Основы администрирования информационных систем

Учебное пособие

DirectMEDIA

Основы администрирования информационных систем

Учебное пособие

DirectMEDIA

Москва
Берлин
2021

УДК 004.45(075)
ББК 16.3я73
Б72

Авторский коллектив:
Д. О. Бобынцев, А. Л. Марухленко,
Л. О. Марухленко, С. А. Кужелева, Л. А. Лисицын

Рецензенты:
Довбня В. Г. — профессор, доктор технических наук,
главный научный сотрудник НИЦ (г. Курск) ФГУП «18 ЦНИИ» МО РФ
Иванов И. В. — доцент, кандидат технических наук, заведующий кафедрой
информационных технологий, Белгородского государственного
технологического университета им. В. Г. Шухова

Бобынцев, Д. О.

Б72 Основы администрирования информационных систем : учебное пособие / Д. О. Бобынцев [и др.]. — Москва ; Берлин : Директ-Медиа, 2021. — 200 с.

ISBN 978-5-4499-1674-7

В учебном пособии изложен материал, лежащий в основе курса «Администрирование информационных систем», читаемого на кафедре информационных систем и технологий Юго-Западного государственного университета. Пособие посвящено основным вопросам, которыми должен владеть системный администратор. Рассматриваются системные инструменты администрирования информационных систем, компьютерных сетей и баз данных компании Microsoft и прикладное программное обеспечение. Изучение теоретических аспектов позволит сформировать теоретическую базу, необходимую для того, чтобы начать работать системным администратором.

Учебное пособие соответствует федеральным государственным образовательным стандартам высшего образования по направлениям подготовки «Информационные системы и технологии» и «Математическое обеспечение и администрирование информационных систем» и может быть использовано студентами, обучающимися по программе как бакалавриата, так и магистратуры. При необходимости пособие может быть использовано и для других направлений подготовки, на которых могут быть введены соответствующие дисциплины.

Текст приводится в авторской редакции.

УДК 004.45(075)
ББК 16.3я73

ISBN 978-5-4499-1674-7

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Современный мир сложно представить без компьютеров, цифровых технологий и крупных информационных систем. Мы живём в век цифровых машин, которые повсюду. У директора предприятия, в бухгалтерии, в кабинетах начальников отделов, у рядовых сотрудников – у всех стоят компьютеры на рабочих местах. Корпоративная почта, новости, рынок, банк – все эти блага современного трудового коллектива помогают выжить в жестоком мире бизнеса. Но как заставить всё это работать сообща? Как сделать так, чтобы сотрудники не в «ВК» и «Одноклассниках» сидели в рабочее время, а занимались делом? Как уберечь секретную информацию с рабочего компьютера от хакеров? Вот для этого и существует такая профессия, как системный администратор. В народе такого специалиста называют просто «админ» или «сисадмин».

Должность системного администратора присутствует практически во всех фирмах и учреждениях, где используется большое количество компьютеров. Небольшие фирмы часто пользуются услугами внештатных специалистов. Поэтому профессия системного администратора является одной из самых востребованных в сфере информационных систем и технологий, так как в общем смысле охватывает, пожалуй, наибольшую часть умений и навыков, которыми должен обладать специалист данной отрасли. В долгосрочной перспективе ожидается ее совершенствование и дальнейшее развитие.

Многие пользователи путают профессию системного администратора со смежными, называя его компьютерщиком или программистом. По этой причине иногда круг обязанностей сисадмина либо сужается до просто технической поддержки работающей сети, либо неимоверно расширяется до специалиста, способного решить любую возникшую проблему, связанную с компьютерами. По факту же специализация в системном администрировании зависит от конкретного участка работ и от стадии жизненного цикла информационной системы. Могут быть следующие разновидности профессии: администратор веб-сервера в хостинговой компании, администратор баз данных, администратор сети, системный инженер или системный архитектор и др.

Итак, системный администратор, выражаясь простым языком, – это человек, задача которого – обеспечение устойчивой работы компьютерной техники. Все задачи, которые необходимо для этого решить, в зависимости от того, что имеется под рукой на момент начала Вашей работы, и того, какой у Вас будет штат помощников, ложатся именно на Ваши плечи. Если Вам требуется сначала построить корпоративную информационную систему, то Вы же и должны будете определить, какие аппаратные средства потребуются и в каком количестве, поэтому Вы должны разбираться в технических вопросах компьютерных средств, оргтехники и комплектующих,

программного обеспечения. Всё это устанавливать и настраивать тоже будете Вы. А как Вы думали?

При занятии этим делом главное проложить компьютерные коммуникации так, чтобы даже самый изощрённый работник не выдернул ногой штекер и не залил кофе системный блок. Самое сложное в работе системного администратора – объяснить красивой блондинке-секретарше шефа, что не надо прикалывать провод мышки кнопками к столу, даже если он сильно мешает.

Настройка программ и обеспечение их стабильной, надёжной работы будет являться, пожалуй, самым трудоёмким в работе системного администратора. Программное обеспечение может быть абсолютно разным, в зависимости от специфики предприятия. Основной проблемой в работе программного обеспечения является, к сожалению, именно человеческий фактор, так как обычно жалобы сотрудников сисадмину начинаются со слов: «не туда нажала, и всё вдруг куда-то пропало». Синхронизировать работу программ, выставить правильную защиту «от дураков», следить, чтобы у всех всё работало, – это половина работы системного администратора.

Кроме того, администратор должен знать и уметь правильно настроить параметры доступа, чтобы каждый работник мог включить только свой компьютер, свой профиль, запустить только те программы, к которым имеет доступ согласно своему статусу в кампании. Если работа сотрудников связана с Интернетом, то системному администратору предстоит настроить параметры доступа во всемирную паутину так, дабы ни «ВК», ни так любимые офисными работниками «Одноклассники» не были доступны с рабочих компьютеров.

Системный администратор – это весьма весомая фигура в компании, подобно коню на шахматной доске. С одной стороны, не самая важная персона, с другой без него невозможно. Иногда достаточно закончить курсы системного администратора, чтобы стать властителем офисных компьютеров, которые имеют свойство ломаться, причём как у рядового сотрудника, так и у генерального директора.

К достоинствам работы можно отнести почти полную самостоятельность, ведь найти человека на фирме, который будет разбираться в компьютерах на уровне системного администратора и будет проверять вашу работу, практически невозможно. Именно системный администратор является «виртуальным директором» в кампании, поэтому, грамотно настроив оборудование и дав ценные указания «офисному планктону», можно смело проводить трудовые будни, предаваясь социальным сетям или поглощающим современную молодёжь онлайн-играм. Кроме того, решение проблем с техникой начальства и высшего руководства (топ-менеджер, главный бухгалтер, кадровики и т.д.) обеспечит продвижение по служебной лестнице.

Бумажной волокиты у представителей этой профессии значительно меньше, чем у других работников. Как показывает практика, пока началь-

ники отделов и подчинённые усиленно пишут отчёты, сводят счета и подводят итоги, системные администраторы у себя в кабинете крутят солдатиков из витой пары. А самое главное – данная работа обычно достаточно хорошо оплачивается, если Вы покажете себя настоящим властителем компьютерной техники, способным решить любую проблему.

Существуют, конечно, и минусы данной работы. Чем нерадивее пользователи, тем чаще ломаются компьютеры. Что бы ни сломалось у сотрудников, за всё обычно отвечает администратор. Возможности карьерного роста тоже несколько ограничены, если деятельность компании не связана с оказанием услуг в сфере информационных технологий, однако в каждой работе можно найти свои недостатки, а востребованность и хорошая оплата работы системного администратора вполне позволяют данные недостатки преодолеть.

В то же время системный администратор должен иметь хорошее техническое образование. С управлением крупными корпоративными информационными системами связано множество терминов и понятий, которые необходимо просто знать и понимать, что это такое, зачем нужно и как используется. Знаний того, что у компьютера есть системный блок и монитор, и как переустановить операционную систему, Вам будет явно недостаточно, и для решения этой проблемы и предназначено данное пособие. Крайне желательно знание английского языка на уровне выше базового, так как техническое описание многих программных средств прилагается на английском языке. И, наконец, Вы должны быть коммуникабельны, поэтому учитесь находить подход к людям – это поможет зарабатывать премии.

В учебном пособии изложен материал, лежащий в основе курса «Администрирование информационных систем», читаемого на кафедре информационных систем и технологий Юго-Западного государственного университета. Учебное пособие соответствует федеральным государственным образовательным стандартам высшего образования по направлениям подготовки «Информационные системы и технологии» и «Математическое обеспечение и администрирование информационных систем» и может быть использовано студентами, обучающимися по программе как бакалавриата, так и магистратуры. При необходимости пособие может быть использовано и для других направлений подготовки.

ПЕРЕЧЕНЬ ОСНОВНЫХ ИСПОЛЬЗУЕМЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ

Active Directory (AD) – служба каталогов для операционных систем семейства Windows Server, позволяющая выполнять централизованное управление корпоративной сетью.
CMS – система управления контентом.
CN – компонент (класс) отличительного имени в Active Directory, отражающий общее имя объекта.
DC – компонент (класс) отличительного имени в Active Directory, отражающий имя домена, в котором находится объект.
Destination SAT – трансляция внешнего адреса назначения (*Outside local)* в адрес назначения (*Outside global).*
DHCP – служба автоматизации выдачи и учёта IP-адресов.
DHCPDISCOVER – сообщение, рассылаемое клиентом компьютерной сети, адресованное DHCP-серверу, содержащее запрос на получение IP-адреса.
DHCPOFFER – сообщение-ответ DHCP-сервера на сообщение DHCPDISCOVER, содержит предлагаемые клиенту, отправившему DHCPDISCOVER, настройки сети.
DHCPACK – сообщение-ответ DHCP-сервера на сообщение DHCPREQUEST, подтверждающее разрешение клиенту применить предложенные сервером настройки сети.
DHCPREQUEST – сообщение-ответ клиента, запросившего аренду IP-адреса, если запрошенные настройки предложены более чем одним DHCP-сервером, содержит предложенные настройки сети и IP-адрес выбранного клиентом сервера, который их предложил.
DNS – система доменных имён, отвечающая за преобразование доменного имени в IP-адрес и обратно.
DNS-зона – часть пространства имён DNS, за которую отвечает определённый сервер или группа серверов, представляющая собой особую базу данных, которая содержит полномочную информацию, необходимую для разрешения имён.
DNS-сервер – компьютер или программное обеспечение, обслуживание DNS-запросы пользователей компьютерных сетей.
DNS-запрос – запрос на получение IP-адреса узла по доменному имени или доменных имён по IP-адресу.
ECC-память – оперативная память, в которой обеспечивается исправление однократных ошибок и обнаружение двукратных ошибок.
Ethernet – семейство технологий пакетной передачи данных между устройствами в компьютерных сетях.
FTP – протокол передачи файлов в компьютерных сетях.
FTP-сервер – сервер, предназначенный для обмена файлами, работающий по протоколу FTP.

HTTP – протокол прикладного уровня ЭВМОС для передачи данных в виде гипертекстовых страниц.
IP-адрес – сетевой адрес устройства, используемый на сетевом уровне ЭВМОС.
IPv4 – первая широко используемая версия протокола IP, согласно которой IP-адрес представляется четырьмя 8-битными числами, часть из которых показывает адрес подсети, другая часть – адрес узла.
IPv6 – новая версия протокола IP, призванная заменить IPv4, в которой представление IP-адреса расширено до 128 бит и переведено в 16-ричный формат.
IRC – протокол прикладного уровня для обмена сообщениями в режиме реального времени.
Kerberos – сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними, учитывающий, что начальный обмен информацией между клиентом и сервером происходит в незащищённой среде, а передаваемые пакеты могут быть перехвачены и модифицированы.
LDAP – облегчённый протокол прикладного уровня для доступа к службе каталогов.
MAC-адрес – физический адрес устройства, постоянно закреплённый за ним и используемый на канальном уровне ЭМВОС.
NAT – механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.
OU – компонент (класс) отличительного имени в Active Directory, отражающий организационное подразделение, в котором находится объект.
POP3 – протокол входящих сообщений в электронной почте.
RAID – избыточный массив независимых жёстких дисков.
SET – стандарт безопасных транзакций в сети Интернет, применяемый в платёжных системах.
SIP – протокол передачи данных, описывающий способ установки и завершения пользовательского интернет-сеанса, включающего обмен мультимедийным содержимым.
SMTP – протокол исходящих сообщений в электронной почте.
SSL – криптографический протокол защиты данных в компьютерных сетях, в основе которого лежит система цифровых сертификатов.
Source SAT – трансляция внутреннего адреса источника (*Inside local*) в зарегистрированный адрес источника (*Inside global*).
Static Address Translation (SAT) – замена IP-адреса источника или приёмника на некоторый адрес, при этом возможна одновременная замена порта.
TCP-сегмент – единица данных протокола TCP.
TCP-IP – основная сетевая модель передачи данных, основанная на двух важнейших протоколах – TCP и IP.
Telnet – сетевой протокол для реализации текстового терминального интерфейса по сети.

UDP-датаграмма – единица данных протокола UDP.
URI – унифицированный идентификатор ресурса.
URL – единый указатель ресурса, определяющий его местоположение.
Visa – крупнейшая зарубежная платёжная система.
WWW – распределённая система предоставления доступа к связанным между собой документам на компьютерах, подключённых к сети Интернет.
Windows Server – семейство серверных сетевых операционных систем корпорации Microsoft.
Авторизация – процесс установления системой соответствия запрошенных прав доступа к ресурсу и фактических прав пользователя на ресурс и формирования управляемой реакции: разрешить или отвергнуть доступ пользователя к ресурсу.
Авторитативность (авторитетность) DNS-сервера – характеристика, означающая, что сервер хранит запрашиваемую зону.
Адаптивная маршрутизация – основной вид алгоритмов маршрутизации, применяющихся маршрутизаторами в современных сетях со сложной топологией, основан на том, что маршрутизаторы периодически обмениваются специальной топологической информацией об имеющихся в интерсети сетях, а также о связях между маршрутизаторами.
Администратор – должностное лицо, ответственное за работоспособность и надлежащее функционирование всех частей ИВС.
Активное сетевое оборудование – цифровые устройства, обеспечивающие интеграцию вычислительных установок в компьютерную сеть, требующие питания и выполняющие цифровую обработку сигналов.
Аппаратный RAID – массив жёстких дисков, реализуемый в виде множества физических устройств.
Аппаратное обеспечение ИВС – физические устройства ИВС и средства их сопряжения, обеспечивающие объединение устройств в единую информационно-вычислительную систему.
Архитектура клиент-сервер – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.
Архитектура файл-сервер – вычислительная или сетевая архитектура, в которой обработка заданий заказчика услуг концентрируется на рабочих станциях.
Асимметричное шифрование – метод шифрования, при котором используется один открытый и один закрытый ключ, то есть операции шифрования производятся с помощью разных ключей.
Аудит/контроль использования ресурсов – процесс контроля использования ресурсов, включающий возможность ведения журнала попыток доступа к ресурсам.
Аутентификация – процедура проверки подлинности пользователя.

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в конкретной предметной области.
Брандмауэр (firewall) – прикладное программное обеспечение, защищающее вычислительную установку от атак вредоносного программного обеспечения.
Бюджет/учётная запись пользователя (account) – запись в специализированной БД (БД учетных записей), содержащая информацию о пользователе ИВС.
Веб-служба – идентифицируемая веб-адресом программная система со стандартизированными интерфейсами.
Веб-узел – информационный ресурс, дающий возможность предоставлять доступ к информации, организовать работу пользователей с информационной системой.
Виртуализация – программная эмуляция процесса или объекта.
Виртуальная машина – программная абстракция, эмулирующая компьютер, запускаемая на платформе реальных аппаратно-программных систем.
Виртуализация платформ – вид виртуализации, продуктом которого виртуализации являются виртуальные машины.
Виртуализация ресурсов – вид виртуализации, который преследует своей целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей и т.п.
Виртуализация уровня операционной системы – виртуализация физического сервера на уровне операционной системы в целях создания нескольких защищённых виртуализованных серверов на одном физическом.
Виртуализация уровня приложений – вид виртуализации, при котором приложение помещается в контейнер с необходимыми элементами для своей работы: файлами реестра, конфигурационными файлами, пользовательскими и системными объектами.
Витая пара – изолированные медные провода, попарно скрученные и заключённые в гибкую оболочку, используемые в локальных компьютерных сетях.
Внешние глобальные адреса – IP-адреса, назначаемые владельцами узлов, этим узлам для использования во внешней сети.
Внешние локальные адреса – IP-адреса внешних узлов, в том виде как они известны узлам внутренней сети.
Внутренние глобальные адреса – зарегистрированные IP-адреса, назначаемые провайдером службы или выделяемые из регионального регистра Internet (Regional Internet Registries, RIR).
Внутренние локальные адреса – IP-адреса, назначенные хосту во внутренней сети, соответствующие RFC 1918.
Внутренние операции базы данных – действия СУБД, вызываемые в ответ на выполнение запросов логики данных, такие как поиск записи по определенным признакам.

Встроенная учётная запись – учётная запись, создаваемая операционной системой, которая не может быть удалена.
Высокая готовность – способность системы сохранять рабочее состояние без продолжительных периодов простоя, приближая отношение времени пребывания в рабочем состоянии ко времени существования системы к единице.
Вычислительная установка (ВУ) – составная часть информационно-вычислительной системы, представляющая собой компьютер, выполняющий роль сервера или рабочей станции.
Гибридная адаптивная маршрутизация – вид адаптивной маршрутизации, основанный на использовании таблицы, периодически рассылаемой центром, и на анализе длины очереди в самом узле.
Глобальная адаптивная маршрутизация – вид адаптивной маршрутизации, основанный на использовании информации, получаемой от соседних узлов.
Глобальная группа – группа безопасности, которая может содержать только глобальные учётные записи пользователей «своего» домена, и её права доступа могут действовать на ресурсы любого домена в лесу.
Глобальная сеть Интернет – всемирная система объединённых компьютерных сетей для хранения и передачи информации.
Глобальный каталог – перечень всех объектов, которые существуют в лесу Active Directory.
Группа безопасности – вид групп пользователей в AD, предназначенный для назначения прав доступа к ресурсам.
Группа распределения – вид групп пользователей в AD, предназначенный для настройки массовой рассылки сообщений пользователям.
Групповая политика – совокупность параметров рабочего окружения компьютеров и пользовательской рабочей среды.
Двухзвенная схема распределённого приложения – способ построения распределённого приложения, при котором приложение выполняется на двух видах вычислительных установках – сервере и клиенте.
Дерево доменов – набор доменов, которые используют единое связанное пространство имён, и связаны друг с другом отношениями "дочерний"/"родительский".
Динамическая маршрутизация – процесс протокола маршрутизации, определяющий взаимодействие устройства с соседними маршрутизаторами.
Дифференциальное резервное копирование – копирование данных, изменённых со времени последнего полного копирования.
Домен – компьютерная сеть с централизованным управлением и единой для всех компьютеров базой данных служб каталогов. Доменом также называется модель безопасности в такой компьютерной сети.
Домен второго уровня – домен пользовательского уровня иерархии системы DNS, который можно зарегистрировать в компьютерной сети, отделяется от домена первого уровня точкой в полном доменном имени.

Домен первого уровня – домен следующего за корневым доменом уровня иерархии в системе DNS, который, как правило, является региональным или тематическим и не подлежит пользовательской регистрации.
Доменная учётная запись – учётная запись, хранимая в специальных контейнерах AD.
Дополнительная DNS-зона – второстепенный вид зон, который не может храниться в Active Directory, не разрешает создавать ресурсные записи, и используется для повышения отказоустойчивости основной DNS-зоны.
Дополнительное оборудование – оборудование, необходимое для более эффективной и надежной работы основного оборудования ИВС.
Зона-заглушка DNS – вид зон, который имеет только записи NS и SOA и служит для повышения эффективности разрешения имён.
Зона обратного просмотра – DNS-зона, обслуживающая обратные DNS-запросы.
Зона прямого просмотра – DNS-зона, обслуживающая прямые DNS-запросы.
Инкрементное резервное копирование – копирование данных, изменённых со времени последнего полного или инкрементного копирования.
Информационно-вычислительная система (ИВС) – комплекс программных и аппаратных средств для обеспечения автоматизации производства и других сфер жизнедеятельности человека, включающий в качестве составных частей серверное и сетевое оборудование.
Информационное обеспечение КИС – совокупность информационных массивов данных, единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.
Итеративный DNS-запрос – вид DNS-запроса, заключающийся в самостоятельном опросе DNS-сервером авторитативных DNS-серверов в поиске IP-адреса для запрошенного доменного имени.
Кадр – единица данных канального уровня ЭМВОС.
Канало- и сетеобразующее оборудование – оборудование для сопряжения кабельной системы ИВС с ВУ, а также различных частей кабельной системы.
Канальный уровень ЭМВОС – второй после физического уровень ЭМВОС, который обеспечивает обмен информацией между аппаратной частью включенного в сеть компьютера и сетевым программным обеспечением.
Клиентская операционная система – операционная система, которая хранится на дисках рабочей станции (или на дисках сервера), выполняется на процессоре рабочей станции, обеспечивая пользователю ИВС базовый интерфейс (средство взаимодействия) для доступа к ресурсам ИВС.
Ключ шифрования – секретная информация, используемая криптографическим алгоритмом при шифровании и расшифровке сообщений, постановке и проверке цифровой подписи, вычислении кодов аутентичности.

Коаксиальный кабель – изолированная медная жила, помещённая в медную оплётку, покрытую гибкой изоляционной оболочкой, используемый в локальных компьютерных сетях и системах цифрового телевидения.
Коммутатор – устройство соединения компьютеров в сеть на канальном уровне ЭМВОС.
Коммутационное оборудование – концентраторы, мосты и коммутаторы, маршрутизаторы, которые служат для связи частей кабельной системы в единую сетевую инфраструктуру.
Контейнер групповой политики – объект каталога Active Directory, хранящий свойства объекта групповой политики.
Контроллер домена – специальный сервер, которые хранит соответствующую данному домену часть базы данных Active Directory.
Концентратор – устройство соединения компьютеров в сеть на физическом уровне ЭМВОС.
Корневой DNS-сервер – DNS-сервер из пула общеизвестных серверов, обеспечивающий работу корневого домена.
Корневой домен – домен самого верхнего (нулевого) уровня в системе DNS, обозначаемый в имени символом точки, которую допускается не указывать.
Корпоративная информационная система (КИС) – масштабируемая система, предназначенная для комплексной автоматизации всех видов хозяйственной деятельности больших и средних предприятий, в том числе корпораций, состоящих из группы компаний, требующих единого управления.
Лавинная маршрутизация – передача сообщения из узла во всех направлениях, кроме направления, по которому сообщение поступило в узел.
Лес доменов – одно или несколько деревьев, которые разделяют общую *схему*, серверы глобального каталога и конфигурационную информацию, и все домены в нём объединены транзитивными двухсторонними доверительными отношениями.
Логика данных – функциональная часть распределённого приложения, представляющая собой операции с данными, хранящимися в некоторой базе, которые нужно выполнить для реализации прикладной логики.
Логика представления данных – функциональная часть распределённого приложения, которая описывает правила и возможные сценарии взаимодействия пользователя с приложением.
Логическая топология ЛВС – правила взаимодействия сетевых станций при передаче данных.
Локальная адаптивная маршрутизация – вид адаптивной маршрутизации, при котором каждый узел содержит информацию о состоянии линии связи, длины очереди и таблицу маршрутизации.
Локальная база данных (SAM) – локальная база данных учётных записей, которая хранится в реестре операционной системы.

