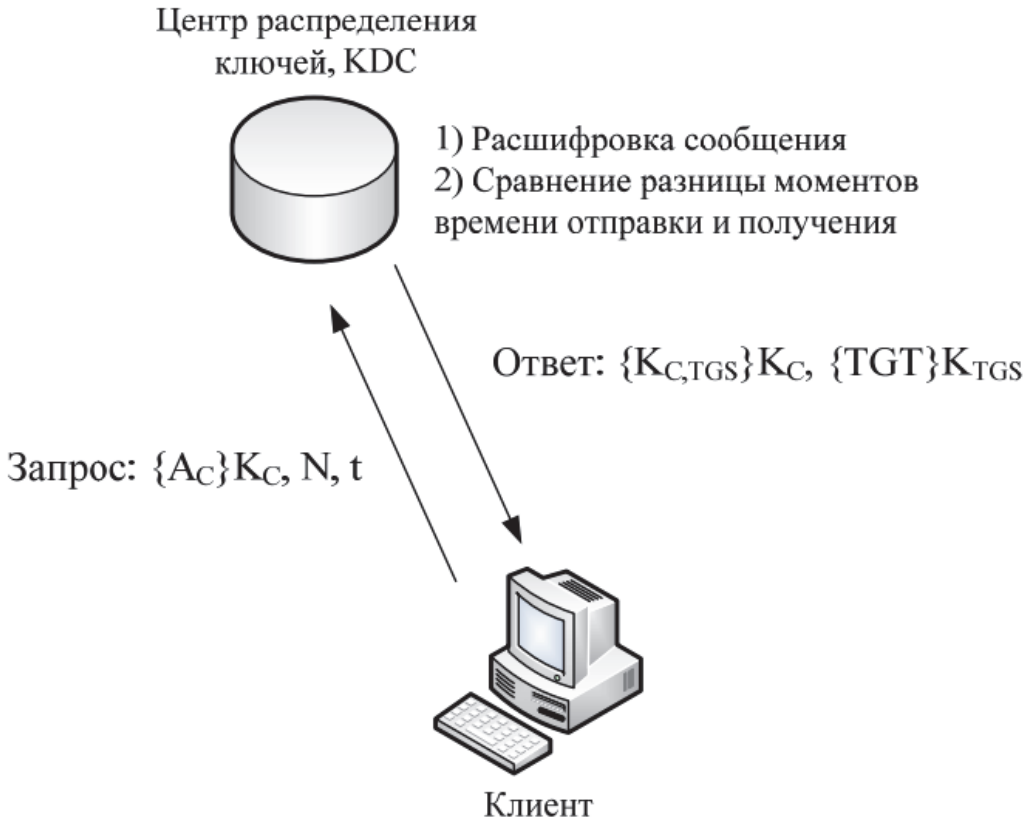

Рис. 6.3. Этап регистрации клиента

Сеансовый ключ K_{CTGS} клиент в состоянии расшифровать, используя свой ключ KC, 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 у службы TGS 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.

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 минут.

6.2.2. 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 6.4):

- $\{A_C\}K_{CTGS}$ – 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.

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.

На запрос клиента служба TGS в случае успешной аутентификации отвечает следующей информацией:

- $\{K_{CS}, t\}\ K_{C,TGS}$ – сеансовый ключ K_{CS} 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 серверу, 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 серве-

ром) клиент будет получать новые сеансовые ключи. Клиент может расшифровать сеансовый ключ, так как он зашифрован ключом K_{CTGS}, известным клиенту.

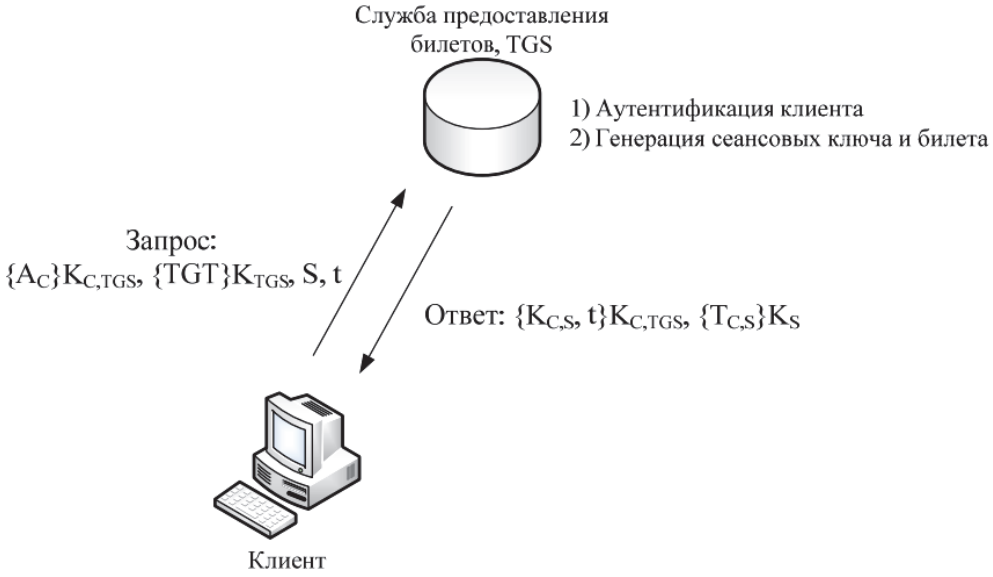

Рис. 6.4. Этап получения сеансового билета

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.