Локальная вычислительная сеть (ЛВС) – компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).
Локальная доменная группа – группа, которая может содержать глобальные группы из любого домена, универсальные группы, глобальные учётные записи пользователей из любого домена леса, и её права доступа могут действовать только на ресурсы "своего" домена.
Локальная учётная запись – учётная запись локальной базы данных.
Маска подсети – битовая маска для определения по IP-адресу адреса подсети и адреса узла этой подсети.
Математическое и программное обеспечение КИС – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.
Маршрутизатор – специализированный компьютер, который пересылает пакеты между различными сегментами сети на основе правил и таблиц маршрутизации.
Маршрутизация – процесс определения маршрута следования данных в сетях связи.
Маршрутизация по предыдущему опыту – способ маршрутизации, при котором каждый пакет имеет счётчик числа пройденных узлов, в каждом узле связи анализируется счетчик, и запоминается тот маршрут, который соответствует минимальному значению счетчика.
Масштабируемость – возможность наращивания мощности ВУ (количество и быстродействие процессоров, объем оперативной и внешней памяти) для пропорционального увеличения скорости и плотности (определённое количество запросов в единицу времени) обработки запросов, а также объёмов хранимой информации.
«МИР» – национальная платёжная система Российской Федерации.
Многопутевая фиксированная маршрутизация – вид фиксированной маршрутизации, при котором может быть установлено несколько возможных путей, и вводится правило выбора пути.
Назначение прав доступа к ресурсу – процедура создания в системе специальной записи, с помощью которой учетной записи пользователя или ее аналогу (например, учетной записи группы пользователей) присваиваются определенные права доступа к ресурсу.
Обратный DNS-запрос – запрос доменного имени для IP-адреса.
Объект групповой политики – общее название набора файлов, каталогов и записей в базе Active Directory (если это не локальный объект), которые хранят настройки и определяют, какие параметры можно изменить с помощью групповых политик.
Однопутевая фиксированная маршрутизация – вид фиксированной маршрутизации, при котором между двумя абонентами устанавливается единственный путь.

Оконечное оборудование – сетевые платы и модемы, которые устанавливаются в ВУ и обеспечивают подключение ВУ к сети.
Оптоволоконный кабель – стеклянная жила (*световод*), заключённая в гибкую оболочку, используемая для передачи данных между территориально удалёнными на большие расстояния компьютерными сетями.
Организационная единица (подразделение) – контейнеры внутри Active Directory, которые создаются для объединения объектов в целях делегирования административных прав и применения групповых политик в домене.
Организационное обеспечение КИС – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.
Основная DNS-зона – главный вид DNS-зон, даёт возможность читать и создавать ресурсные записи и может храниться как в файлах, так и в Active Directory.
Отказоустойчивость (Fault-tolerance) – возможность системы полностью восстанавливать свою работоспособность при аппаратных сбоях.
Отказоустойчивый кластер – кластер (группа серверов), спроектированный в соответствии с методиками обеспечения высокой доступности и гарантирующий минимальное время простоя за счёт аппаратной избыточности.
Пакет – единица данных сетевого уровня ЭМВОС.
Паравиртуализация – вид виртуализации, при котором вместо симуляции аппаратных средств используется специальный программный интерфейс (API) для взаимодействия с гостевой операционной системой.
Пассивное сетевое оборудование – физические элементы компьютерной сети, не требующие питания и не выполняющие цифровой обработки сигналов.
Периферийное оборудование – оборудование, расширяющее функциональные возможности ВУ (прежде всего функциями ввода, вывода), подключаемое к ВУ посредством специализированных *интерфейсов*, либо посредством канало- и сетеобразующего оборудования.
Полная эмуляция (симуляция) – вид виртуализации, при котором виртуальная машина полностью виртуализует все аппаратное обеспечение при сохранении гостевой операционной системы в неизменном виде.
Полное резервное копирование – резервное копирование, затрагивающее всю систему и все файлы.
Полоса пропускания – диапазон частот, в пределах которого амплитудно-частотная характеристика (АЧХ) акустического, радиотехнического, оптического или механического устройства достаточно равномерна для того, чтобы обеспечить передачу сигнала без существенного искажения его формы.

Пользователь (User) – физическое лицо, имеющее доступ к определенным ресурсам ИВС, идентифицируемое бюджетом пользователя (учетной записью).
Почтовая система – программная система, которая служит для взаимодействия пользователей ИВС посредством самой ИВС, аналог обычной почты, реализованный в электронном виде.
Права доступа к ресурсу – степень свободы действий пользователя (просмотр, использование, владение) по отношению к данному ресурсу.
Правовое обеспечение КИС – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.
Представительский уровень ЭМВОС – шестой уровень ЭМВОС, обеспечивающий трансформацию данных в универсальный распознаваемый обоими участниками обмена формат без изменения содержания, а также возможность шифровки и дешифровки данных для обеспечения секретности.
Прикладная логика – функциональная часть распределённого приложения, отражающая набор правил для принятия решений, вычислительные процедуры и операции.
Прикладное программное обеспечение – класс программного обеспечения, который служит для выполнения информационно-вычислительных задач, решаемых обычными пользователями ИВС.
Прикладной уровень ЭМВОС – самый верхний уровень ЭМВОС, обеспечивающий прикладному программному обеспечению доступ к ЭМВОС, взаимодействие сети и пользователя, а также отвечающий за передачу служебной информации, предоставление приложениям информации об ошибках и формирование запросов к представительскому уровню.
Программный RAID – виртуальный массив независимых жёстких дисков.
Прокси-сервер – промежуточный сервер (комплекс программ) в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером (при этом о посредничестве могут как знать, так и не знать обе стороны), позволяющий клиентам как выполнять косвенные запросы (принимая и передавая их через прокси-сервер) к другим сетевым службам, так и получать ответы.
Пропускная способность – максимально возможная скорость передачи данных, которую может обеспечить канал связи.
Простая маршрутизация – способ маршрутизации, не изменяющийся при изменении топологии и состоянии сети передачи данных.
Пространство имён X.500 – иерархическая структура имён, которая идентифицирует уникальный ключ контейнеру службы каталога.
Прямой DNS-запрос – запрос IP-адреса для доменного имени.
Рабочая группа – компьютерная сеть и модель безопасности, характеризующаяся автономным управлением каждого компьютера, входящего в

сеть, и наличием у каждого компьютера своей базы данных учётных записей.

Рабочая станция – вычислительная установка, которая преимущественно используется как индивидуальное рабочее место пользователя ИВС и служит точкой входа в ИВС.

Разрешение имён – процесс преобразования доменного имени в IP-адрес.

Распределённая информационная система – любая информационная система, позволяющая организовать взаимодействие независимых, но связанных между собой ЭВМ.

Распределённая модель вычислений – обработка и хранение данных на двух и более ВУ.

Распределённое ПО – приложение, реализующее распределённую модель вычислений.

Регистрация пользователя в системе – создание администратором ИВС (или другим уполномоченным липом) бюджета пользователя для данного физического липа.

Рекурсивный DNS-запрос – вид DNS-запроса, при котором DNS-сервер клиента возлагает задачу поиска IP-адреса на другой сервер с целью получить готовый ответ на вопрос клиента.

Репликация данных – механизм синхронизации содержимого нескольких копий объекта, например, содержимого базы данных, процесс, под которым понимается копирование данных из одного источника на другой или на множество других и наоборот.

Ресурсная запись DNS – специальная запись DNS-сервера о соответствии имени и служебной информации в системе доменных имён.

Ресурсная запись A – основная запись, определяющая, какой IP-адрес соответствует запрашиваемому доменному имени.

Ресурсная запись CNAME – вспомогательная ресурсная запись, определяющая альтернативное доменное имя, которое можно применять к узлу с заданным доменным именем.

Ресурсная запись MX – запись, содержащая адрес почтового шлюза для домена.

Ресурсная запись NS – запись, содержащая адрес узла, отвечающего за доменную зону.

Ресурсная запись PTR – запись, обратная записи A, связывает IP-адрес сервера с его каноническим именем (доменом).

Ресурсная запись SOA – запись, указывающая, на каком сервере хранится эталонная информация о доменном имени.

Ресурсы ИВС – физические и логические объекты ИВС, имеющие определенную функциональность, доступную для использования.

Роль Cashing-only – роль DNS-сервера, означающая, что он не хранит на себе никаких зон, только хранится кэш запросов.

Роль Conditional forwards - роль DNS-сервера, отличающаяся от Forward-only явным заданием параметра условной пересылки запроса.

Роль Forward-only – роль DNS-сервера, означающая, что он занимается только пересылкой запросов на другие сервера (обычный рекурсивный запрос отключён).
Роль Non-recursive – роль DNS-сервера, означающая, что на сервере хранится DNS-зона, и у него отключена возможность рекурсивного разрешения имени.
Роль сервера – программный комплекс, который обеспечивает выполнение сервером определённой функции.
Световод – закрытое устройство для направленной передачи (канализации) света.
Сеансовый уровень ЭВМОС – пятый уровень ЭМВОС, обеспечивающий синхронизацию обмена информацией между устройствами сети.
Сервер – вычислительная установка, которая служит преимущественно для совместного использования его информационно-вычислительных ресурсов, к которым относятся, прежде всего, центральный процессор или процессоры (например, если это SMP-система), оперативная и внешняя память (прежде всего, жесткие диски).
Серверная операционная система – операционная система, которая хранится на дисках сервера и выполняется на процессоре(-ах) сервера, обслуживая другие информационно-вычислительные задачи (СУБД, почтовая система и т.д.).
Сетевой протокол – набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть компьютерами.
Сетевой уровень ЭМВОС – третий уровень ЭМВОС, полностью обеспечивающий передачу пакета от исходной до целевой системы.
Симметричное шифрование – метод шифрования, основанный на использовании закрытых, секретных ключей, когда и шифрование, и дешифрация производятся с помощью одного и того же ключа.
Система управления базами данных (СУБД) – совокупность языковых и программных средств для создания, ведения и совместного использования базы данных пользователями.
Система управления контентом – информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым.
Системное программное обеспечение – класс программного обеспечения, который служит для выполнения задач по обслуживанию ИВС, прежде всего ее аппаратного обеспечения.
Служба каталогов – комплекс серверных программных средств, обеспечивающих управление доменом и его участниками.
Случайная маршрутизация – передача сообщения из узла в любом случайно выбранном направлении, за исключением направлений, по которым сообщение поступило узел.

Смешанная топология ЛВС – объединение компьютеров в ЛВС с комбинированием базовых топологий сети.
Совместное использование ресурса ИВС – использование ресурса двумя и более пользователями ИВС.
Список управления доступом (Access control list/ACL) – отдельные записи, которые хранят информацию о том, кто обладает правами на ресурс и каковы эти права.
Схема Active Directory – набор определений типов, или классов, объектов в базе данных Active Directory.
Таблица маршрутизации – электронная таблица (файл) или база данных, хранящаяся на маршрутизаторе или сетевом компьютере, которая описывает соответствие между адресами назначения и интерфейсами, через которые следует отправить пакет данных до следующего маршрутизатора.
Техническое обеспечение КИС – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.
Топология ЛВС – общая схема соединения компьютеров в локальную сеть.
Топология "звезда" – подключение компьютеров к сети через единый центральный узел.
Топология "кольцо" – подключение компьютеров к сети посредством кабеля, замкнутого в кольцо.
Топология "общая шина" – линейное соединение компьютеров в сеть одним кабелем.
Транспортный уровень ЭМВОС – четвёртый уровень ЭМВОС, дополняющий функции сетевого уровня и обеспечивающий необходимую степень надёжности передачи информации.
Трёхзвенная схема распределённого приложения – способ построения распределённого приложения, при котором приложение выполняется на трёх видах вычислительных установок – клиенте, сервере приложений и сервере баз данных.
Удалённое администрирование – управление компьютером через компьютерную сеть с другого компьютера.
Универсальная группа – группа безопасности, которая может содержать другие универсальные группы всего леса, глобальные группы всего леса, глобальные учётные записи пользователей из любого домена леса, и её права доступа могут действовать на ресурсы любого домена в лесу.
Управляемость – возможность удаленного управления, сбора сведений о работе подсистем сервера.
Файловые операции – стандартные операции над файлами и файловой системой, которые обычно являются функциями операционной системы.
Физическая топология ЛВС – способ соединения носителей данных в ЛВС.
Физический уровень ЭМВОС – нижний уровень модели, на котором определяются характеристики элементов оборудования сети.

Фиксированная маршрутизация – вид маршрутизации, применяемый в сетях с простой топологией связей и основанный на ручном составлении таблицы маршрутизации администратором сети.

Центр распределения ключей Kerberos – сторонний посредник между клиентом и сервером в протоколе Kerberos, который ручается за подлинность клиента.

Централизованная адаптивная маршрутизация – вид адаптивной маршрутизации, предполагающий центральный узел, который формирует управляющие пакеты, содержащие таблицы маршрутизации, и рассылает их в узлы связи.

Централизованная обработка данных – двухзвенная схема распределённого приложения, при которой компьютер пользователя работает как терминал, выполняющий лишь функции представления данных, тогда как все остальные функции передаются центральному компьютеру.

Цифровой сертификат – выпущенный удостоверяющим центром электронный или печатный документ, подтверждающий принадлежность владельцу открытого ключа или каких-либо атрибутов.

Частичная виртуализация – вид виртуализации, при котором виртуальная машина симулирует несколько экземпляров аппаратного окружения, в частности, пространства адресов.

Частичная эмуляция (нативная виртуализация) – вид виртуализации, при котором виртуальная машина виртуализует лишь необходимое количество аппаратного обеспечения, чтобы она могла быть запущена изолированно.

Шаблон групповой политики – структура папок в системном каталоге жёсткого диска контроллера домена, содержащая ряд параметров политики безопасности, административные шаблоны, файлы сценариев и прочие файлы, связанные с объектами групповой политики.

Шлюз – сетевое устройство, предназначенное для объединения двух сетей, которые обладают различными характеристиками, используют различные протоколы или технологии.

Электронная подпись – реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа, позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа.

Эталонная модель взаимодействия открытых систем (ЭВМОС/OSI) – единая 7-уровневая модель взаимодействия сетевых устройств, определяющая универсальную схему работы мировых компьютерных сетей.

ВВЕДЕНИЕ

Работа системного администратора связана с владением теоретической частью многих аспектов построения и сопровождения информационных систем и технологий. Это и понимание аппаратной части информационной системы, владение наиболее распространёнными операционными системами, как клиентскими, так и серверными, умение устанавливать и настраивать прикладное программное обеспечение. Администратор должен также знать основные методы обеспечения информационной безопасности, владеть технологиями построения компьютерных сетей и управления ими.

Положительно влияет на уровень администратора владение интерфейсом командной строки и операционными системами семейства UNIX. Известно, что данные операционные системы обладают большей защищённостью, чем продукция компании Microsoft, однако операционные системы Windows более просты в освоении, поэтому начинать изучение системного администрирования рекомендуется на примере серверных операционных систем Windows с графическим интерфейсом.

Данное учебное пособие посвящено основным вопросам, которыми должен владеть любой системный администратор:

1. Задачи и обязанности системного администратора.
2. Типовые схемы построения корпоративных информационных систем.
3. Основные положения построения и функционирования вычислительных сетей.
4. Модели безопасности в вычислительных сетях, доменная модель, функционирование DNS и службы DHCP.
5. Удалённое администрирование.
6. Основы администрирования баз данных и веб-служб.
7. Виртуализация.

Изучение вышеописанных вопросов позволит сформировать теоретическую базу, которая понадобится Вам для того, чтобы начать работать системным администратором.

1. ВВЕДЕНИЕ В АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1. Системный администратор, его функции, основные понятия

Деятельность системного администратора в компании состоит в общей организации и поддержании работы корпоративной информационной системы этой компании, обеспечение бесперебойной и корректной работы системы, выполнение всех связанных с этим технических задач [1]. Должность системного администратора относится к общепринятой категории специалистов. В больших компаниях администратор обычно подчиняется руководителю информационного-технического отдела, а в небольших структурах, как правило, все функции технического и информационного обеспечения возложены на одного человека. В период отсутствия специалиста на небольшой срок эти функции могут быть возложены на другого сотрудника по решению непосредственного начальника. В своей работе администратор обязан следовать всем положениям своей должностной инструкции и другим внутренним положениям и инструкциям организации, а также соблюдать законодательные акты и общепринятые российские стандарты и инструкции в работе с техническим оборудованием, персональными данными сотрудников и клиентов компании и т.д.

Специалист, претендующий на эту должность, должен иметь высшее образование. Считается также, что для претендентов на эту должность важно иметь аттестованную квалификационную группу по электробезопасности не ниже 3. Системный администратор должен **знать**:

1) общепринятую российскую законодательную базу: законы «об информации, информационных технологиях и о защите информации», «о персональных данных», «о связи», «о защите прав потребителей» и т.д.;
2) различные технические инструкции и нормативные акты по эксплуатации инженерного оборудования, вычислительного оборудования;
3) правила информационной безопасности, современные способы защиты и хранения информации для предотвращения несанкционированного доступа или какого-либо повреждения информации;
4) положения техники безопасности, охраны труда, противопожарную безопасность;
5) основные нормы и правила, существующие в области обработки и защиты персональных данных, коммерческой тайны и иной конфиденциальной информации;
6) внутренние положения и инструкции компании;
7) технические характеристики, устройство и конструктивные основные особенности, правила эксплуатации оборудования, задействованного в информационных локальных сетях компании, оргтехники;

8) технические характеристики, устройство и конструктивные особенности, правила технической эксплуатации оборудования телефонных станций и иных средств связи, используемых в организациях;
9) основные методы классификации и защитной кодировки передаваемой информации, действующие современные стандарты по обработке и передаче информации в глобальных сетях;
10) правила и порядок оформления различной технической документации;
11) новейшее программное обеспечение, используемое российскими компаниями;

уметь:

1) налаживать локальную сетевую инфраструктуру организации;
2) проводить диагностику и несложный ремонта аппаратного обеспечения компании;
3) вести необходимый учет и оформлять сопутствующую документацию организации по своему профессиональному профилю.

Администратор должен быть технически грамотным специалистом, соблюдать этику делового общения, быть коммуникабельным и внимательным.

Перечень ключевых функций и обязанностей довольно широк и может разделяться между верховным администратором и его подчинёнными и выглядит следующим образом:

1) налаживать и поддерживать работу технического оборудования, задействованного в вычислительных локальных сетях компании и локальных сетях связи;
2) устанавливать, налаживать и поддерживать работу прикладных программ, которые используются или могут быть использованы в организации для обеспечения ее деятельности;
3) налаживать и поддерживать работу серверов, вычислительных локальных сетей компании;
4) работать с персоналом организации в рамках своих профессиональных компетенций - консультировать работников по возникающим вопросам и обучать их необходимым навыкам работы с локальными служебными программами, регистрировать новых пользователей в локальных сетях компании, обеспечивать их доступ к необходимой им по должности служебной информации и базам данных компании;
5) осуществлять своевременную оперативную техническую и программную поддержку локальных пользователей компании;
6) при необходимости обучать отдельных работников организации по вопросам электробезопасности, проводить им вводный инструктаж;

7) принимать меры по соблюдению информационной безопасности баз данных фирмы на ее серверах и служебной цифровой информации компании в целом;
8) осуществлять своевременное необходимое периодическое резервное копирование и архивирование служебной цифровой информации компании;
9) контролировать использование ресурсов цифровых сетей компании и устанавливать необходимые права доступа сотрудников к информации по различным сегментам;
10) своевременно принимать меры по восстановлению работоспособности вышедшего из строя вычислительного и сетевого оборудования, оргтехники, средств связи, задействованного в локальных сетях компании;
11) своевременно и оперативно принимать все необходимые меры по восстановлению нормальной работоспособности программного обеспечения и локальных баз данных, используемых в компании;
12) оперативно выявлять ошибки отдельных пользователей и программного обеспечения, своевременно принимать меры по их исправлению;
13) осуществлять текущий мониторинг локальных сетей компании и предлагать меры развития ее цифровой инфраструктуры, внедрению нового программного обеспечения;
14) обеспечивать локальную сетевую безопасность компании - защиту от внешнего несанкционированного доступа к служебной информации фирмы;
15) осуществлять мониторинг антивирусной защиты сетей компании, используя необходимое стандартное программное обеспечение;
16) готовить и вносить предложения по приобретению и модернизации вычислительного оборудования и оргтехники, нового программного обеспечения;
17) при необходимости контролировать установку, наладку или ремонт оборудования компании специалистами сторонних организаций в рамках своих компетенций;
18) информировать руководство о фактах нарушения предусмотренных правил обращения с оборудованием компании, ее локальных сетей, программного обеспечения, баз данных или иной служебной цифровой информацией;
19) своевременно и качественно вести предусмотренный в компании технический учет вычислительного оборудования и оргтехники, составлять необходимую внутреннюю служебную документацию и отчетность;
20) оперативно выполнять поручения руководства в рамках своих профессиональных компетенций;

21) взаимодействовать с другими подразделениями компании в рамках своих должностных обязанностей и компетенций;
22) соблюдать все необходимые требования нормативных документов по защите служебной конфиденциальной информации компании, персональных данных сотрудников и контрагентов фирмы;
23) соблюдать действующую в компании трудовую и производственную дисциплину, положения трудового договора, внутренние распоряжения и инструкции фирмы;
24) соблюдать все необходимые технические регламенты по работе и обслуживанию технического оборудования, требования по технике безопасности и пожаробезопасности;
25) повышать свою квалификацию самостоятельно, участвовать в дополнительных образовательных мероприятиях.

Системный администратор может запрашивать дополнительную информацию у других сотрудников компании и сторонних специалистов, а также руководства, в рамках своих компетенций, необходимую ему для выполнения своих текущих служебных обязанностей, в том числе получать необходимые для этого документы и материалы. Администратор вправе устанавливать правила пользования корпоративной сетью и вносить свои предложения по устранению существующих недостатков и улучшению работы информационной системы, требовать от руководства обеспечения организационно-технических условий для нормального функционирования корпоративной информационной системы и выполнения своих должностных обязанностей. При этом он не может каким-либо образом разглашать внутреннюю служебную конфиденциальную информацию компании, делать служебную информацию компании каким-либо образом достоянием общественности, отвечать на любые внешние запросы или давать личные комментарии, без согласования с руководством компании.

Системный администратор несёт персональную ответственность:

1) за ненадлежащее исполнение своих должностных обязанностей, предусмотренных своей должностной инструкцией, согласно мерам ответственности, предусмотренным российским законодательством (трудовым, гражданским, административным, уголовным кодексами и т.д.);
2) за совершенные правонарушения, при некорректном выполнении его обязанностей при работе с техническим оборудованием или конфиденциальной информацией;
3) за сохранность и поддержание в рабочем состоянии технического оборудования компании, входящего в зону его профессиональных обязанностей;
4) за причинение материального ущерба компании в соответствии с действующим законодательством;

Для компании очень важна бесперебойная корректная работа оборудования и локальных сетей, системные сбои наносят существенный материальный ущерб организации, поэтому в случаях их возникновения, требуются незамедлительные высококвалифицированные действия системного администратора, и за его бездействие или грубые некорректные действия, предусматривается отдельная дисциплинарная ответственность, налагаются взыскания, штрафы. Системный администратор обязан исполнять все свои текущие обязанности оперативно и быть внимательным при их выполнении, так как несвоевременное исполнение его обязанностей также влечет убытки для компании, за что может быть предусмотрена дисциплинарная ответственность, налагаться взыскания, штрафы. За ненадлежащее выполнение или невыполнение распоряжений и поручений руководства организации или его непосредственного начальника предусмотрена дисциплинарная или иная ответственность, согласно законодательству РФ. За неверное или недостоверное ведение учета, составление отчетности и иной документации возможно привлечение системного администратора к ответственности, согласно, внутренних положений компании и законодательства РФ, в зависимости от конкретного нарушения.

Системный администратор также несет ответственность за сохранность вверенного ему имущества компании, а также программных продуктов принадлежащих или приобретенных фирмой у третьих лиц. За разглашение служебной или конфиденциальной информации без согласия руководства компании, системный администратор несет гражданско-правовую ответственность, предусмотренную российским законодательством. Поэтому любую передачу конфиденциальной служебной информации третьим лицам, или официальные комментарии, системный администратор должен согласовывать с руководителем фирмы.

Рассмотрим *основные понятия*, используемые в работе системного администратора [2].

Информационно-вычислительная система (ИВС) – комплекс программных и аппаратных средств для обеспечения автоматизации производства и других сфер жизнедеятельности человека, включающий в качестве составных частей серверное и сетевое оборудование.

Пользователь ИВС (User) – физическое лицо, имеющее доступ к определенным ресурсам ИВС, идентифицируемое бюджетом пользователя (учетной записью). Администратор ИВС также является пользователем ИВС, обладая, в общем случае, неограниченным доступом ко всем ресурсам ИВС.

Администратор ИВС (Administrator) – должностное лицо, ответственное за работоспособность и надлежащее функционирование всех частей ИВС.

У администратора большой ИВС в подчинении могут находиться администраторы частей и подсистем ИВС – например, администратор локально-вычислительной сети, администратор сетевой ОС, администратор базы данных (БД), а также технический персонал. Администратор подсистемы ИВС

отвечает за работоспособность и надлежащее функционирование вверенных ему компонентов этой подсистемы ИВС.

Бюджет/учетная запись пользователя (Account) – запись в специализированной БД (БД учетных записей), содержащая информацию о пользователе ИВС. Используется для идентификации пользователя в системе, проверки полномочий пользователя и обеспечения доступа пользователя к тем или иным ресурсам системы. Характеризуется атрибутами, например, имя для входа, пароль, профиль в системе, список принадлежности к группам и т.п. Пароль служит для защиты бюджета от несанкционированного использования.

Регистрация пользователя в системе (Registration) – создание администратором ИВС (или другим уполномоченным лицом) бюджета пользователя для данного физического лица.

Аутентификация в системе (Authentication) – процесс установления подлинности пользователя ИВС. Заключается в предъявлении пользователем своего имени для входа и пароля, а также в проверке системой наличия бюджета в БД бюджетов пользователей и соответствия указанного пользователем пароля и пароля, хранящегося в БД. После успешной аутентификации в системе для пользователя на время сеанса работы создается дескриптор безопасности, отражающий его цифровой идентификатор в системе, а также принадлежность группам пользователей, профилям и другим объектам системы безопасности.

Ресурсы ИВС (Resources) – физические и логические объекты ИВС, имеющие определенную функциональность, доступную для использования. Примеры физических ресурсов – сервер ЛВС, каталог совместного использования на сервере, сетевой принтер; логических ресурсов – пользователь, группа пользователей, профиль в системе, очередь на печать и т.д.

Совместное использование ресурса (Resource sharing) – использование ресурса двумя и более пользователями ИВС.

Права доступа к ресурсу (Access rights to the resource) – степень свободы действий пользователя (просмотр, использование, владение) по отношению к данному ресурсу. Имеют специфику применительно к разным ресурсам и подсистемам ИВС (создание, чтение, запись, удаление файлов и каталогов – для файловой службы; создание, печать документов/управление очередью на печать — для службы печати и т.д.). По определению, администратор ИВС имеет полные права на все ресурсы ИВС. Администратор части ИВС - полные права на ресурсы части ИВС. Права доступа на прямую связаны с *ответственностью пользователя,* которую он несет, пользуясь этими правами.

Назначение прав доступа к ресурсу (User's rights assignment) – процедура создания в системе специальной записи, с помощью которой учетной записи пользователя или ее аналогу (например, учетной записи группы пользователей) присваиваются определенные права доступа к ресурсу. Назначение прав доступа в современных информационно-вычислительных системах

осуществляется через списки управления доступом (Access Control List/ACL).

Аудит/Контроль использования ресурсов (Audit) – процесс контроля использования ресурсов, включающий возможность ведения журнала попыток доступа к ресурсам. Журнал аудита ведется на основе данных, поступающих от процедур авторизации.

Список управления доступом (Access Control List /ACL) – в виде отдельных записей хранит информацию о том, кто обладает правами на ресурс и каковы эти права. Например, для одного пользователя в ACL каталога файловой системы могут быть указаны права на чтение, а для другого пользователя – права на чтение и запись.

Авторизация/Проверка прав доступа (Authorization/ Rights verification) – процесс установления системой соответствия запрошенных прав доступа к ресурсу и фактических прав пользователя на ресурс и формирования управляемой реакции: разрешить или отвергнуть доступ пользователя к ресурсу. Например, пользователь выполняет операцию открытия файла на запись (запрашиваемые права), обладая при этом только правом просмотра (фактические права). Система запретит выполнение операции, мотивируя свое поведение недостатком прав у пользователя.

У администратора есть несколько «золотых правил»:

1. Никогда не проводить экспериментов на работающей системе. Если это все-таки необходимо, сделать сначала полную резервную копию данных.

2. Никогда не менять конфигурацию сервера (как аппаратную, так и программную), не сделав предварительно полную резервную копию данных.

3. Всегда документировать свои действия в журнале администратора. Если это возможно – пользоваться встроенными в серверные ОС средствами аудита и журналирования.

4. Если можно переложить часть работы на подчиненного, перекладывать. Но если не уверен, что подчиненный справится с заданием должным образом, делать самостоятельно.

5. Всегда соотносить назначаемые права с мерой ответственности, связанной с теми или иными правами, т.е. пользователь, имеющий больше прав, берет на себя больше ответственности. Администратор должен обладать полными правами на вверенную ему систему.

6. При работе с ресурсами ИВС в качестве пользователя (например, при выполнении таких действий, как редактирование документов, просмотр и отправка почтовых сообщений, разработка ПО и т. п.) использовать учетную запись с обычными правами доступа, а не учетную запись администратора.

7. Регулярно менять пароль учетной записи администратора. Но не полагаться на свою память – записывать пароль на бумаге и ограничить доступ

посторонних лиц (сейф, от которого ключи только у администратора и, может быть, его прямого начальника).

1.2. Корпоративная информационная система и её структура

Корпоративная информационная система (КИС) – это масштабируемая система, предназначенная для комплексной автоматизации всех видов хозяйственной деятельности больших и средних предприятий, в том числе корпораций, состоящих из группы компаний, требующих единого управления [3]. Объединяет систему управления персоналом, материальными, финансовыми и другими ресурсами компании, используется для поддержки планирования и управления компанией, для поддержки принятия управленческих решений ее руководителями. Под КИС можно понимать управленческую идеологию, объединяющую бизнес-стратегию и информационные технологии.

КИС предполагает возможность работы системы в распределенной структуре (корпорации) по вертикали и горизонтали.

Основные принципы, на которых должна быть построена КИС:

- интеллектуальность (управление организацией – регистрация и накопление информации);
- интегрированность (сквозное прохождение документов через различные службы);
- модульность (возможность поэтапного внедрения системы);
- доступность;
- открытость (возможность взаимодействовать с другими программами);
- адаптивность (мощность механизма настроек).

Основные требования КИС:

- использование архитектуры клиент-сервер с возможностью применения промышленных СУБД;
- обеспечение безопасности методами контроля и разграничения доступа к информационным ресурсам;
- поддержка распределенной обработки информации;
- модульный принцип построения из оперативно-независимых функциональных блоков с расширением за счет открытых стандартов (API, COM и другие).

Структуру информационной системы можно рассматривать как совокупность подсистем независимо от сферы применения. Подсистема – это часть системы, выделенная по какому-либо признаку. В этом случае говорят о структурном признаке классификации, а подсистемы называют обеспечивающими.

Таким образом, структура любой информационной системы может быть представлена совокупностью обеспечивающих подсистем.

Среди обеспечивающих подсистем обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение – совокупность информационных массивов данных, единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных. Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Техническое обеспечение – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно–справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров. Такая форма организации облегчает управление и внедрение стандартизации, но понижает ответственность и инициативу персонала.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах. В этом случае от персонала требуется больше персональной ответственности, руководству труднее внедрять стандартизацию.

В настоящее время более распространен частично децентрализованный подход – организация технического обеспечения на базе распределен-

ных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав программного обеспечения входят общесистемные и специальные программные продукты, а также техническая документация.

К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи, контрольные примеры.

Организационное обеспечение – это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации ИС.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на 1-м этапе построения БД.

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности. В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования информационной системы включает:

- статус информационной системы;
- права, обязанности и ответственность персонала;
- порядок создания и использования информации и др.

Этот набор подсистем носит общий характер практически для всех типов автоматизированных информационных систем (АИС). Однако структура и сложность обеспечивающих подсистем зависит от типа АИС, области применения и других факторов. Так, подсистема математического обеспечения имеет место в АИС оригинальной разработки программного обеспечения (ПО) – в АИС с типовым ПО, она отсутствует. Подсистема правового обеспечения может отсутствовать в АИС внутрифирменного назначения – в этом случае можно ограничиться подсистемой организационного обеспечения, в которой в том числе решаются вопросы правового обеспечения; АИС самостоятельного назначения, например, системы информационного обслуживания, могут иметь подсистему правового обеспечения. АИС, имеющие БД фактографического характера, имеют только подсистему информационного обеспечения, в которой может возникать необходимость решения отдельных лингвистических вопросов. Документальные системы имеют развитую подсистему лингвистического обеспечения, так как в этих системах решаются сложные задачи обеспечения смысловой релевантности запросов пользователей содержанию выданных документов. А это, как правило, не только программные модули морфологического анализа, но и совокупность словарей и правил их ведения.

1.3. Составные части информационной системы

Аппаратное обеспечение является основой ИВС и определяет вычислительную мощность ИВС в целом. Все аппаратное обеспечение можно разделить на вычислительные установки, кабельное, канало- и сетеобразующее, периферийное и дополнительное оборудование [2].

Вычислительные установки (ВУ) служат для выполнения основных вычислительных задач, т.е. задач по хранению и обработке информации. Вычислительные установки можно разделить на две большие группы: серверы и рабочие станции.

Сервер (Server) – это вычислительная установка, которая служит преимущественно для совместного использования его информационно-вычислительных ресурсов, к которым относятся, прежде всего, центральный процессор или процессоры (например, если это SMP-система), оперативная и внешняя память (прежде всего, жесткие диски).

Основные требования к современному серверу:

1. Масштабируемость (Scalability) – возможность наращивания мощности ВУ (количество и быстродействие процессоров, объем оперативной и внешней памяти) для пропорционального увеличения скорости и плотности (определенное количество запросов в единицу времени) обработки запросов, а также объемов хранимой информации.

2. Отказоустойчивость (Fault-tolerance) – возможность системы полностью восстанавливать свою работоспособность при аппаратных сбоях и *высокая доступность* (*High Level of Availability*) – возможность системы продолжать обслуживание запросов при аппаратных сбоях. Обеспечивается *дублированием* (*Duplexing*) основных аппаратных компонентов ВУ, чаше всего выходящих из строя (обычно имеющих механические части, а также *избыточностью (Redundancy)* хранящейся информации.

3. Управляемость (Manageability) – возможность удаленного управления, сбора сведений о работе подсистем сервера. Обеспечивается специальными программно-аппаратными комплексами, разрабатываемыми и поставляемыми производителями серверов.

Для обеспечения отказоустойчивости и высокой доступности в современных серверах используются следующие технологии и компоненты:

- горячая замена компонент *(Hot Swapping)* – позволяет менять компоненты аппаратного обеспечения, не отключая электропитания от ВУ. Есть решения для жестких дисков, источников питания, вентиляторов и плат расширения;

- оперативная память (ОЗУ) с хранением *избыточной информации;*

- память с паритетом *(Parity Checking)* – обеспечивается обнаружение однократных ошибок в ОЗУ;

- ECC-память (Enhanced Correction Code – улучшенный код коррекции), обеспечивается исправление однократных ошибок и обнаружение двукратных ошибок в ОЗУ;

- *Массивы независимых резервных дисков (Redundant Array of Independent Disks / RAID).* Применяются в серверах для обеспечения отказоустойчивости внешней памяти.

Классификация RAID по способу исполнения:

1. *Аппаратный RAID.* Существует две реализации:

- в виде хост-адаптера – вместо SCSI-адаптера шина со SCSI-дисками подключается к RAID-адаптеру;

- SCSI-to-SCSI – такой RAID является обычным SCSI-устройством с точки зрения SCSI-адаптера, при этом можно организовать более емкую внешнюю память, являющуюся отказоустойчивой.

2. *Программный RAID. Реализуется* системным ПО на уровне ядра ОС.

Классификация по принципу функционирования:

1. *RAID0 – разделение*;

2. *RAID1 – зеркалирование* (дублирование) данных;

3. *RAID4 – разделение* данных с избыточностью (с выделенным диском четности);

4. *RAID5 – разделение* данных с избыточностью (с равноправными дисками, т.е. информация о четности размыта по дискам).

Кластер – это объединение двух и более ВУ (точнее, пары «процессор + оперативная память»), называемых узлами кластера, для работы с общей внешней памятью. При выходе из строя одного из узлов кластера, остальные узлы кластера берут на себя нагрузку по обслуживанию клиентских подключении. Для клиентов кластер выглядит как один узел сети.

Рабочая станция (Workstation) – это вычислительная установка, которая преимущественно используется как индивидуальное рабочее место пользователя ИВС и служит точкой входа в ИВС.

Основные требования к рабочей станции:

1. *Удобство работы (Convenience)* – обеспечивается прежде всего установкой и поддержкой высокоскоростной графической подсистемой ввода-вывода (графическая плата, монитор, мышь).

2. У*правляемость (Managability)* – обеспечивается ПО, разрабатываемым и поставляемым производителями рабочих станций, а также независимыми производителями.

Кабельное оборудование (кабельная система) представляет собой физическую среду, которая связывает воедино разрозненные ВУ и другое оборудование ИВС. Кабельное оборудование представлено *кабелями* различных типов, а также специальными *розетками* и *вилками* для подключения кабельных сегментов друг к другу и к сетевому оборудованию.

Распространенные типы кабелей:

- коаксиальный кабель;

- витая пара;

- оптоволоконный кабель.

Коаксиальный кабель (Coaxial Cable) – представляет собой изолированную медную жилу, помещенную в медную оплетку, покрытую гибкой изоляционной оболочкой.

Кабель на основе «витых пар» (Twisted Pairs Cable) – представляет собой изолированные медные провода, попарно скрученные и заключенные в гибкую оболочку. Существует *неизолированный (UTP)* и *изолированный (STP)* варианты данного типа кабеля. В последнем случае скрученные пары проводов заключаются в медную оплетку, которая заземляется. Характеризуется так называемой *категорией,* в частности, для сетей на базе технологии Ethernet допускается использование кабеля категории 3 и выше. С кабелем данного типа используются вилки и розетки стандарта *RJ-45*. Используется для построения сети по топологии «звезда».

Оптоволоконный кабель (Fiber Optical Cable) – представляет собой стеклянную жилу (*световод),* заключенную в гибкую оболочку. Используется для построения сети по топологии *«точка-точка».* Применяется для построения магистралей, т.е. создания каналов связи между удаленными частями сети, а также для подключения серверов.

Существуют две разновидности данного кабеля:

- *многомодовый* – допускается передача нескольких пучков света («мод») по одному световоду, при этом обеспечивается дальность связи *до 2 км:*

- *одномодовый* – вследствие меньшего диаметра световода возможна передача только одного пучка света, при этом обеспечивается дальность связи до *80 км* (теоретически возможная).

Канало- и сетеобразующее оборудование (или просто «сетевое оборудование») – это оборудование для сопряжения кабельной системы ИВС с ВУ, а также различных частей кабельной системы. *Каналообразующее* оборудование обеспечивает функции канального уровня модели OSI для организации сети, а *сетеобразующее* – функции канального и сетевого уровня модели OSI.

Сетевое оборудование можно разделить на две группы:

Оконечное оборудование – сетевые платы и модемы, которые устанавливаются в ВУ и обеспечивают подключение ВУ к сети.

Коммутационное оборудование – концентраторы, мосты и коммутаторы, маршрутизаторы, которые служат для связи частей кабельной системы в единую сетевую инфраструктуру.

Периферийное оборудование – это оборудование, расширяющее функциональные возможности ВУ (прежде всего функциями ввода, вывода). Периферийное оборудование подключается прямо к ВУ посредством специализированных *интерфейсов,* либо посредством канало- и сетеобразующего оборудования. Включает мониторы, клавиатуры, мыши, принтеры, сканеры, дисковые массивы и т.д.

Дополнительное оборудование – оборудование, необходимое для более эффективной и надежной работы основного оборудования ИВС.

Включает, прежде всего, источники бесперебойного питания (далее ИБП), а также анализаторы сети, датчики состояния окружающей среды и т.п.

Существуют *два подхода* к защите оборудования от неисправностей электропитания, предусматривающих использование ИБП:

Централизованный подход – все компьютерное оборудование подключено к одному мощному ИБП, который постоянно работает и обеспечивает это оборудование электропитанием в течение достаточно продолжительного периода времени в случае сбоев.

Подход на основе *распределенной схемы защиты* электропитания – каждый узел сети (рабочая станция, сервер, маршрутизатор и т.д.) подключается при необходимости к отдельному ИБП, который и обеспечивает некоторое время работу узла сети в случае сбоев в электропитании.

Программное обеспечение (Software) служит посредником между аппаратным обеспечением ИВС и пользователем ИВС при доступе последнего к ресурсам ИВС и выполнении различных информационно-вычислительных задач.

Деление по функциональным возможностям:

1. *Серверная операционная система (далее СОС)* – хранится на дисках сервера и выполняется на процессоре(-ах) сервера, обслуживая другие информационно-вычислительные задачи (СУБД, почтовая система и т.д.). В зависимости от производителя и версии СОС обладает различной функциональностью и возможностями.

2. *Клиентская операционная система (далее КОС)* – хранится на дисках рабочей станции (или на дисках сервера), выполняется на процессоре рабочей станции, обеспечивая пользователю ИВС базовый интерфейс (средство взаимодействия) для доступа к ресурсам ИВС. Также может обслуживать дополнительные задачи.

3. *Система управления базами данных (далее СУБД)* – служит для эффективного хранения и обработки большого объема упорядоченной определенным способом информации. На сегодняшний день чаще всего используются СУБД, поддерживающие реляционную модель хранения данных.

4. *Почтовая система* – служит для взаимодействия пользователей ИВС посредством самой ИВС, аналог обычной почты, реализованный в электронном виде. Система групповой работы (Groupware) – более совершенное средство взаимодействия пользователей, позволяет упорядочить и формализовать обмен сообщениями.

5. *Средства обеспечения взаимодействия с Internet/Intranet* — работа пользователей в ИВС на базе ГВС предполагает на сегодня работу в Internet. Intranet – ИВС предприятия, использующая средства Internet для транспортировки своих информационных потоков между разбросанными по земному шару частями ИВС.

6. *ПО для обеспечения прикладных сервисов* – серверы WWW. FTP, SMTP/POP3 и т.п.

7. *ПО для получения доступа к прикладным сервисам* – броузеры Интернет, FTP-клиенты, POP3-клиенты.

8. *ПО на границе ЛВС/ГВС* для обеспечения безопасности корпоративных сетей – брандмауэры (Firewalls), прокси-серверы (Proxy), шлюзы (Gateways), туннели (Tunnels).

9. *Средства сетевого и системного управления.* Администратору большой ИВС требуется специальный инструментарий, позволяющий легко выполнять задачи по администрированию, сопровождению и управлению частями и компонентами ИВС.

10. *Прикладное ПО* – не связанное напрямую с ресурсами ИВС ПО. Служит для решения задач прикладной области: работа в офисе, автоматизация работы бухгалтерии, графическое макетирование и издательская деятельность и т.п.

11. *Дополнительное ПО* – облегчающее и делающее более удобной работу пользователей ИВС.

Деление ПО на системное и прикладное:

- *системное ПО* – служит для выполнения задач по обслуживанию ИВС, прежде всего ее аппаратного обеспечения. К системному ПО относится большая часть программных компонент в составе ОС, а также различное ПО для обслуживания аппаратного обеспечения ИВС: ПО для резервного копирования, ПО для настройки сетевого оборудования и т.д.

- *прикладное ПО* – служит для выполнения информационно-вычислительных задач, решаемых обычными пользователями ИВС. К прикладному ПО относятся СУБД, почтовая система, программные пакеты для работы в офисе и т.д.

Деление по месту выполнения:

- *Серверное ПО* – выполняющееся как один и более процессов на ВУ, выполняющей роль сервера.

- *Клиентское ПО* – выполняющееся как один и более процессов на ВУ, выполняющей роль рабочей станции.

- *Клиент-серверное ПО* – распределенное ПО, выполняющееся как два и более процесса на двух и более ВУ.