6.2.3. Этап доступа к серверу

Получив сеансовый билет TC,s и сеансовый ключ KC,s, клиент может проходить процедуру аутентификации на требуемом сервере

и в случае успешного прохождения начинать обмен данными. Запрос на аутентификацию включает следующие параметры (рис. 6.5):

• $\{A_C\}K_{CS}$ – аутентификатор A_C, зашифрованный ключом K_{CS}. Содержит информацию об имени клиента, времени отправления, а также ключ KC,s;

• $\{TC,s\}K_s$ – сеансовый билет, зашифрованный ключом K_s.

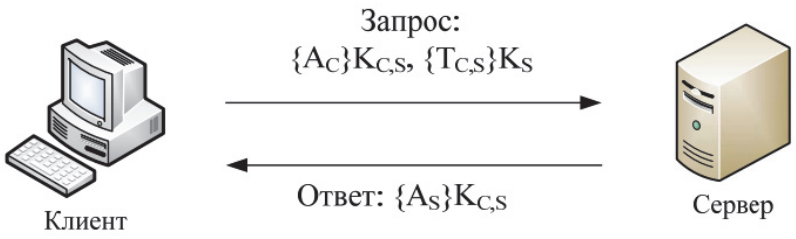

Рис. 6.5. Этап доступа к клиенту

Подлинность клиента удостоверятся следующим образом. В аутентификатор AC клиент 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}. 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.

6.3. 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 применяется и 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.

6.3.1. 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. 6.6 представлена структура протокола IPsec и взаимосвязь основных протоколов, входящих в его состав.

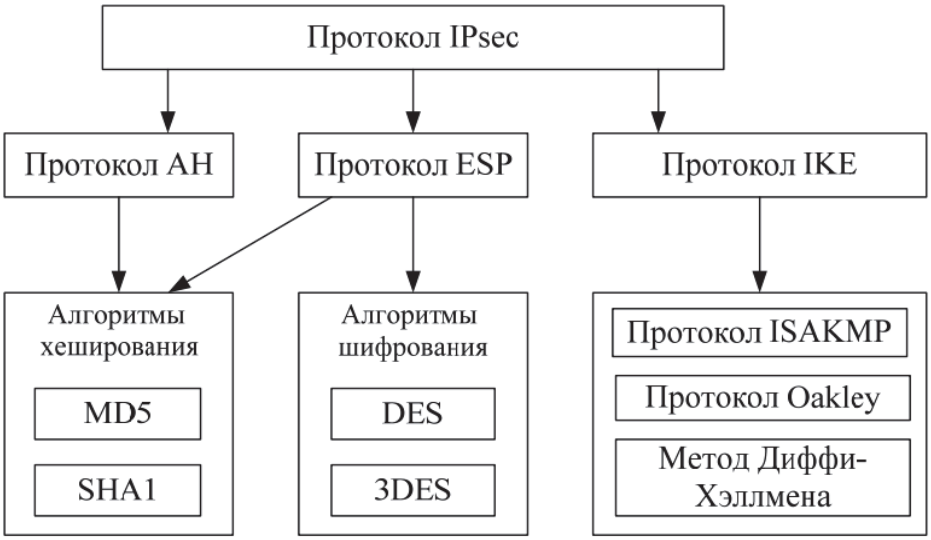

Рис. 6.6. Структура протокола IPsec

6.3.2. 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 (Алгоритм MD5 (Message Digest – алгоритм формирования профиля сообщения) разработан Рональдом Ривестом (Ronald Rivest). См. RFC 2403) или SHA1 (Алгоритм SHA1 (Secure Hash Algorithm – алгоритм безопасного хеша) разработан Национальным институтом стандартов и технологий (NIST, National Institute of Standards and Technology). Является более стойким по сравнению с MD5. Описан в RFC 2404)) к исходным данным. 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).

6.3.3. 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.

Выводы

В данном разделе рассмотрены вопросы, связанные с решением задач обеспечения безопасности функционирования информационных систем (аутентификация, авторизация, шифрование сетевого траффика). Вопросы в тесте по материалу данного раздела будут носить прежде всего теоретический характер и связаны будут с описанием принципов работы протокола Kerberos, а также с рассмотрением структуры протокола IPSecurity.

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. Опишите этап 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?

ЛИТЕРАТУРА

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. – СПБ.: БВХ-санкт-Петербург, 2005. – 320 с.

Учебное издание

Романенко Дмитрий Михайлович

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебно-методическое пособие

Редактор *И. В. Дворникова*
Компьютерная верстка *И. В. Дворникова*
Корректор *И. В. Дворникова*

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.

94