Современное ПО не является монолитным и чаще всего строится по модульному принципу на основе уровневой архитектуры. В современном ПО можно выделить следующие основные уровни (или слои):

1. *Уровень представления информации* (уровень интерфейса с пользователем) – является передним краем приложения *(FrantEnd),* обращенным к пользователям. На этом уровне реализуется ввод информации для последующей обработки функциональными блоками и вывод обработанной информации. На сегодняшний день этот уровень чаще всего реализуется через функции программного интерфейса ОС, реализующие работу с примитивами графического интерфейса (например, Windows GDI API): окна, меню, панели инструментов, кнопки.

2. *Уровень бизнес-правил* (функциональный уровень) – является функциональной частью приложения и отвечает за проверку на допустимость, обработку и преобразование информации. На сегодняшний день налицо тенденция распределять слой бизнес-правил по нескольким ВУ.

3. *Уровень именования и идентификации* – отвечает за именование и идентификацию информационных ресурсов, а также аутентификацию пользователей в рамках программной системы. Данный уровень может использовать внешнюю службу именования и идентификации ресурсов и пользователей (например, службу справочника в составе серверной ОС).

4. *Уровень безопасности* – отвечает за разграничение прав доступа пользователей и проверку полномочий при доступе к информационным ресурсам через уровень представления. Данный уровень тесно взаимодействует с уровнем именования и идентификации, поэтому также может использовать внешнюю службу для обеспечения безопасности.

5. *Уровень* оптимизации – выполняет анализ занятости вычислительных ресурсов и оптимально перераспределяет вычислительную и т.п. (см. выше рассмотренные уровни) нагрузку по установкам, доступным приложению.

6. *Уровень хранения и извлечения информации* – является базовой и наиболее удаленной от пользователей частью приложения, обращенной к ресурсам ВУ *(BackEnd),* обеспечивает эффективные структуры хранения введенной через приложение информации, а также алгоритмы извлечения информации для последующей обработки и отображения. Может использовать внешнюю СУБД либо самостоятельно реализовывать вышеуказанные структуры н алгоритмы (например, файловая система в составе ОС).

На сегодняшний день программное обеспечение разрабатывается на основе нескольких *моделей вычислений* в зависимости от места реализации тех или иных уровней приложения:

Локализованная / централизованная модель вычислений – обработка и хранение данных осуществляется на одной ВУ. На основе этой модели реализуется большинство примеров современного прикладного ПО, некоторые почтовые системы и т.д.

Модель вычислений *на основе файлового хранилища* – разновидность локальной модели вычислений, только данные хранятся не на локальном диске ВУ, а на файловом сервере.

Распределенная модель вычислений – обработка и хранение данных осуществляется на двух и более ВУ. Наиболее яркими и распространенными на сегодняшний день разновидностями являются:

– *клиент-серверная модель.* Такая модель вычислений реализована в современных СУБД с поддержкой SQL, также в современных почтовых системах и ПО групповой работы. С использованием этой модели работает большинство служб сетевых ОС, имеются успешные попытки встраивания этой модели вычислений в ОС для выполнения прикладного ПО.

– модель на основе *сервера приложений/монитора транзакций* – реализуется пока ограниченно, чаще для доступа к ресурсам обычных клиент-серверных приложений через Web-интерфейс. Также есть попытки встраивания в ОС.

ПО, реализующее распределенную модель вычислений, называется *распределенным ПО*. В составе распределенного ПО должен быть реализован уровень взаимодействия – дополнительный уровень, который обеспечивает взаимодействие программных компонент, выполняющихся на разных ВУ.

1.4. Распределённая информационная система и схемы её построения

Под распределенными понимаются ИС, которые не располагаются на одной контролируемой территории, на одном объекте.

Распределенная информационная система (РИС) – любая информационная система, позволяющая организовать взаимодействие независимых, но связанных между собой ЭВМ. Эти системы предназначены для автоматизации таких объектов, которые характеризуются территориальной распределённостью пунктов возникновения и потребления информации.

В общем случае распределенная информационная система (РИС) представляет собой множество сосредоточенных ИС, связанных в единую систему с помощью коммуникационной подсистемы.

Сосредоточенными ИС могут быть:

- отдельные ЭВМ, в том числе и ПЭВМ,
- вычислительные системы и комплексы,
- локальные вычислительные сети (ЛВС).

В настоящее время практически не используются неинтеллектуальные абонентские пункты, не имеющие в своем составе ЭВМ. Поэтому правомочно считать, что наименьшей структурной единицей РИС является ЭВМ (рис. 1).

Распределенные ИС строятся по сетевым технологиям и представляют собой вычислительные сети.

Термин «распределенная система», подразумевает взаимосвязанный набор автономных компьютеров, процессов или процессоров. Компьютеры, процессы или процессоры упоминаются как узлы распределенной системы. Будучи определенными как «автономные», узлы должны быть, по крайней мере, оборудованы своим собственным блоком управления. Таким образом, параллельный компьютер с одним потоком управления и несколькими потоками данных (SIMD) не подпадает под определение распределенной системы. Чтобы быть определенными как «взаимосвязанными», узлы должны иметь возможность обмениваться информацией.

Так как процессы могут играть роль узлов системы, определение включает программные системы, построенные как набор взаимодейст-

вующих процессов, даже если они выполняются на одной аппаратной платформе. В большинстве случаев, однако, распределенная система будет, по крайней мере, содержать несколько процессоров, соединенный коммутирующей аппаратурой.

Коммуникационная подсистема включает в себя:

- коммуникационные модули (КМ);
- каналы связи;
- концентраторы;
- межсетевые шлюзы (мосты).

Основной функцией коммуникационных модулей является передача полученного пакета к другому КМ или абонентском пункту в соответствии с маршрутом передачи. Коммуникационный модуль называют также центром коммутации пакетов.

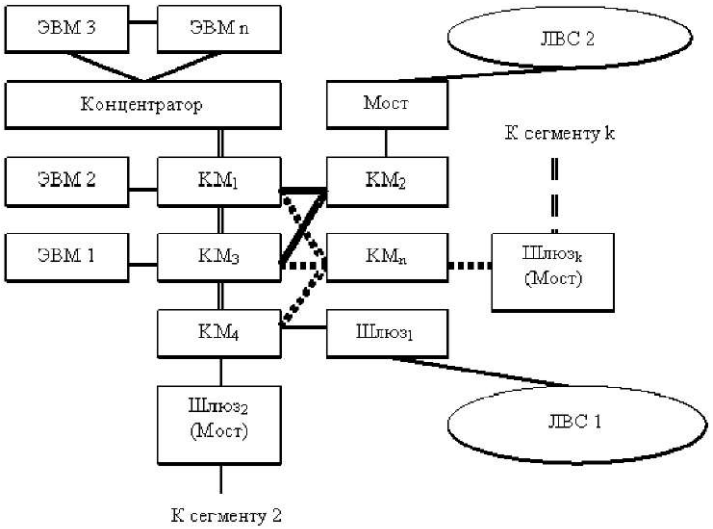


Рисунок 1 – Фрагмент распределенной информационной системы

Каналы связи объединяют элементы сети в единую сеть, каналы могут иметь различную скорость передачи данных.

Концентраторы используются для уплотнения информации перед передачей ее по высокоскоростным каналам.

Межсетевые шлюзы и мосты используются для связи сети с ЛВС или для связи сегментов глобальных сетей. С помощью мостов связываются сегменты сети с одинаковыми сетевыми протоколами.

В любой РИС в соответствии с функциональным назначением может быть выделено три подсистемы:

- пользовательская подсистема;
- подсистема управления;
- коммуникационная подсистема.

Пользовательская или абонентская подсистема включает в себя информационные системы пользователей (абонентов) и предназначается для удовлетворения потребностей пользователей в хранении, обработке и получении

Наличие подсистемы управления позволяет объединить все элементы РИС в единую систему, в которой взаимодействие элементов осуществляется по единым правилам. Подсистема обеспечивает взаимодействие элементов системы путем сбора и анализа служебной информации и воздействия на элементы с целью создания оптимальных условий для функционирования всей сети.

Коммуникационная подсистема обеспечивает передачу информации в сети в интересах пользователей и управления РИС.

Функционирование РИС можно рассматривать как взаимодействие удаленных процессов через коммуникационную подсистему.

Процессы вычислительной сети порождаются пользователями (абонентами) и другими процессами.

Взаимодействие удаленных процессов заключается в:

- обмене файлами,
- пересылке сообщений по электронной почте,
- посылке заявок на выполнение программ и получение результатов,
- обращении к базам данных и т. д.

Концептуально распределенная обработка данных подразумевает тот или иной вид организации сети связи и децентрализацию трех категорий ресурсов:

- аппаратных вычислительных средств и собственно вычислительной мощности;
- баз данных;
- управление системой.

В распределенных информационных системах в той или иной степени осуществляется реализация следующих основных функций:

- доступ к ресурсам (вычислительным мощностям, программам, данным и т. п.) с терминалов и из пользовательских программ в режиме «файл-сервер»;
- выполнение заданий и интерактивное общение пользователей с запущенными по их требованию программами в режиме «клиент-сервер»;
- сбор статистики о функционировании системы;
- обеспечение надежности и живучести системы в целом.

В настоящее время применяют различные подходы к классификации распределенных информационных систем по разным критериям.

Объединение компьютеров в сеть предоставляет возможность программам, работающим на отдельных компьютерах, оперативно взаимодействовать и сообща решать задачи пользователей. Связь между некоторыми программами может быть настолько тесной, что их удобно рассматривать в качестве частей одного приложения, которое называют в этом случае распределенным, или сетевым.

Распределенные приложения обладают рядом потенциальных преимуществ по сравнению с локальными. Среди этих преимуществ более высокая производительность, отказоустойчивость, масштабируемость и приближение к пользователю.

Значительная часть приложений, работающих в компьютерах сети, являются сетевыми, но, конечно, не все. Действительно, ничто не мешает пользователю запустить на своем компьютере полностью локальное приложение, не использующее имеющиеся сетевые коммуникационные возможности. Достаточно типичным является сетевое приложение, состоящее из двух частей. Например, одна часть приложения работает на компьютере, хранящем базу данных большого объема, а вторая — на компьютере пользователя, который хочет видеть на экране некоторые статистические характеристики данных, хранящихся в базе. Первая часть приложения выполняет поиск в базе записей, отвечающих определенным критериям, а вторая занимается статистической обработкой этих данных, представлением их в графической форме на экране, а также поддерживает диалог с пользователем, принимая от него новые запросы на вычисление тех или иных статистических характеристик. Можно представить себе случаи, когда приложение распределено и между большим числом компьютеров.

Распределенным в сетях может быть не только прикладное, но и системное программное обеспечение – компоненты операционных систем. Как и в случае локальных служб, программы, которые выполняют некоторые общие и часто встречающиеся в распределенных системах функции, обычно становятся частями операционных систем и называются сетевыми службами.

Целесообразно выделить три основных параметра организации работы приложений в сети. К ним относятся:

- способ разделения приложения на части, выполняющиеся на разных компьютерах сети;
- выделение специализированных серверов в сети, на которых выполняются некоторые общие для всех приложений функции;
- способ взаимодействия между частями приложений, работающих на разных компьютерах.

Очевидно, что можно предложить различные схемы разделения приложений на части, причем для каждого конкретного приложения можно предложить свою схему. Существуют и типовые модели распределенных приложений. В следующей достаточно детальной модели предлагается разделить приложение на шесть функциональных частей [4]:

• средства представления данных на экране, например, средства графического пользовательского интерфейса;

• логика представления данных на экране описывает правила и возможные сценарии взаимодействия пользователя с приложением: выбор из системы меню, выбор элемента из списка и т.п.;

• прикладная логика – набор правил для принятия решений, вычислительные процедуры и операции;

• логика данных – операции с данными, хранящимися в некоторой базе, которые нужно выполнить для реализации прикладной логики;

• внутренние операции базы данных – действия СУБД, вызываемые в ответ на выполнение запросов логики данных, такие как поиск записи по определенным признакам;

• файловые операции – стандартные операции над файлами и файловой системой, которые обычно являются функциями операционной системы.

На основе этой модели можно построить несколько схем распределения частей приложения между компьютерами сети.

Распределение приложения между большим числом компьютеров может повысить качество его выполнения (скорость, количество одновременно обслуживаемых пользователей и т.д.), но при этом существенно усложняется организация самого приложения, что может просто не позволить воспользоваться потенциальными преимуществами распределенной обработки. Поэтому на практике приложение обычно разделяют на две или три части и достаточно редко – на большее число частей. Наиболее распространенной является двухзвенная схема, распределяющая приложение между двумя компьютерами. Перечисленные выше типовые функциональные части приложения можно разделить между двумя компьютерами различными способами.

Рассмотрим сначала два крайних случая двухзвенной схемы, когда нагрузка в основном ложится на один узел – либо на центральный компьютер, либо на клиентскую машину.

В централизованной схеме (рис. 2 а) компьютер пользователя работает как терминал, выполняющий лишь функции представления данных, тогда как все остальные функции передаются центральному компьютеру. Ресурсы компьютера пользователя используются в этой схеме в незначительной степени, загруженными оказываются только графические средства подсистемы ввода-вывода ОС, отображающие на экране окна и другие графические примитивы по командам центрального компьютера, а также сетевые средства ОС, принимающие из сети команды центрального компьютера и возвращающие данные о нажатии клавиш и координатах мыши. Программа, работающая на компьютере пользователя, часто называется эмулятором терминала – графическим или текстовым, в зависимости от поддерживаемого режима. Фактически эта схема повторяет организацию многотерминальной системы на базе мэйнфрейма с тем лишь отличием,

что вместо терминалов используются компьютеры, подключенные не через локальный интерфейс, а через сеть, локальную или глобальную.

Главным и очень серьезным недостатком централизованной схемы является ее недостаточная масштабируемость и отсутствие отказоустойчивости. Производительность центрального компьютера всегда будет ограничителем количества пользователей, работающих с данным приложением, а отказ центрального компьютера приводит к прекращению работы всех пользователей. Именно из-за этих недостатков централизованные вычислительные системы, представленные мэйнфреймами, уступили место сетям, состоящим из мини-компьютеров, RISC-серверов и персональных компьютеров. Тем не менее централизованная схема иногда применяется как из-за простоты организации программы, которая почти целиком работает на одном компьютере, так и из-за наличия большого парка не распределенных приложений.

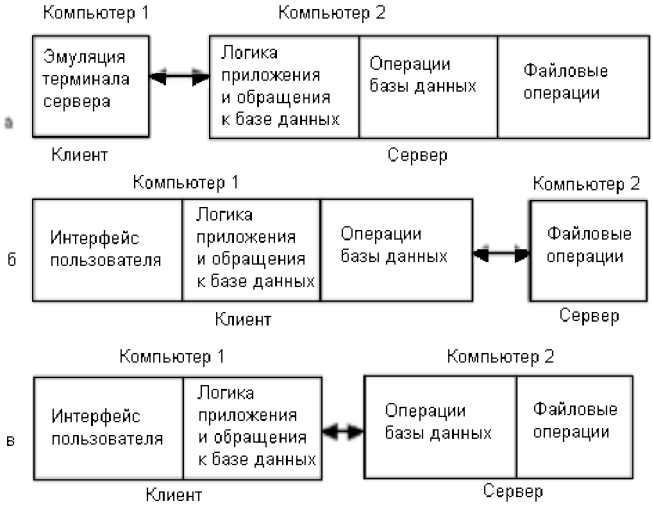


Рис. 2 – Варианты распределений частей приложения по двухзвенной схеме

В схеме «файловый сервер» (рис. 2, б) на клиентской машине выполняются все части приложения, кроме файловых операций. В сети имеется достаточно мощный компьютер, имеющий дисковую подсистему большого объема, который хранит файлы, доступ к которым необходим большому числу пользователей. Этот компьютер играет роль файлового сервера, представляя собой централизованное хранилище данных, находящихся в разделяемом доступе. Распределенное приложение в этой схеме мало отличается от полностью локального приложения. Единственным отличием является обращение к удаленным файлам вместо локальных. Для того чтобы в этой схеме можно было использовать локальные приложения, в сете-

вые операционные системы ввели такой компонент сетевой файловой службы, как редиректор, который перехватывает обращения к удаленным файлам (с помощью специальной нотации для сетевых имен, такой, например, как //server"!/doc/file1.txt) и направляет запросы в сеть, освобождая приложение от необходимости явно задействовать сетевые системные вызовы.

Файловый сервер представляет собой компонент наиболее популярной сетевой службы – сетевой файловой системы, которая лежит в основе многих распределенных приложений и некоторых других сетевых служб. Первые сетевые ОС (NetWare компании Novell, IBM PC LAN Program, Microsoft MS-Net) обычно поддерживали две сетевые службы – файловую службу и службу печати, оставляя реализацию остальных функций разработчикам распределенных приложений.

Такая схема обладает хорошей масштабируемостью, так как дополнительные пользователи и приложения добавляют лишь незначительную нагрузку на центральный узел – файловый сервер. Однако эта архитектура имеет и свои недостатки:

- во многих случаях резко возрастает сетевая нагрузка (например, многочисленные запросы к базе данных могут приводить к загрузке всей базы данных в клиентскую машину для последующего локального поиска нужных записей), что приводит к увеличению времени реакции приложения;
- компьютер клиента должен обладать высокой вычислительной мощностью, чтобы справляться с представлением данных, логикой приложения, логикой данных и поддержкой операций базы данных.

Другие варианты двухзвенной модели более равномерно распределяют функции между клиентской и серверной частями системы. Наиболее часто используется схема, в которой на серверный компьютер возлагаются функции проведения внутренних операций базы данных и файловых операций (рис. 2, в). Клиентский компьютер при этом выполняет все функции, специфические для данного приложения, а сервер – функции, реализация которых не зависит от специфики приложения, из-за чего эти функции могут быть оформлены в виде сетевых служб. Поскольку функции управления базами данных нужны далеко не всем приложениям, то в отличие от файловой системы они чаще всего не реализуются в виде службы сетевой ОС, а являются независимой распределенной прикладной системой. Система управления базами данных (СУБД) является одним из наиболее часто применяемых в сетях распределенных приложений. Не все СУБД являются распределенными, но практически все мощные СУБД, позволяющие поддерживать большое число сетевых пользователей, построены в соответствии с описанной моделью клиент-сервер. Сам термин «клиент-сервер» справедлив для любой двухзвенной схемы распределения функций, но исторически он оказался наиболее тесно связанным со схемой, в которой сервер выполняет функции по управлению базами данных (и, конечно,

файлами, в которых хранятся эти базы) и часто используется как синоним этой схемы.

Трёхзвенная архитектура позволяет еще лучше сбалансировать нагрузку на различные компьютеры в сети, а также способствует дальнейшей специализации серверов и средств разработки распределенных приложений. Примером трёхзвенной архитектуры может служить такая организация приложения, при которой на клиентской машине выполняются средства представления и логика представления, а также поддерживается программный интерфейс для вызова частей приложения второго звена – промежуточного сервера (рис. 3).

Промежуточный сервер называют в этом варианте сервером приложений, так как на нем выполняются прикладная логика и логика обработки данных, представляющих собой наиболее специфические и важные части большинства приложений. Слой логики обработки данных вызывает внутренние операции базы данных, которые реализуются третьим звеном схемы – сервером баз данных.

Сервер баз данных, как и в двухзвенной модели, выполняет функции двух последних слоев – операции внутри базы данных и файловые операции.

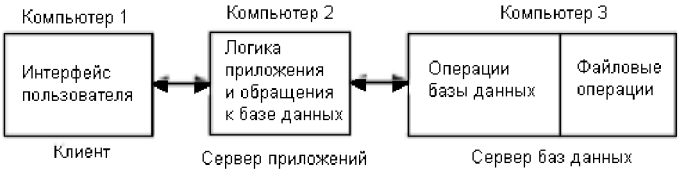


Рисунок 3 – Трёхзвенная схема распределения частей приложения

Централизованная реализация логики приложения решает проблему недостаточной вычислительной мощности клиентских компьютеров для сложных приложений, а также упрощает администрирование и сопровождение. В том случае, когда сервер приложений сам становится узким местом, в сети можно применить несколько серверов приложений, распределив каким-то образом запросы пользователей между ними. Упрощается и разработка крупных приложений, так как в этом случае четко разделяются платформы и инструменты для реализации интерфейса и прикладной логики, что позволяет с наибольшей эффективностью реализовывать их силами специалистов узкого профиля.

Монитор транзакций представляет собой популярный пример программного обеспечения, не входящего в состав сетевой ОС, но выполняющего функции, полезные для большого количества приложений. Такой монитор управляет транзакциями с базой данных и поддерживает целостность распределенной базы данных.

Трехзвенные схемы часто применяются для централизованной реализации в сети некоторых общих для распределенных приложений функций,

отличных от файлового сервиса и управления базами данных. Программные модули, выполняющие такие функции, относят к классу middleware, то есть промежуточному слою, располагающемуся между индивидуальной для каждого приложения логикой и сервером баз данных.

В крупных сетях для связи клиентских и серверных частей приложений также используется и ряд других средств, относящихся к классу middleware, в том числе:

- средства асинхронной обработки сообщений (message-oriented middleware, MOM);
- средства удаленного вызова процедур (Remote Procedure Call, RFC);
- брокеры запроса объектов (Object Request Broker, ORB), которые находят объекты, хранящиеся на различных компьютерах, и помогают их использовать в одном приложении или документе.

Эти средства помогают улучшить качество взаимодействия клиентов с серверами за счет промышленной реализации достаточно важных и сложных функций, а также упорядочить поток запросов от множества клиентов к множеству серверов, играя роль регулировщика, распределяющего нагрузку на серверы.

Сервер приложений должен базироваться на мощной аппаратной платформе (мультипроцессорные системы, специализированные кластерные архитектуры). ОС сервера приложений должна обеспечивать высокую производительность вычислений, а значит, поддерживать многопоточную обработку, вытесняющую многозадачность, мультипроцессирование, виртуальную память и наиболее популярные прикладные среды.

Контрольные вопросы

1. Перечислите, что должен знать системный администратор.
2. Перечислите, что должен уметь системный администратор.
3. За что несёт ответственность администратор?
4. Что такое ИВС?
5. Что такое учётная запись пользователя?
6. Что такое аутентификация?
7. Что такое авторизация?
8. Что такое права доступа к ресурсу?
9. Что такое учётная запись пользователя?
10. Что такое список управления доступом?
11. Назовите «золотые правила» администратора.
12. Что понимается под корпоративной информационной системой?
13. Что понимается под техническим обеспечением КИС?
14. Что понимается под математическим и программным обеспечением КИС?
15. Что понимается под информационным обеспечением КИС?
16. Что понимается под организационным обеспечением КИС?

17. Что понимается под правовым обеспечением КИС?
18. Какие основные требования предъявляются к КИС?
19. Что такое сервер?
20. Что такое отказоустойчивость?
21. Какие технологии и компоненты используются для обеспечения отказоустойчивости?
22. Что такое рабочая станция?
23. Что представляет собой коаксиальный кабель?
24. Что является носителем сигнала в витой паре?
25. Что является носителем сигнала в оптоволокне?
26. Какие существуют классификации программного обеспечения?
27. Что понимается под распределённой моделью вычислений?
28. Какое программное обеспечение называется распределённым?
29. Какие функции выполняет системное программное обеспечение?
30. Какие функции выполняет прикладное программное обеспечение?
31. На какие функциональные части обычно делится распределённое приложение?
32. Что означает понятие двухзвенной схемы распределённого приложения?
33. Что означает понятие трёхзвенной схемы распределённого приложения?
34. В чём заключается схема централизованной обработки данных?
35. В чём заключается схема «файл-сервер»?
36. В чём заключается схема «клиент-сервер»?
37. Назовите участников трёхзвенной схемы распределённого приложения?
38. Какие недостатки имеет схема «файл-сервер»?

2. АДМИНИСТРИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

2.1. Сетевое оборудование и топологии вычислительных сетей

Активное (конфигурируемое) сетевое оборудование обладает гибкой функциональностью, зачастую, модульностью. Это оборудование, в отличие от пассивного сетевого оборудования, выполняет ту или иную цифровую обработку сигналов, тогда как пассивное сетевое оборудование выполняет только передачу сигналов в том виде, в каком получает.

Примеры активного оборудования:

– коммутатор (Switch);

– маршрутизатор (Router);

– конвертер среды передачи данных (Gateway).

Активное сетевое оборудование почти всегда поставляется в состоянии, готовом к немедленной работе, но действует оно так же, как не активное. Исключение составляют конвертеры сред передачи данных.

Для использования всего функционала подобных сетевых устройств необходима дополнительная настройка.

В крупных организациях сетевая инфраструктура постоянно изменяется, это влечёт за собой перенастройку (замену) сетевых устройств. Неактивное оборудование создаёт большое количество "проблем" при изменении топологии сети, самой большой из которых является реакция на увеличивающийся размер сетевых сегментов.

Физическая организация активного сетевого оборудования во многом повторяет архитектуру любого персонального компьютера, изменённую под выполнение специальных задач:

– шина данных обладает большей шириной (128-2048 бит против 32-64 бит);

– большое количество разнообразных сетевых интерфейсов;

– для многих задач используются сопроцессоры;

– дублирование отдельных блоков или целых устройств для повышения отказоустойчивости.

Логическая структура активного сетевого оборудования опирается на UNIX-подобную ОС, в которой для администратора открыты все возможности настройки сетевых протоколов и интерфейсов.

Каждая используемая технология может иметь общую для всех интерфейсов часть, но обязательно должна быть применена к сетевому интерфейсу устройства и настроена в соответствии с требованиями среды передачи данных.

Сложность архитектуры современного сетевого оборудования, во многом, обусловлена неодновременным появлением технологий. Устройства должны поддерживать как современные технологии, так и технологии, возраст которых нередко превышает два десятилетия (x.25).

Таким образом, наиболее сложной задачей для сетевого администратора является нахождение таких настроек оборудование, чтобы переход данных между областями применения различных технологий осуществлялся максимально эффективно.

Пассивное сетевое оборудование представляет собой трассу и тракт, а именно это – кабели и розетки соответственно. И то, и другое оборудование обеспечивает соединения, но разными способами, однако один вид без другого просто не мог бы существовать.

Пассивное оборудование отличается от активного в первую очередь тем, что не питается непосредственно от электросети и передает сигнал без его усиления. Пассивное сетевое оборудование делится условно на две группы. Первая группа включает в себя оборудование, являющееся трассой для кабелей: кронштейны, кабельные каналы и аксессуары для них, металлические лотки, закладные трубы, клипсы, гофрошланги и коммутационные шкафы. Во вторую группу входит оборудование, которое служит трактом передачи данных. Сюда относят розетки, кабели и коммутационные панели.

Общая схема соединения компьютеров в локальные сети называется топологией сети.

Топология – это физическая конфигурация сети в совокупности с ее логическими характеристиками, это стандартный термин, который используется при описании основной компоновки сети. Если понять, как используются различные топологии, то можно будет определить, какими возможностями обладают различные типы сетей.

Существует два основных типа топологий:

- физическая;
- логическая.

Логическая топология описывает правила взаимодействия сетевых станций при передаче данных.

Физическая топология определяет способ соединения носителей данных.

Термин "топология сети" характеризует физическое расположение компьютеров, кабелей и других компонентов сети. Топология физических связей может принимать разные «геометрические» формы, при этом существенным является не геометрическое расположение кабеля, а лишь наличие связи между узлами (замкнутость/незамкнутость, наличие центра и т.д.). Топология сети обуславливает ее характеристики.

Выбор той или иной топологии влияет на:

- состав необходимого сетевого оборудования;
- характеристики сетевого оборудования;
- возможности расширения сети;
- способ управления сетью.

Конфигурация сети может быть или децентрализованной (когда кабель "обегает" каждую станцию в сети), или централизованной (когда каж-

дая станция физически подключается к некоторому центральному устройству, распределяющему фреймы и пакеты между станциями). Примером централизованной конфигурации является звезда с рабочими станциями, располагающимися на концах ее лучей. Децентрализованная конфигурация похожа на цепочку альпинистов, где каждый имеет свое положение в связке, а все вместе соединены одной веревкой. Логические характеристики топологии сети определяют маршрут, проходимый пакетом при передаче по сети.

При выборке топологии нужно учитывать, чтобы она обеспечивала надежную и эффективную работу сети, удобное управление потоками сетевых данных. Желательно также, чтобы сеть по стоимости создания и сопровождения получилась недорогой, но в то же время оставались возможности для ее дальнейшего расширения и, желательно, для перехода к более высокоскоростным технологиям связи. Чтобы решить эту непростую задачу, необходимо знать, какие бывают сетевые топологии.

По топологии связей различают:

- сети с топологией "общая шина (шина)";
- сети с топологией "звезда";
- сети с топологией "кольцо"»;
- сети с древовидной топологией;
- сети со смешанной топологией.

Базовыми являются топологии «шина», «звезда» и «кольцо».

"Шиной" называется топология, в которой компьютеры подключены вдоль одного кабеля.

"Звездой" называется топология, в которой компьютеры подключены к сегментам кабеля, исходящим из одной точки или концентратора.

"Кольцом" называется топология, если кабель, к которому подключены компьютеры, замкнут в кольцо.

Хотя сами по себе базовые топологии несложны, в реальности часто встречаются довольно сложные комбинации, объединяющие свойства нескольких топологий.

В шинной топологии все компьютеры соединяются друг с другом одним кабелем. Каждый компьютер присоединяется к общему кабелю, на концах которого устанавливаются терминаторы. Сигнал проходит по сети через все компьютеры, отражаясь от конечных терминаторов.

Топология "шина" порождается линейной структурой связей между узлами. Аппаратно такая топология может быть реализована, например, путём установки на центральные компьютеры двух сетевых адаптеров. В целях предотвращения отражения сигнала на концах кабеля должны быть установлены терминаторы, поглощающие сигнал.

В сети с топологией "шина" компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов - аппаратных MAC-адресов. Чтобы понять процесс взаимодействия компьютеров по шине, нужно уяснить следующие понятия:

• передача сигнала;
• отражение сигнала;
• терминатор.

1. Передача сигнала

Данные в виде электрических сигналов, передаются всем компьютерам сети; однако информацию принимает только тот, адрес которого соответствует адресу получателя, зашифрованному в этих сигналах. Причем в каждый момент времени только один компьютер может вести передачу. Так как данные в сеть передаются лишь одним компьютером, ее производительность зависит от количества компьютеров, подключенных к шине. Чем их больше, т.е. чем больше компьютеров, ожидающих передачи данных, тем медленнее сеть. Однако вывести прямую зависимость между пропускной способностью сети и количеством компьютеров в ней нельзя. Ибо, кроме числа компьютеров, на быстродействие сети влияет множество факторов, в том числе:

- характеристики аппаратного обеспечения компьютеров в сети;
- частота, с которой компьютеры передают данные;
- тип работающих сетевых приложений;
- тип сетевого кабеля;
- расстояние между компьютерами в сети.

Шина - пассивная топология. Это значит, что компьютеры только "слушают" передаваемые по сети данные, но не перемещают их от отправителя к получателю. Поэтому, если один из компьютеров выйдет из строя, это не скажется на работе остальных. В активных топологиях компьютеры регенерируют сигналы и передают их по сети.

2. Отражение сигнала

Данные, или электрические сигналы, распространяются по всей сети - от одного конца кабеля к другому. Если не предпринимать никаких специальных действий, сигнал, достигая конца кабеля, будет отражаться и не позволит другим компьютерам осуществлять передачу. Поэтому, после того как данные достигнут адресата, электрические сигналы необходимо погасить.

3. Терминатор

Чтобы предотвратить отражение электрических сигналов, на каждом конце кабеля устанавливают заглушки (терминаторы, terminators), поглощающие эти сигналы. Все концы сетевого кабеля должны быть к чему-нибудь подключены, например, к компьютеру или к баррел-коннектору - для увеличения длины кабеля. К любому свободному (неподключенному ни к чему) концу кабеля должен быть подсоединен терминатор, чтобы предотвратить отражение электрических сигналов.

Нарушение целостности сети может произойти, если разрыв сетевого кабеля происходит при его физическом разрыве или отсоединении одного из его концов. Возможна также ситуация, когда на одном или нескольких концах кабеля отсутствуют терминаторы, что приводит к отражению элек-

трических сигналов в кабеле и прекращению функционирования сети. Сеть "падает". Сами по себе компьютеры в сети остаются полностью работоспособными, но до тех пор, пока сегмент разорван, они не могут взаимодействовать друг с другом. У такой топологии сети есть достоинства и недостатки.

Достоинства:

- небольшое время установки сети;
- дешевизна (требуется меньше кабеля и сетевых устройств);
- простота настройки;
- выход из строя рабочей станции не отражается на работе сети.

Недостатки:

- такие сети трудно расширять (увеличивать число компьютеров в сети и количество сегментов – отдельных отрезков кабеля, их соединяющих);
- поскольку шина используется совместно, в каждый момент времени передачу может вести только один из компьютеров;
- "шина" является пассивной топологией – компьютеры только "слушают" кабель и не могут восстанавливать затухающие при передаче по сети сигналы;
- надёжность сети с топологией "шина" невысокая. Когда электрический сигнал достигает конца кабеля, он (если не приняты специальные меры) отражается, нарушая работу всего сегмента сети.

Проблемы, характерные для топологии "шина", привели к тому, что эти сети сейчас уже практически не используются.

Топология сети типа "шина" известна как логическая топология Ethernet 10 Мбит/с.

При топологии "звезда" все компьютеры подключаются к центральному компоненту, именуемому концентратором (hub). Каждый компьютер подсоединяется к сети при помощи отдельного соединительного кабеля. Сигналы от передающего компьютера поступают через концентратор ко всем остальным.

В «звезде» всегда есть центр, через который проходит любой сигнал в сети. Функции центрального звена выполняют специальные сетевые устройства, причём передача сигнала в них может идти по-разному: в одних случаях устройство направляет данные всем узлам, кроме узла-отправителя, в других устройство анализирует, какому узлу предназначаются данные и направляет их только ему.

Эта топология возникла на заре вычислительной техники, когда компьютеры были подключены к центральному, главному, компьютеру.

Достоинства топологии "звезда":

- выход из строя одной рабочей станции не отражается на работе всей сети в целом;
- хорошая масштабируемость сети;
- лёгкий поиск неисправностей и обрывов в сети;

• высокая производительность сети (при условии правильного проектирования);

• гибкие возможности администрирования.

Недостатки топологии "звезда":

• выход из строя центрального концентратора обернётся неработоспособностью сети (или сегмента сети) в целом;

• для прокладки сети зачастую требуется больше кабеля, чем для большинства других топологий;

• конечное число рабочих станций в сети (или сегменте сети) ограничено количеством портов в центральном концентраторе.

Одна из наиболее распространённых топологий, поскольку проста в обслуживании. В основном используется в сетях, где носителем выступает кабель витая пара. UTP категория 3 или 5. (Категории кабеля «витая пара», которые нумеруются от 1 до 7 и определяют эффективный пропускаемый частотный диапазон. Кабель более высокой категории обычно содержит больше пар проводов, и каждая пара имеет больше витков на единицу длины).

Топология типа "звезда" нашла свое отражение в технологии Fast Ethernet6.

При топологии "кольцо" компьютеры подключаются к кабелю, замкнутому в кольцо. Поэтому у кабеля просто не может быть свободного конца, к которому надо подключать терминатор. Сигналы передаются по кольцу в одном направлении и проходят через каждый компьютер. В отличие от пассивной топологии "шина", здесь каждый компьютер выступает в роли репитера (повторителя), усиливая сигналы и передавая их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть.

Функционирование замкнутой топологии «кольцо» основано на передаче маркера.

Маркер – пакет данных, разрешающий компьютеру передавать данные в сеть. Маркер последовательно, от одного компьютера к другому, передается до тех пор, пока его не получит тот, который "хочет" передать данные. Компьютер, желающий начать передачу, «захватывает» маркер, изменяет его, помещает адрес получателя в данные и посылает их по кольцу получателю.

Данные проходят через каждый компьютер, пока не окажутся у того, чей адрес совпадает с адресом получателя, указанным в данных. После этого принимающий компьютер посылает передающему сообщение, где подтверждает факт приёма данных. Получив подтверждение, передающий компьютер создаёт новый маркер и возвращает его в сеть.

На первый взгляд кажется, что передача маркера отнимает много времени, однако на самом деле маркер передвигается практически со скоростью света. В кольце диаметром 200 метров маркер может циркулировать с частотой 10 000 оборотов в секунду.

Достоинства топологии "кольцо":

- простота установки;
- практически полное отсутствие дополнительного оборудования;
- возможность устойчивой работы без существенного падения скорости передачи данных при интенсивной загрузке сети, поскольку использование маркера исключает возможность возникновения коллизий.

Недостатки топологии "кольцо":

- выход из строя одной рабочей станции, и другие неполадки (обрыв кабеля), отражаются на работоспособности всей сети;
- сложность конфигурирования и настройки;
- сложность поиска неисправностей.

Наиболее широкое применение получила в оптоволоконных сетях. Используется в стандартах FDDI8, Token ring9.

Реальные компьютерные сети постоянно расширяются и модернизируются. Поэтому почти всегда такая сеть является гибридной, т.е. ее топология представляет собой комбинацию нескольких базовых топологий. Легко представить себе гибридные топологии, являющиеся комбинацией "звезды" и "шины", либо "кольца" и "звезды".

Топологию "дерево" (tree), можно рассматривать как объединение нескольких "звезд". Именно эта топология сегодня является наиболее популярной при построении локальных сетей.

В древовидной топологии есть корень дерева, от которого произрастают ветви и листья.

Дерево может быть активным или истинным и пассивным. При активном дереве в центрах объединения нескольких линий связи находятся центральные компьютеры, а при пассивном - концентраторы (хабы).

Довольно часто применяются комбинированные топологии, среди них наиболее распространены звездно-шинная и звездно-кольцевая.

В звездно-шинной (star-bus) топологии используется комбинация шины и пассивной звезды.

К концентратору подключаются как отдельные компьютеры, так и целые шинные сегменты. На самом деле реализуется физическая топология шина, включающая все компьютеры сети. В данной топологии может использоваться и несколько концентраторов, соединенных между собой и образующих так называемую магистральную, опорную шину. К каждому из концентраторов при этом подключаются отдельные компьютеры или шинные сегменты. В результате получается звездно-шинное дерево. Таким образом, пользователь может гибко комбинировать преимущества шинной и звездной топологий, а также легко изменять количество компьютеров, подключенных к сети. С точки зрения распространения информации данная топология равноценна классической шине.

В случае звездно-кольцевой (star-ring) топологии в кольцо объединяются не сами компьютеры, а специальные концентраторы, к которым в

свою очередь подключаются компьютеры с помощью звездообразных двойных линий связи.

В действительности все компьютеры сети включаются в замкнутое кольцо, так как внутри концентраторов линии связи образуют замкнутый контур. Данная топология дает возможность комбинировать преимущества звездной и кольцевой топологий. Например, концентраторы позволяют собрать в одно место все точки подключения кабелей сети. Если говорить о распространении информации, данная топология равноценна классическому кольцу.

Наконец, следует упомянуть о сетчатой, или сеточной (mesh) топологии, в которой все либо многие компьютеры и другие устройства соединены друг с другом напрямую.

Такая топология исключительно надежна - при обрыве любого канала передача данных не прекращается, поскольку возможно несколько маршрутов доставки информации. Сеточные топологии (чаще всего не полные, а частичные) используются там, где требуется обеспечить максимальную отказоустойчивость сети, например, при объединении нескольких участков сети крупного предприятия или при подключении к Интернету, хотя за это, конечно, приходится платить: существенно увеличивается расход кабеля, усложняется сетевое оборудование и его настройка.

В настоящее время, подавляющее большинство современных сетей используют топологию "звезда" или гибридную топологию, представляющую собой объединение нескольких "звезд" (например, топологию типа "дерево"), и метод доступа к среде передачи CSMA/CD (множественный доступ с контролем несущей и обнаружением столкновений).

2.2. Эталонная модель взаимодействия открытых систем (ЭМВОС/OSI) и её основные аспекты

Эталонная модель взаимодействия открытых систем была создана с целью унификации принципов построения и функционирования компьютерных сетей, чтобы позволить любым компьютерным средствам во всём мире объединиться в глобальную сеть и обеспечить правильное взаимодействие между собой при обмене данными посредством различных сетевых технологий. Если бы каждый компьютер интегрировался в компьютерную сеть по своим стандартам, построить глобальную сеть Интернет было бы невозможно.

Для наглядности процесс работы сети в эталонной модели OSI разделен на семь уровней. Эта теоретическая конструкция облегчает изучение и понимание довольно сложных концепций. В верхней части модели OSI располагается приложение, которому нужен доступ к ресурсам сети, в нижней – сама сетевая среда. По мере того как данные продвигаются от уровня к уровню вниз, действующие на этих уровнях протоколы постепенно подготавливают их для передачи по сети. Добравшись до целевой сис-

темы, данные продвигаются по уровням вверх, причем те же протоколы выполняют те же действия, только в обратном порядке. В 1983 г. Международная организация по стандартизации (International Organization for Standardization, ISO) и Сектор стандартизации телекоммуникаций Международного телекоммуникационного союза (Telecommunication Standardization Sector of International Telecommunication Union, ITU-T) опубликовали документ «The Basic Reference Model for Open Systems Interconnection», где была описана модель распределения сетевых функций между 7 различными уровнями [5].

Предполагалось, что эта семиуровневая структура станет основой для нового стека протоколов, но в коммерческой форме он так и не был реализован. Вместо этого модель OSI используется с существующими стеками протоколов в качестве обучающего и справочного пособия. Большая часть популярных в наши дни протоколов появилась до разработки модели OSI, поэтому в точности с ее семиуровневой структурой они не согласуются. Зачастую в одном протоколе совмещены функции двух или даже нескольких уровней модели, да и границы протоколов часто не соответствуют границам уровней OSI. Тем не менее модель OSI остается отличным наглядным пособием для исследования сетевых процессов, и профессионалы часто связывают функции и протоколы с определенными уровнями.

По сути, взаимодействие протоколов, работающих на разных уровнях модели OSI, проявляется в том, что каждый протокол добавляет *заголовок* (header) или (в одном случае) *трейлер* (footer) к информации, которую он получил от уровня, расположенного выше. Например, приложение генерирует запрос к сетевому ресурсу. Этот запрос продвигается по стеку протоколов вниз. Когда он достигает транспортного уровня, протоколы этого уровня добавляют к запросу собственный заголовок, состоящий из полей с информацией, специфической для функций данного протокола. Сам исходный запрос становится для протокола транспортного уровня полем данных (полезной нагрузкой). Добавив свой заголовок, протокол транспортного уровня передает запрос сетевому уровню. Протокол сетевого уровня добавляет к заголовку протокола транспортного уровня свой собственный заголовок. Таким образом, для протокола сетевого уровня полезной нагрузкой становятся исходный запрос и заголовок протокола транспортного уровня. Вся эта конструкция становится полезной нагрузкой для протокола канального уровня, который добавляет к ней заголовок и трейлер.

Итогом этой деятельности является *пакет* (packet), готовый для передачи по сети. Когда пакет достигает места назначения, процесс повторяется в обратном порядке. Протокол каждого следующего уровня стека (теперь снизу-вверх) обрабатывает и удаляет заголовок эквивалентного протокола передающей системы. Когда процесс завершен, исходный запрос достигает приложения которому он предназначен, в том же виде, в каком он был сгенерирован. Процесс добавления заголовков к запросу (рис. 1.8),

сгенериро ванному приложением, называется *инкапсуляцией данных* (data encapsulation). По сути эта процедура напоминает процесс подготовки письма для отправки по почте. Запрос — это само письмо, а добавление заголовков аналогично вкладыванию письма в конверт, написанию адреса, штемпелеванию и собственно отправке.

2.3. Физический уровень

На самом нижнем уровне модели OSI – *физическом* (physical) – определяются характеристики элементов оборудования сети – сетевая среда, способ установки, тип сигналов, используемых для передачи по сети двоичных данных. Кроме того, на физическом уровне определяется, какой тип сетевого адаптера нужно установить на каждом компьютере и какой использовать концентратор (если это нужно). На физическом уровне мы имеем дело с медным или оптоволоконным кабелем, или с каким-либо беспроводным соединением. В ЛВС спецификации физического уровня напрямую связаны с используемым в сети протоколом канального уровня. Выбрав протокол канального уровня, Вы должны использовать одну из спецификаций физического уровня, поддерживаемую этим протоколом. Например, протокол канального уровня Ethernet поддерживает несколько различных вариантов физического уровня – один из двух типов коаксиального кабеля, любой кабель типа «витая пара», оптоволоконный кабель. Параметры каждого из этих вариантов формируются из многочисленных сведений о требованиях физического уровня, например, к типу кабеля и разъемов, допустимой длине кабелей, числу концентраторов и др. Соблюдение этих требований необходимо для нормальной работы протоколов. Например, в чересчур длинном кабеле система Ethernet может не заметить коллизию пакетов, а если система не в состоянии обнаружить ошибки, она не может и исправить их, результат – потеря данных. Стандартом протокола канального уровня определяются не все аспекты физического уровня. Некоторые из них определяются отдельно.

Одна из наиболее часто используемых спецификаций физического уровня описана в документе «Commercial Building Telecommunications Cabling Standard», известном как EIA/TIA 568A. Он опубликован совместно *Американским национальным институтом стандартов* (American National Standards Institute, ANSI), *Ассоциации отраслей электронной промышленности* (Electronics Industry Association, EIA) и *Ассоциацией промышленности средств связи* (Telecommunications Industry Association, TIA). В этот документ включено подробное описание кабелей для сетей передачи данных в промышленных условиях, в том числе минимальное расстояние от источников электромагнитных помех и другие правила прокладки кабеля.

Сегодня кладку кабеля в больших сетях чаще всего поручают специализированным фирмам. Нанятый подрядчик должен быть хорошо знаком с

EIA/TIA 568A и другими подобными документами, а также с правилами эксплуатации зданий в городе. Другой коммуникационный элемент, определяемый на физическом уровне, — тип сигнала для передачи данных по сетевой среде. Для кабелей с медной основой таким сигналом является электрический заряд, для оптоволоконного кабеля — световой импульс. В сетевых средах других типов могут использоваться радиоволны, инфракрасные импульсы и другие сигналы. Помимо природы сигналов, на физическом уровне устанавливается схема их передачи, т. е. комбинация электрических зарядов или световых импульсов, используемая для кодирования двоичной информации, которая сгенерирована вышестоящими уровнями. В системах Ethernet применяется схема передачи сигналов, известная как *манчестерская кодировка* (Manchester encoding), а в системах Token Ring используется *дифференциальная манчестерская* (Differential Manchester) схема.

2.4. Канальный уровень

Протокол *канального* (data-link) уровня обеспечивает обмен информацией между аппаратной частью включенного в сеть компьютера и сетевым ПО. Он подготавливает для отправки в сеть данные, переданные ему протоколом сетевого уровня, и передает на сетевой уровень данные, полученные системой из сети. При проектировании и создании ЛВС используемый протокол канального уровня – самый важный фактор для выбора оборудования и способа его установки. Для реализации протокола канального уровня необходимо следующее аппаратное и программное обеспечение:

• адаптеры сетевого интерфейса (если адаптер представляет собой отдельное устройство, подключаемое к шине, его называют платой сетевого интерфейса или просто сетевой платой);

• драйверы сетевого адаптера;

• сетевые кабели (или другая сетевая среда) и вспомогательное соединительное оборудование;

• сетевые концентраторы (в некоторых случаях).

Как сетевые адаптеры, так и концентраторы разрабатываются для определенных протоколов канального уровня. Некоторые сетевые кабели также приспособлены для конкретных протоколов, но есть и кабели, подходящие для разных протоколов. Безусловно, сегодня (как и всегда) самый популярный протокол канального уровня – Ethernet. Далеко отстал от него Token Ring, за которым следуют другие протоколы, например, FDDI (Fiber Distributed Data Interface). В спецификацию протокола канального уровня обычно включаются три основных элемента:

• формат кадра (т. е. заголовок и трейлер, добавляемые к данным сетевого уровня перед передачей в сеть);

• механизм контроля доступа к сетевой среде;

• одна или несколько спецификаций физического уровня, применяемые с данным протоколом.

Протокол канального уровня добавляет к данным, полученным от протокола сетевого уровня, заголовок и трейлер, превращая их в *кадр* (frame). Если снова прибегнуть к аналогии с почтой, заголовок и трейлер – это конверт для отправки письма. В них содержатся адреса системы-отправителя и системы-получателя пакета. Для протоколов ЛВС, подобных Ethernet и Token Ring, эти адреса представляют собой 6-байтные шестнадцатеричные строки, присвоенные сетевым адаптерам на заводе-изготовителе. Они, в отличие от адресов, используемых на других уровнях модели OSI, называются *аппаратными адресами* (hardware address) или MAC-адресами.

Примечание. Протоколы различных уровней модели OSI по-разному называют структуры, создаваемые ими путем добавления заголовка к данным, пришедшим от вышестоящего протокола. Например, то, что протокол канального уровня называет кадром, для сетевого уровня будет дейтаграммой. Более общим названием для структурной единицы данных на любом уровне является *пакет*.

Важно понимать, что протоколы канального уровня обеспечивают связь только между компьютерами одной и той же ЛВС. Аппаратный адрес в заголовке всегда принадлежит компьютеру в той же ЛВС, даже если целевая система находится в другой сети. Другие важные функции кадра канального уровня – идентификация протокола сетевого уровня, сгенерировавшего данные в пакете, и информация для обнаружения ошибок. На сетевом уровне могут использоваться различные протоколы, и потому в кадр протокола канального уровня обычно включается код, с помощью которого можно установить, какой именно протокол сетевого уровня сгенерировал данные в этом пакете. Руководствуясь этим кодом, протокол канального уровня компьютера-получателя пересылает данные соответствующему протоколу своего сетевого уровня. Для выявления ошибок передающая система вычисляет *циклический избыточный код* (cyclical redundancy check, CRC) полезной нагрузки и записывает его в трейлер кадра. Получив пакет, целевой компьютер выполняет те же вычисления и сравнивает результат с содержимым трейлера. Если результаты совпадают, информация передана без ошибок. В противном случае получатель предполагает, что пакет испорчен, и не принимает его.

Компьютеры в ЛВС обычно используют общую полудуплексную сетевую среду. При этом вполне возможно, что передавать данные начнут одновременно два компьютера. В таких случаях происходит своего рода столкновение пакетов, *коллизия* (collision), при котором данные в обоих пакетах теряются. Одна из главных функций протокола канального уровня – управление доступом к сетевой среде (media access control, MAC), т. е. контроль за передачей данных каждым из компьютеров и сведение к минимуму случаев столкновения пакетов. Механизм управления доступом к

среде – одна из важнейших характеристик протокола канального уровня. В Ethernet для управления доступом к среде используется механизм с контролем несущей и обнаружением коллизий (Carrier Sense Multiple Access with Collision Detection, CSMA/CD). В некоторых других протоколах, например, в Token Ring, используется передача маркера (token passing).

Протоколы канального уровня, используемые в ЛВС, часто поддерживают более одной сетевой среды, и в стандарт протокола включены одна или несколько спецификаций физического уровня. Канальный и физический уровни тесно связаны, т. к. свойства сетевой среды существенно влияют на то, как протокол управляет доступом к среде. Поэтому можно сказать, что в локальных сетях протоколы канального уровня осуществляют также функции физического уровня. В глобальных сетях используются протоколы канального уровня, в которые информация физического уровня не включается, например, SLIP (Serial Line Internet Protocol) и PPP (Point-to-Point Protocol).

2.5. Сетевой уровень

На первый взгляд может показаться, что *сетевой* (network) уровень дублирует некоторые функции канального уровня. Но это не так: протоколы сетевого уровня «отвечают» за *сквозные* (end-to-end) связи, тогда как протоколы канального уровня функционируют только в пределах ЛВС. Иными словами, протоколы сетевого уровня полностью обеспечивают передачу пакета от исходной до целевой системы. В зависимости от типа сети, отправитель и получатель могут находиться в одной ЛВС, в различных ЛВС в пределах одного здания или в ЛВС, разделенных тысячами километров. Например, когда Вы связываетесь с сервером в Интернете, на пути к нему пакеты, созданные Вашим компьютером, проходят через десятки сетей. Подстраиваясь под эти сети, протокол канального уровня неоднократно изменится, но протокол сетевого уровня на всем пути останется тем же самым. Краеугольным камнем набора протоколов TCP/IP (Transmission Control Protocol/Internet Protocol) и наиболее часто используемым протоколом сетевого уровня является протокол IP (Internet Protocol). У Novell NetWare есть собственный сетевой протокол IPX (Internetwork Packet Exchange), а в небольших сетях Microsoft Windows обычно используется протокол NetBEUI (NetBIOS Enhanced User Interface). Большинство функций, приписываемых сетевому уровню, определяются возможностями протокола IP. Подобно протоколу канального уровня, протокол сетевого уровня добавляет заголовок к данным, которые он получил от вышестоящего уровня (рис. 1.10). Элемент данных, созданный протоколом сетевого уровня, состоит из данных транспортного уровня и заголовка сетевого уровня и называется *дейтаграммой* (datagram).

Заголовок протокола сетевого уровня, как и заголовок протокола канального уровня, содержит поля с адресами исходной и целевой систем.

Однако в данном случае адрес целевой системы принадлежит конечному назначению пакета и может отличаться от адреса получателя в заголовке протокола канального уровня. Например, когда Вы вводите в адресной строке браузера адрес Web-узла, в пакете, сгенерированном Вашим компьютером, в качестве адреса целевой системы сетевого уровня указан адрес Web-сервера, тогда как на канальном уровне на целевую систему указывает адрес маршрутизатора в Вашей ЛВС, обеспечивающего выход в Интернет. В IP используется собственная система адресации, которая совершенно не зависит от адресов канального уровня. Каждому компьютеру в сети с протоколом IP вручную или автоматически назначается 32-битовый IP- адрес, идентифицирующий как сам компьютер, так и сеть, в которой он находится. В IPX же для идентификации самого компьютера используется аппаратный адрес, кроме того, специальный адрес используется для идентификации сети, в которой находится компьютер. В NetBEUI компьютеры различаются по NetBIOS-именам, присваиваемым каждой системе во время ее установки.

Дейтаграммам сетевого уровня на пути к месту назначения приходится проходить через множество сетей, сталкиваясь при этом со специфическими свойствами и ограничениями различных протоколов канального уровня. Одно из таких ограничений – максимальный размер пакета, разрешенный протоколом. Например, размер кадра Token Ring может достигать 4500 байт, тогда как размер кадров Ethernet не может превышать 1500 байтов. Когда большая дейтаграмма, сформированная в сети Token Ring, передается в сеть Ethernet, протокол сетевого уровня должен разбить ее на несколько фрагментов размером не более 1500 байт. Этот процесс называется *фрагментацией* (fragmentation). В процессе фрагментации протокол сетевого уровня разбивает дейтаграмму на фрагменты, размер которых соответствует возможностям используемого протокола канального уровня. Каждый фрагмент становится самостоятельным пакетом и продолжает путь к целевой системе сетевого уровня. Исходная дейтаграмма формируется лишь после того, как места назначения достигнут все фрагменты. Иногда на пути к целевой системе фрагменты, на которые разбита дейтаграмма, приходится фрагментировать повторно.

Маршрутизацией (routing) называется процесс выбора в сети самого эффективного маршрута для передачи дейтаграмм от системы-отправителя к системе-получателю. В сложных интерсетях, например, в Интернете или больших корпоративных сетях, часто от одного компьютера к другому можно добраться несколькими путями. Проектировщики сетей специально создают избыточные связи, чтобы трафик нашел дорогу к месту назначения даже в случае сбоя одного из маршрутизаторов. С помощью маршрутизаторов соединяют отдельные ЛВС, входящие в интерсеть. Назначение маршрутизатора — принимать входящий трафик от одной сети и передавать его конкретной системе в другой. В интерсетях различают системы двух видов: *оконечные* (end systems) и *промежуточные* (intermediate

systems). Оконечные системы являются отправителями и получателями пакетов. Маршрутизатор – промежуточная система. В оконечных системах используются все семь уровней модели OSI, тогда как пакеты, поступающие в промежуточные системы, не поднимаются выше сетевого уровня. Там маршрутизатор обрабатывает пакет и отправляет его вниз по стеку для передачи следующей целевой системе.

Чтобы верно направить пакет к цели, маршрутизаторы хранят в памяти таблицы с информацией о сети. Эта информация может быть внесена администратором вручную или собрана автоматически с других маршрутизаторов с помощью специализированных протоколов. В состав типичного элемента таблицы маршрутизации входят адрес другой сети и адрес маршрутизатора, через который пакеты должны добираться до этой сети. Кроме того, в элементе таблицы маршрутизации содержится *метрика маршрута* – условная оценка его эффективности. Если к некой системе имеется несколько маршрутов, маршрутизатор выбирает из них самый эффективный и отправляет дейтаграмму на канальный уровень для передачи маршрутизатору, указанному в элементе таблицы с наилучшей метрикой. В больших сетях маршрутизация может быть необычайно сложным процессом, но чаще всего она осуществляется автоматически и незаметно для пользователя.

Так же, как в заголовке канального уровня указан протокол сетевого уровня, сгенерировавший и передавший данные, в заголовке сетевого уровня содержится информация о протоколе транспортного уровня, от которого эти данные были получены. В соответствии с этой информацией система-получатель передает входящие дейтаграммы соответствующему протоколу транспортного уровня.

2.6. Транспортный уровень

Функции, выполняемые протоколами *транспортного* (transport) уровня, дополняют функции протоколов сетевого уровня. Часто протоколы этих уровней, используемые для передачи данных, образуют взаимосвязанную пару, что видно на примере TCP/IP: протокол TCP функционирует на транспортном уровне, IP – на сетевом. В большинстве наборов протоколов имеется два или несколько протоколов транспортного уровня, выполняющих разные функции. Альтернативой TCP является протокол UDP (User Datagram Protocol). В набор протоколов IPX также включено несколько протоколов транспортного уровня, в том числе NCP (NetWare Core Protocol) и SPX (Sequenced Packet Exchange). Разница между протоколами транспортного уровня из определенного набора заключается в том, что некоторые из них ориентированы на соединение, а другие – нет. Системы, использующие протокол, *ориентированный на соединение* (connection-oriented), перед передачей данных обмениваются сообщениями, чтобы ус-

тановить связь друг с другом. Это гарантирует, что системы включены и готовы к работе. Протокол TCP, например, ориентирован на соединение.

Когда Вы с помощью браузера подключаетесь к серверу Интернета, браузер и сервер для установления связи сначала выполняют так называемое *трехшаговое рукопожатие* (three-way handshake). Лишь после этого браузер передает серверу адрес нужной Web-страницы. Когда передача данных завершена, системы выполняют такое же рукопожатие для прекращения связи. Кроме того, протоколы, ориентированные на соединение, выполняют дополнительные действия, например, отправляют сигнал подтверждения приема пакета, сегментируют данные, управляют потоком, а также обнаруживают и исправляют ошибки. Как правило, протоколы этого типа используются для передачи больших объемов информации, в которых не должно содержаться ни единого ошибочного бита, например, файлов данных или программ. Дополнительные функции протоколов с ориентацией на соединение гарантируют корректную передачу данных. Вот почему эти протоколы часто называют *надежными* (reliable). Надежность в данном случае является техническим термином и означает, что каждый передаваемый пакет проверяется на наличие ошибок, кроме того, система-отправитель уведомляется о доставке каждого пакета.

Недостаток протоколов этого типа состоит в значительном объеме управляющих данных, которыми обмениваются две системы. Во-первых, дополнительные сообщения передаются при установлении и завершении связи. Во-вторых, заголовок, добавляемый к пакету протоколом с ориентацией на соединение, существенно превосходит по размеру заголовок протокола, не ориентированного на соединение. Например, заголовок протокола TCP/IP занимает 20 байт, а заголовок UDP – 8 байт. Протокол, *не ориентированный на соединение* (connectionless), не устанавливает соединение между двумя системами до передачи данных. Отправитель просто передает информацию целевой системе, не беспокоясь о том, готова ли она принять данные и существует ли эта система вообще. Обычно системы прибегают к протоколам, не ориентированным на соединение, например, к UDP, для коротких транзакций, состоящих только из запросов и ответных сигналов. Ответный сигнал от получателя неявно выполняет функцию сигнала подтверждения о передаче.

Примечание. Ориентированные и не ориентированные на соединение протоколы есть не только на транспортном уровне. Например, протоколы сетевого уровня обычно не ориентированы на соединение, по скольку обеспечение надежности связи они возлагают на транспортный уровень.

Протоколы транспортного уровня (как и сетевого и канального уровней) обычно содержат информацию с вышестоящих уровней. Например, в заголовки TCP и UDP включаются номера портов, идентифицирующие приложение, породившее пакет, и приложение, которому он предназначен.

2.7. Сеансовый уровень

На *сеансовом* (session) уровне начинается существенное расхождение между реально применяемыми протоколами и моделью OSI. В отличие от нижестоящих уровней, выделенных протоколов сеансового уровня не существует. Функции этого уровня интегрированы в протоколы, которые выполняют также функции представительского и прикладного уровней. Транспортный, сетевой, канальный и физический уровни занимаются собственно передачей данных по сети. Протоколы сеансового и вышестоящих уровней к процессу связи отношения не имеют. К сеансовому уровню относятся 22 службы, многие из которых задают способы обмена информацией между системами, включенными в сеть. Наиболее важны службы управления диалогом и разделения диалога. Обмен информацией между двумя системами в сети называется *диалогом* (dialog). *Управление диалогом* (dialog control) заключается в выборе режима, в котором системы будут обмениваться сообщениями. Таких режимов два: *полудуплексный* (two-way alternate, TWA) и *дуплексный* (two-way simultaneous, TWS). В полудуплексном режиме две системы вместе с данными передают также маркеры. Передавать информацию можно только компьютеру, у которого в данный момент находится маркер. Так удается избежать столкновения сообщений в пути. Дуплексная модель сложнее. Маркеров в ней нет; обе системы могут передавать данные в любой момент, даже одновременно. *Разделение диалога* (dialog separation) состоит во включении в поток данных *контрольных точек* (checkpoints), позволяющих синхронизировать работу двух систем. Степень сложности разделения диалога зависит от того, в каком режиме он осуществляется. В полудуплексном режиме системы выполняют малую синхронизацию, заключающуюся в обмене сообщениями о контрольных точках. В дуплексном режиме системы выполняют полную синхронизацию с помощью главного/активного маркера.

2.8. Представительский уровень

На *представительском* (presentation) уровне выполняется единственная функция: трансляция синтаксиса между различными системами. Иногда компьютеры в сети применяют разные синтаксисы. Представительский уровень позволяет им «договориться» об общем синтаксисе для обмена данными. Устанавливая соединение на представительском уровне, системы обмениваются сообщениями с информацией о том, какие синтаксисы в них имеются, и выбирают тот, который они будут использовать во время сеанса.

У обеих систем, участвующих в соединении, есть *абстрактный синтаксис* (abstract syntax) – их «родная» форма связи. Абстрактные синтаксисы различных компьютерных платформ могут отличаться. В процессе согласования системы выбирают общий *синтаксис передачи данных* (transfer syntax). Передающая система преобразует свой абстрактный син-

таксис в синтаксис передачи данных, а система-получатель по завершению передачи — наоборот. При необходимости система может выбрать синтаксис передачи данных с дополнительными функциями, например, сжатием или шифрованием данных.

2.9. Прикладной уровень

Прикладной уровень – это точка входа, через которую программы получают доступ к модели OSI и сетевым ресурсам. Большинство про токолов прикладного уровня предоставляет службы доступа к сети. Например, протоколом SMTP (Simple Mail Transfer Protocol) большинство программ электронной почты пользуется для отправки сообщений. Другие протоколы прикладного уровня, например, FTP (File Transfer Protocol), сами являются программами. В протоколы прикладного уровня часто включают функции сеансового и представительского уровня. В результате типичный стек протоколов содержит четыре отдельных протокола, которые работают на прикладном, транспортном, сетевом и канальном уровнях.

2.10. Адресация и маршрутизация в компьютерных сетях

Базовыми понятиями компьютерных сетей, относящимися к канальному и сетевому уровню ЭМВОС, являются понятия MAC-адреса, IP-адреса, маски подсети, а также протокола TCP/IP.

MAC-адрес – уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet.

IP-адрес – уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP.

В сети Интернет требуется глобальная уникальность адреса; в случае работы в локальной сети требуется уникальность адреса в пределах сети. В версии протокола IPv4IP-адрес имеет длину 4 байта, а в версии протокола IPv6 IP-адрес имеет длину 16 байт.

IPv4 (Интернет Протокол версии 4) является четвёртой версией Интернет Протокола (IP), и используется для идентификации устройств в сети через адресную систему, позволяя, так же, соединять устройства через глобальную сеть.

IPv4 использует 32-битную адресную схему, позволяя существование 2^{32} (более 4 миллиардов) адресов. При этом вместе с ростом Интернета ожидается, что количество неиспользуемых IPv4 адресов достаточно быстро закончится, так как каждое устройство, включая компьютеры, смартфоны и игровые консоли при подключении к Интернету требует для себя IP-адрес. Новый адресная система Интернет использующая Интернет-Протокол версии 6 (IPv6) разрабатывалась для того, чтобы полностью

удовлетворить возрастающую потребность в необходимом числе свободных интернет-адресов [6].

IPv6 (Интернет-протокол версии 6) также называемый IPng (Internet Protocol next generation – Интернет-протокол следующего поколения) – это обновлённая версия интернет-протокола (IP) созданная с учётом стандартов Инженерного Совета Интернета для замены текущей версии IPv4. IPv6 является наследником IPv4, и был задуман как революционное обновление существующей доныне версии Интернет Протокола, и в настоящее время сосуществует с более старым IPv4. Новый IPv6 создан чтобы обеспечить интернету устойчивый и надёжный рост, касающийся как номера наличных хостов, так и общего количества передаваемого траффика, поддерживая 2^{128} адресов – намного больше устаревшего протокола IPv4. IPv6 часто называют «следующей генерацией» стандартов Интернета, который постоянно развивается с середины 1990х до сегодняшнего дня. Он был рождён как ответ на тревоги о том, что количество требуемых IP-адресов скоро превысит граничные возможности сети Интернет.

После того, как мы узнали, что такое IPv6, рассмотрим дополнения, существующие в ней.

Вместе с увеличением количества возможных адресов, существуют и другие важные технологические изменения в IPv6 по сравнению с IPv4:

1. Нет необходимости в NAT (трансляции сетевых адресов).
2. Авто-конфигурация.
3. Нет коллизий приватных адресов.
4. Упрощённая, более эффективная маршрутизация.
5. Лучшая многоадресная маршрутизация.
6. Более простой формат заголовка.
7. Подтверждённое качество обслуживания (QoS), также называемое «маркировкой потока».
8. Встроенная аутентификация и поддержка конфиденциальности.

При этом, в IPv6 существуют несколько вариантов адресов:

1. Unicast (одноадресные) – используется в сервисах персонального характера, направляется из одного, определённого, источника к одному IP-адресу.

2. Anycast (групповые) – позволяет посылать данные ко всем абонентам определённой ip-сети.

3. Multicast (многоадресные) – данные передаются для неограниченного количества абонентов.

IP-адрес являет собой двоичное число, но он также может быть записан в более удобном для человека формате. Например, 32-битный числовой адрес, используемый в IPv4, может быть оформлен в десятичной системе 4 цифрами, причём каждое цифра может иметь значение от 0 до 255. Например, это могут быть цифры 172.16.254.1.

Адреса протокола IPv6 являются 128-битными, и оформлены в шестнадцатеричной системе. К примеру, адрес в IPv6 может быть записан как 3ffe:1904:4546:3:201:f8ff:fe22:68cf.

Маска подсети – битовая маска для определения по IP-адресу адреса подсети и адреса узла этой подсети. В отличие от IP-адреса маска подсети не является частью IP-пакета.

Благодаря маске можно узнать, какая часть IP-адреса узла сети относится к адресу сети, а какая к адресу самого узла в этой сети.

Transmission Control Protocol (TCP, протокол управления передачей) – один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. Сети и подсети, в которых совместно используются протоколы TCP и IP, называются сетями TCP/IP.

Маршрутизатором, или шлюзом, называется узел сети с несколькими IP-интерфейсами (содержащими свой MAC-адрес и IP-адрес), подключенными к разным IP-сетям, осуществляющий на основе решения задачи маршрутизации перенаправление дейтаграмм из одной сети в другую для доставки от отправителя к получателю.

Маршрутизаторы представляют собой либо специализированные вычислительные машины, либо компьютеры с несколькими IP-интерфейсами, работа которых управляется специальным программным обеспечением.

Маршрутизация служит для приема пакета от одного устройства и передачи его по сети другому устройству через другие сети [7]. Если в сети нет маршрутизаторов, то не поддерживается маршрутизация. Маршрутизаторы направляют (перенаправляют) трафик во все сети, составляющие объединенную сеть.

Для маршрутизации пакета маршрутизатор должен владеть следующей информацией:

- Адрес назначения
- Соседний маршрутизатор, от которого он может узнать об удаленных сетях
- Доступные пути ко всем удаленным сетям
- Наилучший путь к каждой удаленной сети
- Методы обслуживания и проверки информации о маршрутизации

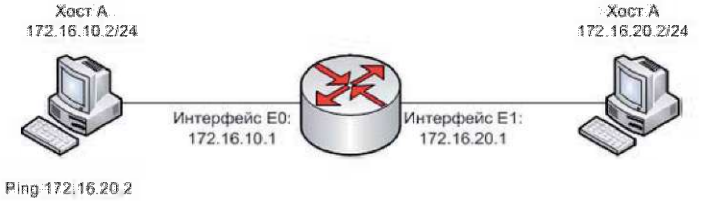
Маршрутизатор «узнаёт» об удаленных сетях от соседних маршрутизаторов или от сетевого администратора. Затем маршрутизатор строит таблицу маршрутизации, которая описывает, как найти удаленные сети.

Если сеть подключена непосредственно к маршрутизатору, он уже знает, как направить пакет в эту сеть. Если же сеть не подключена напрямую, маршрутизатор должен узнать (изучить) пути доступа к удаленной сети с помощью статической маршрутизации (ввод администратором вручную местоположения всех сетей в таблицу маршрутизации) или с помощью динамической маршрутизации.

Динамическая маршрутизация – это процесс протокола маршрутизации, определяющий взаимодействие устройства с соседними маршрутизаторами. Маршрутизатор будет обновлять сведения о каждой изученной им сети. Если в сети произойдет изменение, протокол динамической маршрутизации автоматически информирует об изменении все маршрутизаторы. Если же используется статическая маршрутизация, обновить таблицы маршрутизации на всех устройствах придется системному администратору.

IP-маршрутизация – простой процесс, который одинаков в сетях любого размера. Например, на рисунке показан процесс пошагового взаимодействия хоста A с хостом B в другой сети. В примере пользователь хоста A запрашивает по ping IP-адрес хоста B. Дальнейшие операции не так просты, поэтому рассмотрим их подробнее:

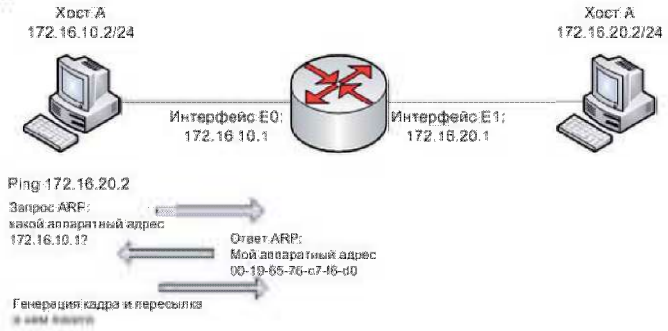
- В командной строке пользователь вводит ping 172.16.20.2. На хосте A генерируется пакет с помощью протоколов сетевого уровня IP и ICMP.



- IP обращается к протоколу ARP для выяснения сети назначения для пакета, просматривая IP-адрес и маску подсети хоста A. Это запрос к удаленному хосту, т.е. пакет не предназначен хосту локальной сети, поэтому пакет должен быть направлен маршрутизатору для перенаправления в нужную удаленную сеть.
- Чтобы хост A смог послать пакет маршрутизатору, хост должен знать аппаратный адрес интерфейса маршрутизатора, подключенный к локальной сети. Сетевой уровень передает пакет и аппаратный адрес назначения канальному уровню для деления на кадры и пересылки локальному хосту. Для получения аппаратного адреса хост ищет местоположение точки назначения в собственной памяти, называемой кэшем ARP.
- Если IP-адрес еще не был доступен и не присутствует в кэше ARP, хост посылает широковещательную рассылку ARP для поиска аппаратного адреса по IP-адресу 172.16.10.1. Именно поэтому первый запрос Ping обычно заканчивается тайм-аутом, но четыре остальные запроса будут успешны. После кэширования адреса тайм-аута обычно не возникает.
- Маршрутизатор отвечает и сообщает аппаратный адрес интерфейса Ethernet, подключенного к локальной сети. Теперь хост имеет всю информацию для пересылки пакета маршрутизатору по ло-

кальной сети. Сетевой уровень спускает пакет вниз для генерации эхо-запроса ICMP (Ping) на канальном уровне, дополняя пакет аппаратным адресом, по которому хост должен послать пакет. Пакет имеет IP-адреса источника и назначения вместе с указанием на тип пакета (ICMP) в поле протокола сетевого уровня.

- Канальный уровень формирует кадр, в котором инкапсулируется пакет вместе с управляющей информацией, необходимой для пересылки по локальной сети. К такой информации относятся аппаратные адреса источника и назначения, а также значение в поле типа, установленное протоколом сетевого уровня (это будет поле типа, поскольку IP по умолчанию пользуется кадрами Ethernet_II). Рисунок 3 показывает кадр, генерируемый на канальном уровне и пересылаемый по локальному носителю. На рисунке 3 показана вся информация, необходимая для взаимодействия с маршрутизатором: аппаратные адреса источника и назначения, IP-адреса источника и назначения, данные, а также контрольная сумма CRC кадра, находящаяся в поле FCS (Frame Check Sequence).
- Канальный уровень хоста A передает кадр физическому уровню. Там выполняется кодирование нулей и единиц в цифровой сигнал с последующей передачей этого сигнала по локальной физической сети.



- Сигнал достигает интерфейса Ethernet 0 маршрутизатора, который синхронизируется по преамбуле цифрового сигнала для извлечения кадра. Интерфейс маршрутизатора после построения кадра проверяет CRC, а в конце приема кадра сравнивает полученное значение с содержимым поля FCS. Кроме того, он проверяет процесс передачи на отсутствие фрагментации и конфликтов носителя.
- Проверяется аппаратный адрес назначения. Поскольку он совпадает с адресом маршрутизатора, анализируется поле типа кадра для определения дальнейших действий с этим пакетом данных. В поле типа указан протокол IP, поэтому маршрутизатор передает пакет процессу протокола IP, исполняемому маршрутизатором. Кадр

удаляется. Исходный пакет (сгенерированный хостом А) помещается в буфер маршрутизатора.

- Протокол IP смотрит на IP-адрес назначения в пакете, чтобы определить, не направлен ли пакет самому маршрутизатору. Поскольку IP-адрес назначения равен 172.16.20.2, маршрутизатор определяет по своей таблице маршрутизации, что сеть 172.16.20.0 непосредственно подключена к интерфейсу Ethernet 1.
- Маршрутизатор передает пакет из буфера в интерфейс Ethernet 1. Маршрутизатору необходимо сформировать кадр для пересылки пакета хосту назначения. Сначала маршрутизатор проверяет свой кэш ARP, чтобы определить, был ли уже разрешен аппаратный адрес во время предыдущих взаимодействий с данной сетью. Если адреса нет в кэше ARP, маршрутизатор посылает широковещательный запрос ARP в интерфейс Ethernet 1 для поиска аппаратного адреса для IP-адреса 172.16.20.2.
- Хост В откликается аппаратным адресом своего сетевого адаптера на запрос ARP. Интерфейс Ethernet 1 маршрутизатора теперь имеет все необходимое для пересылки пакета в точку окончательного приема. На рисунке показывает кадр, сгенерированный маршрутизатором и переданный по локальной физической сети.

Кадр, сгенерированный интерфейсом Ethernet 1 маршрутизатора, имеет аппаратный адрес источника от интерфейса Ethernet 1 и аппаратный адрес назначения для сетевого адаптера хоста В. Важно отметить, что, несмотря на изменения аппаратных адресов источника и назначения, в каждом передавшем пакет интерфейсе маршрутизатора, IP-адреса источника и назначения никогда не изменяются. Пакет никоим образом не модифицируется, но меняются кадры.

- Хост В принимает кадр и проверяет CRC. Если проверка будет успешной, кадр удаляется, а пакет передается протоколу IP. Он анализирует IP-адрес назначения. Поскольку IP-адрес назначения совпадает с установленным в хосте В адресе, протокол IP исследует поле протокола для определения цели пакета.

- В нашем пакете содержится эхо-запрос ICMP, поэтому хост B генерирует новый эхо-ответ ICMP с IP-адресом источника, равным адресу хоста B, и IP-адресом назначения, равным адресу хоста A. Процесс запускается заново, но в противоположном направлении. Однако аппаратные адреса всех устройств по пути следования пакета уже известны, поэтому все устройства смогут получить аппаратные адреса интерфейсов из собственных кэшей ARP.

В крупных сетях процесс происходит аналогично, но пакету придется пройти больше участков по пути к хосту назначения.

В стеке TCP/IP маршрутизаторы и конечные узлы принимают решения о том, кому передавать пакет для его успешной доставки узлу назначения, на основании так называемых таблиц маршрутизации (routing tables).

Таблица 1 представляет собой типичный пример таблицы маршрутов, использующей IP-адреса сетей, для сети, представленной на рисунке 4.

Таблица 1 – Пример таблицы маршрутов

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
129.13.0.0	255.255.0.0	-	129.13.0.1	подключен
198.21.17.0	255.255.255.0	-	198.21.17.6	подключен
213.34.12.0	255.255.255.0	198.21.17.1	198.21.17.6	1
56.0.0.0	255.0.0.0	198.21.17.7	198.21.17.6	1
116.0.0.0	255.0.0.0	198.21.17.7	198.21.17.6	2
116.0.0.0	255.0.0.0	198.21.17.1	198.21.17.6	2
0.0.0.0	0.0.0.0	198.21.17.7	198.21.17.6	-

В таблице 1 представлена таблица маршрутизации многомаршрутная, так как содержится два маршрута до сети 116.0.0.0. В случае построения одномаршрутной таблицы маршрутизации, необходимо указывать только один путь до сети 116.0.0.0 по наименьшему значению метрики.

Как нетрудно видеть, в таблице определено несколько маршрутов с разными параметрами. Читать каждую такую запись в таблице маршрутизации нужно следующим образом:

Чтобы доставить пакет в сеть с адресом из поля Сетевой адрес и маской из поля Маска сети, нужно с интерфейса с IP-адресом из поля Интерфейс послать пакет по IP-адресу из поля Адрес шлюза, а «стоимость» такой доставки будет равна числу из поля Метрика.

В этой таблице в столбце "Адрес сети назначения" указываются адреса всех сетей, которым данный маршрутизатор может передавать пакеты. В стеке TCP/IP принят так называемый одношаговый подход к оптимизации маршрута продвижения пакета (next-hop routing) – каждый маршрутизатор и конечный узел принимает участие в выборе только одного шага передачи пакета. Поэтому в каждой строке таблицы маршрутизации указывается не весь маршрут в виде последовательности IP-адресов маршрутизаторов, через которые должен пройти пакет, а только один IP-адрес – адрес следующего маршрутизатора, которому нужно передать пакет. Вместе с пакетом следующему маршрутизатору передается ответственность за выбор следующего шага маршрутизации. Одношаговый подход к маршру-

тизации означает распределенное решение задачи выбора маршрута. Это снимает ограничение на максимальное количество транзитных маршрутизаторов на пути пакета.

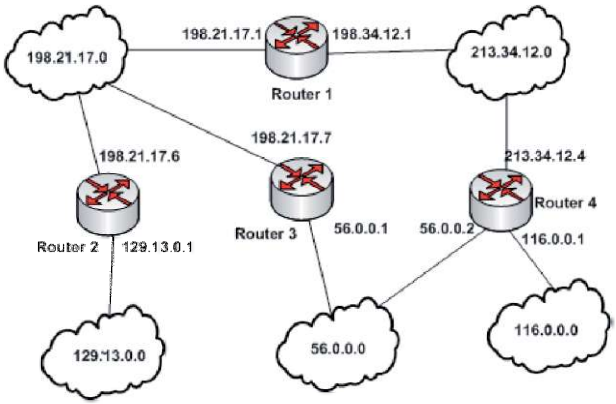


Рисунок 4 – Схема маршрутизации

Для отправки пакета следующему маршрутизатору требуется знание его локального адреса, но в стеке TCP/IP в таблицах маршрутизации принято использование только IP-адресов для сохранения их универсального формата, не зависящего от типа сетей, входящих в интерсеть. Для нахождения локального адреса по известному IP-адресу необходимо воспользоваться протоколом ARP.

Одношаговая маршрутизация обладает еще одним преимуществом – она позволяет сократить объем таблиц маршрутизации в конечных узлах и маршрутизаторах за счет использования в качестве номера сети назначения так называемого маршрута по умолчанию – default (0.0.0.0), который обычно занимает в таблице маршрутизации последнюю строку. Если в таблице маршрутизации есть такая запись, то все пакеты с номерами сетей, которые отсутствуют в таблице маршрутизации, передаются маршрутизатору, указанному в строке default. Поэтому маршрутизаторы часто хранят в своих таблицах ограниченную информацию о сетях интерсети, пересылая пакеты для остальных сетей в порт и маршрутизатор, используемые по умолчанию. Подразумевается, что маршрутизатор, используемый по умолчанию, передаст пакет на магистральную сеть, а маршрутизаторы, подключенные к магистрали, имеют полную информацию о составе интерсети.

Кроме маршрута default, в таблице маршрутизации могут встретиться два типа специальных записей – запись о специфичном для узла маршруте и запись об адресах сетей, непосредственно подключенных к портам маршрутизатора.

Специфичный для узла маршрут содержит вместо номера сети полный IP-адрес, то есть адрес, имеющий ненулевую информацию не только в поле номера сети, но и в поле номера узла. Предполагается, что для такого конечного узла маршрут должен выбираться не так, как для всех остальных узлов сети, к которой он относится. В случае, когда в таблице есть разные записи о продвижении пакетов для всей сети N и ее отдельного узла, имеющего адрес N,D, при поступлении пакета, адресованного узлу N,D, маршрутизатор отдаст предпочтение записи для N,D.

Записи в таблице маршрутизации, относящиеся к сетям, непосредственно подключенным к маршрутизатору, в поле "Метрика" содержат нули («подключено»).

В общем виде к алгоритмам маршрутизации предъявляются следующие требования:

- точность;
- простота;
- надёжность;
- стабильность;
- справедливость;
- оптимальность.

Существуют различные алгоритмы построения таблиц для одношаговой маршрутизации. Их можно разделить на три класса:

- алгоритмы простой маршрутизации;
- алгоритмы фиксированной маршрутизации;
- алгоритмы адаптивной маршрутизации.

Независимо от алгоритма, используемого для построения таблицы маршрутизации, результат их работы имеет единый формат. За счет этого в одной и той же сети различные узлы могут строить таблицы маршрутизации по своим алгоритмам, а затем обмениваться между собой недостающими данными, так как форматы этих таблиц фиксированы. Поэтому маршрутизатор, работающий по алгоритму адаптивной маршрутизации, может снабдить конечный узел, применяющий алгоритм фиксированной маршрутизации, сведениями о пути к сети, о которой конечный узел ничего не знает.

Простая маршрутизация – это способ маршрутизации, не изменяющийся при изменении топологии и состоянии сети передачи данных (СПД).

Простая маршрутизация обеспечивается различными алгоритмами, типичными из которых являются следующие:

- Случайная маршрутизация – это передача сообщения из узла в любом случайно выбранном направлении, за исключением направлений, по которым сообщение поступило узел.
- Лавинная маршрутизация – это передача сообщения из узла во всех направлениях, кроме направления, по которому сообщение поступило в

узел. Такая маршрутизация гарантирует малое время доставки пакета, за-счет ухудшения пропускной способности.

• Маршрутизация по предыдущему опыту – каждый пакет имеет счетчик числа пройденных узлов, в каждом узле связи анализируется счётчик, и запоминается тот маршрут, который соответствует минимальному значению счетчика. Такой алгоритм позволяет приспосабливаться к изменению топологии сети, но процесс адаптации протекает медленно и неэффективно.

В целом, простая маршрутизация не обеспечивает направленную передачу пакета и имеет низкую эффективности. Основным ее достоинством является обеспечение устойчивой работы сети при выходе из строя различных частей сети.

Фиксированная маршрутизация применяется в сетях с простой топологией связей и основан на ручном составлении таблицы маршрутизации администратором сети. Алгоритм часто эффективно работает также для магистралей крупных сетей, так как сама магистраль может иметь простую структуру с очевидными наилучшими путями следования пакетов в подсети, присоединенные к магистрали, выделяют следующие алгоритмы:

• Однопутевая фиксированная маршрутизация – это когда между двумя абонентами устанавливается единственный путь. Сеть с такой маршрутизацией неустойчива к отказам и перегрузкам.

• Многопутевая фиксированная маршрутизация – может быть установлено несколько возможных путей и вводится правило выбора пути. Эффективность такой маршрутизации падает при увеличении нагрузки. При отказе какой-либо линии связи необходимо менять таблицу маршрутизации, для этого в каждом узле связи храниться несколько таблиц.

Адаптивная маршрутизация – это основной вид алгоритмов маршрутизации, применяющихся маршрутизаторами в современных сетях со сложной топологией. Адаптивная маршрутизация основана на том, что маршрутизаторы периодически обмениваются специальной топологической информацией об имеющихся в интерсети сетях, а также о связях между маршрутизаторами. Обычно учитывается не только топология связей, но и их пропускная способность и состояние.

Адаптивные протоколы позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. Эти протоколы имеют распределенный характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы, которые бы собирали и обобщали топологическую информацию: эта работа распределена между всеми маршрутизаторами, выделяют следующие алгоритмы:

• Локальная адаптивная маршрутизация – каждый узел содержит информацию о состоянии линии связи, длины очереди и таблицу маршрутизации.

• Глобальная адаптивная маршрутизация – основана на использовании информации, получаемой от соседних узлов. Для этого каждый узел содержит таблицу маршрутизации, в которой указано время прохождения сообщений. На основе информации, получаемой из соседних узлов, значение таблицы пересчитывается с учетом длины очереди в самом узле.

• Централизованная адаптивная маршрутизация – существует некоторый центральный узел, который занимается сбором информации о состоянии сети. Этот центр формирует управляющие пакеты, содержащие таблицы маршрутизации, и рассылает их в узлы связи.

• Гибридная адаптивная маршрутизация – основана на использовании таблицы, периодически рассылаемой центром, и на анализе длины очереди в самом узле.

Маршрутные таблицы содержат информацию, которую используют программы коммутации для выбора наилучшего маршрута. Чем характеризуется построение маршрутных таблиц? Какова особенность природы информации, которую они содержат? В данном разделе, посвященном показателям алгоритмов, сделана попытка ответить на вопрос о том, каким образом алгоритм определяет предпочтительность одного маршрута по сравнению с другими.

В алгоритмах маршрутизации используется множество различных показателей. Сложные алгоритмы маршрутизации при выборе маршрута могут базироваться на множестве показателей, комбинируя их таким образом, что в результате получается один гибридный показатель. Ниже перечислены показатели, которые используются в алгоритмах маршрутизации:

- Длина маршрута.
- Надежность.
- Задержка.
- Ширина полосы пропускания.

Длина маршрута является наиболее общим показателем маршрутизации. Некоторые протоколы маршрутизации позволяют администраторам сети назначать произвольные цены на каждый канал сети. В этом случае длиной тракта является сумма расходов, связанных с каждым каналом, который был траверсирован. Другие протоколы маршрутизации определяют "количество пересылок" (количество хопов), т. е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через элементы объединения сетей (такие как маршрутизаторы).

Надежность, в контексте алгоритмов маршрутизации, относится к надежности каждого канала сети (обычно описываемой в терминах соотношения бит/ошибка). Некоторые каналы сети могут отказывать чаще, чем другие. Отказы одних каналов сети могут быть устранены легче или быстрее, чем отказы других каналов. При назначении оценок надежности могут быть приняты в расчет любые факторы надежности. Оценки надежности

обычно назначаются каналам сети администраторами. Как правило, это произвольные цифровые величины.

Под задержкой маршрутизации обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порт каждого маршрутизатора на пути передвижения пакета, перегруженность сети на всех промежуточных каналах сети и физическое расстояние, на которое необходимо переместить пакет. Т. к. здесь имеет место конгломерация нескольких важных переменных, задержка является наиболее общим и полезным показателем.

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. При прочих равных показателях, канал Ethernet 10 Mbps предпочтителен любой арендованной линии с полосой пропускания 64 Кбайт/с. Хотя полоса пропускания является оценкой максимально достижимой пропускной способности канала, маршруты, проходящие через каналы с большей полосой пропускания, не обязательно будут лучше маршрутов, проходящих через менее быстродействующие каналы.

2.11. Стандарты интернета

Своим существованием Интернет обязан открытым стандартам. Под открытостью будем понимать равную доступность стандарта различным группам заинтересованных лиц (разработчикам программного и аппаратного обеспечения, пользователям, международным организациям по стандартизации и т.д.).

Созданием стандартов в интернете занимаются различные международные организации, причем каждая из них специализируется на своих направлениях (сетевые протоколы, доменные имена, стандарты языков представления данных и т.д.). Далеко не все стандарты интернета имеют привычную форму строгих документов наподобие стандартов ISO, IEC, DIN или ГОСТ. Стандарты интернета изменяются так же, как и сам интернет, в котором появляются новые технологии, новые устройства, и это постоянное обновление приводит к новым техническим требованиям и обновлению стандартов.

Сетевой протокол – набор правил и действий (очередности действий), позволяющий осуществлять соединение и обмен данными между двумя и более включенными в сеть устройствами.

Сетевые протоколы содержат:

- структуру или формат сообщения;
- методы обмена информацией о маршрутах сетей между собой;
- форму передачи устройствами сообщений об ошибках и системных сообщений;

- информацию о способе и времени завершения передачи данных по сети.

Основой успеха сети Интернет стала разработка *стека протоколов TCP/IP* (*Transmission Control Protocol/Internet Protocol*), входящего в семиуровневую модель ISO/OSI.

TCP/IP – это стек протоколов передачи данных, используемый в сетях, включая сеть Интернет. Название протокола TCP/IP происходит из названий двух протоколов – *Transmission Control Protocol* (TCP) и *Internet Protocol* (IP). Оба указанных протокола являются главными протоколами семейства. Они были разработаны первыми и описаны в данном стандарте.

TCP (*Transmission Control Protocol)* – протокол управления передачей данных, это один из основных протоколов передачи интернета, который *работает на транспортном уровне* модели OSI.

Этот протокол устанавливает логическое соединение между отправителем и получателем, а также обеспечивает связь между этими узлами, сохраняя порядок потока пакетов. Протокол гарантирует доставку информации, осуществляя контроль ее целостности.

IP (*Internet Protocol)* – межсетевой протокол – это маршрутизируемый протокол сетевого уровня из стека TCP/IP. Главной частью протокола является адресация сети. Протокол IP объединяет отдельные сегменты сети в единую сеть и обеспечивает доставку пакетов адресату. Главная задача протокола IP – маршрутизация пакетов. Он не отвечает за сохранение порядка потока пакетов (могут прийти несколько копий одного пакета), за надежность доставки информации, ее целостность. Безошибочную доставку гарантирует протокол транспортного уровня TCP.

Для обмена данными между приложениями или процессами используются протоколы приложений. Рассмотрим наиболее часто используемые протоколы этого уровня:

- *SMTP (Simple Mail Transport Protocol*), *POP (Post Office Protocol*) – протоколы электронной почты;
- *SNMP (Simple Network Management Protocol)* – протокол управления сетями;
- *FTP (File Transport Protocol)* – протокол, предназначенный для передачи файлов;
- *Telnet* (*Terminal Network*) – сетевой протокол удаленного доступа. Чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом, необходим данный протокол. Этот протокол может быть использован как для связи вида «терминал — терминал», так и для связи «процесс — процесс» в распределенных вычислениях;
- *SIP* (*Session Initiation Protocol)* – протокол установления сеанса, один из протоколов, который лежит в основе технологии VoIP (*Voice over IP) ;*

• *HTTP (Hyper Text Transfer Protocol)* – протокол уровня приложения, предназначен для передачи гипертекста, управляет процессом взаимодействия веб-сервера и веб-клиента.

Существуют и другие протоколы. Уровень безопасности, например, обеспечивают:

• *SSL (Secure Socket Layer)* – сетевой протокол с шифрованием для безопасной передачи данных;

• *SET (Security Electronics Transaction)* – стандарт безопасных транзакций в сети Интернет. Этот протокол является открытым стандартным протоколом для безопасных платежей с использованием пластиковых карт.

Общее число протоколов насчитывает не один десяток, поэтому в данном разделе приведены наиболее важные из них.

2.12. Модели безопасности в вычислительных сетях. Рабочая группа и домен

Современные сети часто состоят из множества различных программных платформ, большого разнообразия оборудования и программного обеспечения. Пользователи зачастую вынуждены запоминать большое количество паролей для доступа к различным сетевым ресурсам. Права доступа могут быть различными для одного и того же сотрудника в зависимости от того, с какими ресурсами он работает. Всё это множество взаимосвязей требует от администратора и пользователя огромного количества времени на анализ, запоминание и обучение.

Решение проблемы управления такой разнородной сетью было найдено с разработкой службы каталога. Службы каталога предоставляют возможности управления любыми ресурсами и сервисами из любой точки независимо от размеров сети, используемых операционных систем и сложности оборудования. Информация о пользователе, заносится единожды в службу каталога, и после этого становится доступной в пределах всей сети. Адреса электронной почты, принадлежность к группам, необходимые права доступа и учетные записи для работы с различными операционными системами – всё это создается и поддерживается в актуальном виде автоматически. Любые изменения, занесенные в службу каталога администратором, сразу обновляются по всей сети. Администраторам уже не нужно беспокоиться об уволенных сотрудниках – просто удалив учётную запись пользователя из службы каталога, он сможет гарантировать автоматическое удаление всех прав доступа на ресурсы сети, предоставленные ранее этому сотруднику.

В настоящее время большинство служб каталогов различных фирм базируются на стандарте *X.500*. Для доступа к информации, хранящейся в службах каталогов, обычно используется протокол *Lightweight Directory Access Protocol (LDAP)*. В связи со стремительным развитием сетей

TCP/IP, протокол LDAP становится стандартом для служб каталогов и приложений, ориентированных на использование службы каталога [8].

Служба каталогов Active Directory является основой логической структуры корпоративных сетей, базирующихся на системе Windows. Термин "*Каталог*" в самом широком смысле означает "*Справочник*", а служба каталогов корпоративной сети – это централизованный корпоративный справочник. Корпоративный каталог может содержать информацию об объектах различных типов. Служба каталогов Active Directory содержит в первую очередь объекты, на которых базируется система безопасности сетей Windows, – учётные записи пользователей, групп и компьютеров. Учётные записи организованы в логические структуры: домен, дерево, лес, организационные подразделения.

Основа сетевой безопасности – база данных учётных записей (accounts) пользователей, групп пользователей и компьютеров, с помощью которой осуществляется управление доступом к сетевым ресурсам. Прежде чем говорить о службе каталогов Active Directory, сравним две модели построения базы данных служб каталогов и управления доступом к ресурсам.

Данная модель управления безопасностью корпоративной сети «Рабочая группа» – самая примитивная. Она предназначена для использования в небольших одноранговых сетях (3 – 10 компьютеров) и основана на том, что каждый компьютер в сети с операционными системами Windows имеет свою собственную локальную базу данных учетных записей и с помощью этой локальной БД осуществляется управление доступом к ресурсам данного компьютера. Локальная БД учетных записей называется база данных *SAM* (*Security Account Manager*) и хранится в реестре операционной системы. Базы данных отдельных компьютеров полностью изолированы друг от друга и никак не связаны между собой. Пример управления доступом при использовании такой модели изображен на рис. 5.

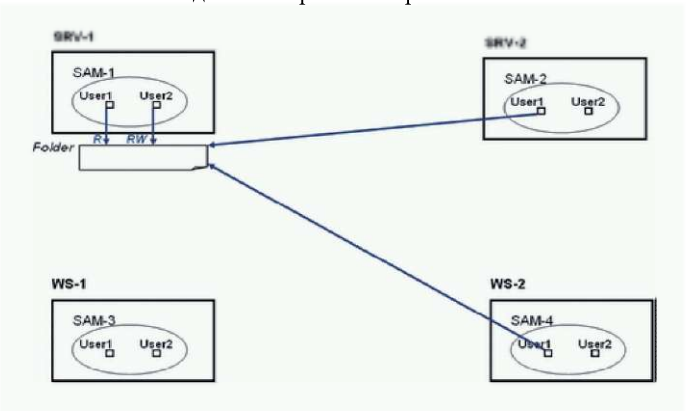


Рисунок 5 – Рабочая группа

В данном примере изображены два сервера (SRV-1 и SRV-2) и две рабочие станции (WS-1 и WS-2). Их базы данных SAM обозначены соответственно SAM-1, SAM-2, SAM-3 и SAM-4 (на рисунке базы SAM изображены в виде овала). В каждой БД есть учетные записи пользователей User1 и User2. Полное имя пользователя User1 на сервере SRV-1 будет выглядеть как "SRV-1\User1", а полное имя пользователя User1 на рабочей станции WS-1 будет выглядеть как "WS-1\User1". Представим, что на сервере SRV-1 создана папка Folder, к которой предоставлен доступ по сети пользователям User1 – на чтение (R), User2 – чтение и запись (RW). Главный момент в этой модели заключается в том, что компьютер SRV-1 ничего "не знает" об учетных записях компьютеров SRV-2, WS-1, WS-2, а также всех остальных компьютеров сети. Если пользователь с именем User1локально зарегистрируется в системе на компьютере, например, WS-2 (или, как еще говорят, "войдет в систему с локальным именем User1 на компьютере WS-2"), то при попытке получить доступ с этого компьютера по сети к папке Folder на сервере SRV-1 сервер запросит пользователя ввести имя и пароль (исключение составляет тот случай, если у пользователей с одинаковыми именами одинаковые пароли).

Модель "Рабочая группа" более проста для изучения, здесь нет необходимости изучать сложные понятия Active Directory. Но при использовании в сети с большим количеством компьютеров и сетевых ресурсов становится очень сложным управлять именами пользователей и их паролями – приходится на каждом компьютере (который предоставляет свои ресурсы для совместного использования в сети) вручную создавать одни и те же учётные записи с одинаковыми паролями, что очень трудоемко, либо делать одну учетную запись на всех пользователей с одним на всех паролем (или вообще без пароля), что сильно снижает уровень защиты информации. Поэтому модель "Рабочая группа" рекомендуется только для сетей с числом компьютеров от 3 до 10 (а еще лучше – не более 5), при условии, что среди всех компьютеров нет ни одного с системой Windows Server.

В доменной модели существует единая база данных служб каталогов, доступная всем компьютерам сети. Для этого в сети устанавливаются специализированные серверы, называемые *контроллерами домена*, которые хранят на своих жестких дисках эту базу. На рис. 6. изображена схема доменной модели. Серверы DC-1 и DC-2 – контроллеры домена, они хранят доменную базу данных учетных записей (каждый контроллер хранит у себя свою собственную копию БД, но все изменения, производимые в БД на одном из серверов, реплицируются на остальные контроллеры).

В такой модели, если, например, на сервере SRV-1, являющемся членом домена, предоставлен общий доступ к папке Folder, то права доступа к данному ресурсу можно назначать не только для учетных записей локальной базы SAM данного сервера, но, самое главное, учетным записям, хранящимся в доменной БД. На рисунке для доступа к папке Folder даны права доступа одной локальной учетной записи компьютера SRV-1 и

нескольким учетным записям домена (пользователя и группам пользователей). В доменной модели управления безопасностью пользователь регистрируется на компьютере ("входит в систему") со своей *доменной учетной записью* и, независимо от компьютера, на котором была выполнена регистрация, получает доступ к необходимым сетевым ресурсам. И нет необходимости на каждом компьютере создавать большое количество локальных учетных записей, все записи созданы *однократно в доменной БД*. И с помощью доменной базы данных осуществляется *централизованное управление доступом* к сетевым ресурсам *независимо от количества компьютеров в сети*.

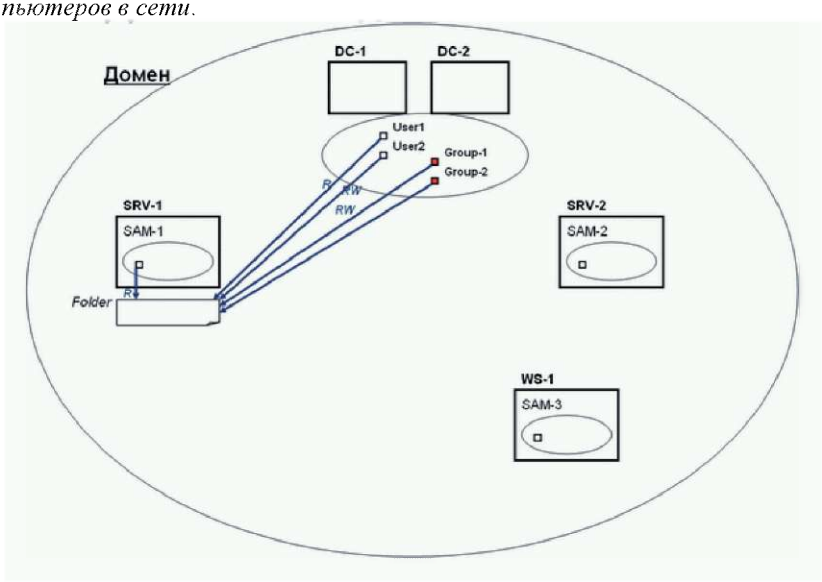


Рис. 6 – Доменная модель

Итак, модель «Рабочая группа» не представляет интереса с точки зрения основ системного администрирования, так как не требует рассмотрения сложных понятий управления сетью, поэтому в следующей главе речь пойдёт о доменной модели безопасности.

Контрольные вопросы

1. Что называется активным сетевым оборудованием?
2. Что называется пассивным сетевым оборудованием?
3. Приведите примеры активного сетевого оборудования.
4. Приведите примеры пассивного сетевого оборудования.
5. В чём заключается топология «шина»?
6. В чём заключается топология «звезда»?

7. В чём заключается топология «кольцо»?
8. Перечислите достоинства и недостатки топологии «шина».
9. Перечислите достоинства и недостатки топологии «звезда».
10. Перечислите достоинства и недостатки топологии «кольцо».
11. Что такое ЭМВОС?
12. С какого уровня ЭВМОС начинается работа браузера в сети?
13. Какие действия происходят на физическом уровне?
14. Какие действия происходят на канальном уровне?
15. Какие действия происходят на сетевом уровне?
16. Какие действия происходят на транспортном уровне?
17. Какие действия происходят на сеансовом уровне?
18. Какие действия происходят на представительском уровне?
19. Какие действия происходят на прикладном уровне?
20. Что является блоком данных на канальном уровне?
21. Что является блоком данных на сетевом уровне?
22. Что такое MAC-адрес?
23. Что такое IP-адрес?
24. Что такое маска подсети?
25. Что такое TCP/IP?
26. В чём отличие протокола IPv6 от протокола IPv4?
27. Что такое маршрутизация?
28. Какие требования предъявляются к маршрутизации?
29. Что такое простая маршрутизация?
30. Что такое фиксированная маршрутизация?
31. Что такое адаптивная маршрутизация?
32. Какие бывают виды адаптивной маршрутизации?
33. Что такое пропускная способность?
34. Что такое динамическая маршрутизация?
35. Что такое статическая маршрутизация?
36. Что такое сетевой протокол?
37. Что содержат сетевые протоколы?
38. Что такое FTP?
39. Что такое SSL?
40. Что такое SET?
41. Какие модели безопасности используются в компьютерных сетях?
42. Для каких сетей применяется рабочая группа?
43. Какие сети организуются в домен?
44. Чем отличается домен от рабочей группы?
45. Что такое SAM?
46. Что такое LDAP?

3. ДОМЕННАЯ МОДЕЛЬ БЕЗОПАСНОСТИ В ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

3.1. Понятие службы каталогов, её назначение, задачи, основные положения

Каталог (справочник) может хранить различную информацию, относящуюся к пользователям, группам, компьютерам, сетевым принтерам, общим файловым ресурсам и так далее – будем называть вс` это объектами. Каталог хранит также информацию о самом объекте, или его свойства, называемые атрибутами. Например, атрибутами, хранимыми в каталоге о пользователе, может быть имя его руководителя, номер телефона, адрес, имя для входа в систему, пароль, группы, в которые он входит, и многое другое. Для того чтобы сделать хранилище каталога полезным для пользователей, должны существовать службы, которые будут взаимодействовать с каталогом. Например, можно использовать каталог как хранилище информации, по которой можно аутентифицировать пользователя, или как место, куда можно послать запрос для того, чтобы найти информацию об объекте.

Active Directory отвечает не только за создание и организацию этих небольших объектов, но также и за большие объекты, такие как домены, OU (организационные единицы или подразделения) и сайты.

Служба каталогов Active Directory (сокращенно AD) обеспечивает эффективную работу сложной корпоративной среды, предоставляя следующие возможности:

- *Единая регистрация в сети.* Пользователи могут регистрироваться в сети с одним именем и паролем и получать при этом доступ ко всем сетевым ресурсам и службам (службы сетевой инфраструктуры, службы файлов и печати, серверы приложений и баз данных и т. д.);
- *Безопасность информации.* Средства аутентификации и управления доступом к ресурсам, встроенные в службу Active Directory, обеспечивают централизованную защиту сети;
- *Централизованное управление.* Администраторы могут централизованно управлять всеми корпоративными ресурсами;
- *Администрирование с использованием групповых политик.* При загрузке компьютера или регистрации пользователя в системе выполняются требования групповых политик; их настройки хранятся в объектах групповых политик (GPO) и применяются ко всем учетным записям пользователей и компьютеров, расположенных в сайтах, доменах или организационных подразделениях;
- *Интеграция с DNS.* Функционирование служб каталогов полностью зависит от работы службы DNS. В свою очередь серверы DNS могут хранить информацию о зонах в базе данных Active Directory;

• *Расширяемость каталога.* Администраторы могут добавлять в схему каталога новые классы объектов или добавлять новые атрибуты к существующим классам;

• *Масштабируемость.* Служба Active Directory может охватывать как один домен, так и множество доменов, объединенных в дерево доменов, а из нескольких деревьев доменов может быть построен лес;

• *Репликация (копирование) информации.* В службе Active Directory используется репликация служебной информации в схеме со многими ведущими (*multi-master*), что позволяет модифицировать БД Active Directory на любом контроллере домена. Наличие в домене нескольких контроллеров обеспечивает отказоустойчивость и возможность распределения сетевой нагрузки;

• *Гибкость запросов к каталогу.* БД Active Directory может использоваться для быстрого поиска любого объекта AD, используя его свойства (например, имя пользователя или адрес его электронной почты, тип принтера или его местоположение и т. п.);

• *Стандартные интерфейсы программирования.* Для разработчиков программного обеспечения служба каталогов предоставляет доступ ко всем возможностям (средствам) каталога и поддерживает принятые стандарты и интерфейсы программирования (API).

В Active Directory может быть создан широкий круг различных объектов. Объект представляет собой уникальную сущность внутри Каталога и обычно обладает многими атрибутами, которые помогают описывать и распознавать его. Учетная запись пользователя является примером объекта. Этот тип объекта может иметь множество атрибутов, таких как имя, фамилия, пароль, номер телефона, адрес и многие другие. Таким же образом общий принтер тоже может быть объектом в Active Directory и его атрибутами являются его имя, местоположение и т.д. Атрибуты объекта не только помогают определить объект, но также позволяют вам искать объекты внутри Каталога.

Служба каталогов системы Windows Server построена на общепринятых технологических стандартах. Изначально для служб каталогов был разработан стандарт *X.500*, который предназначался для построения иерархических древовидных масштабируемых справочников с возможностью расширения как классов объектов, так и наборов атрибутов (свойств) каждого отдельного класса. Однако практическая реализация этого стандарта оказалась неэффективной с точки зрения производительности. Тогда на базе стандарта X.500 была разработана упрощенная (облегченная) версия стандарта построения каталогов, получившая название *LDAP* (*Lightweight Directory Access Protocol*). Протокол LDAP сохраняет все основные свойства X.500 (иерархическая система построения справочника, масштабируемость, расширяемость), но при этом позволяет достаточно эффективно реализовать данный стандарт на практике. Термин "*lightweight*" ("*облегченный*") в названии LDAP отражает основную цель

разработки протокола: создать инструментарий для построения службы каталогов, которая обладает достаточной функциональной мощью для решения базовых задач, но не перегружена сложными технологиями, делающими реализацию служб каталогов неэффективной. В настоящее время LDAP является стандартным методом доступа к информации сетевых каталогов и играет роль фундамента во множестве продуктов, таких как системы аутентификации, почтовые программы и приложения электронной коммерции. Сегодня на рынке присутствует более 60 коммерческих серверов LDAP, причем около 90% из них представляют собой самостоятельные серверы каталогов LDAP, а остальные предлагаются в качестве компонентов других приложений.

Протокол LDAP четко определяет круг операций над каталогами, которые может выполнять клиентское приложение. Эти операции распадаются на пять групп:

- установление связи с каталогом;
- поиск в нем информации;
- модификация его содержимого;
- добавление объекта;
- удаление объекта.

Кроме протокола LDAP служба каталогов Active Directory использует также протокол аутентификации *Kerberos* и службу DNS для поиска в сети компонент служб каталогов (контроллеры доменов, серверы глобального каталога, службу Kerberos и др.).

3.2. Домен: понятие, физическая и логическая организация

Основной единицей системы безопасности Active Directory является *домен*. Домен формирует область административной ответственности. База данных домена содержит учетные записи *пользователей*, *групп* и *компьютеров*. Большая часть функций по управлению службой каталогов работает на уровне домена (аутентификация пользователей, управление доступом к ресурсам, управление службами, управление репликацией, политики безопасности).

Имена доменов Active Directory формируются по той же схеме, что и имена в пространстве имен DNS. И это не случайно. Служба DNS является средством поиска компонент домена – в первую очередь контроллеров домена.

Контроллеры домена – специальные серверы, которые хранят соответствующую данному домену часть базы данных Active Directory. Основные функции контроллеров домена:

- хранение БД Active Directory (организация доступа к информации, содержащейся в каталоге, включая управление этой информацией и ее модификацию);

- синхронизация изменений в AD (изменения в базу данных AD могут быть внесены на любом из контроллеров домена, любые изменения, осуществляемые на одном из контроллеров, будут синхронизированы с копиями, хранящимися на других контроллерах);
- аутентификация пользователей (любой из контроллеров домена осуществляет проверку полномочий пользователей, регистрирующихся на клиентских системах).

Настоятельно рекомендуется в каждом домене устанавливать не менее двух контроллеров домена. Во-первых, для защиты от потери БД Active Directory в случае выхода из строя какого-либо контроллера, во-вторых, для распределения нагрузки между контроллерами.

Дерево является набором доменов, которые используют единое связанное пространство имен. В этом случае "дочерний" домен наследует свое имя от "родительского" домена. Дочерний домен автоматически устанавливает двухсторонние транзитивные доверительные отношения с родительским доменом. Доверительные отношения означают, что ресурсы одного из доменов могут быть доступны пользователям других доменов.

Пример деревьев Active Directory изображен на рис. 7. В данном примере домен company.ru является доменом Active Directory верхнего уровня. От корневого домена отходят дочерние домены it.company.ru и fin.company.ru. Эти домены могут относиться соответственно к ИТ-службе компании и финансовой службе. У домена it.company.ru есть поддомен dev.it.company.ru, созданный для отдела разработчиков ПО ИТ-службы.

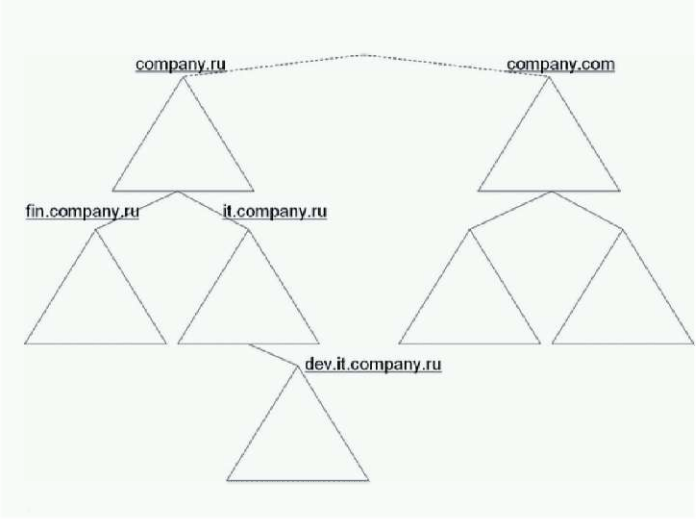


Рисунок 7 – Деревья Active Directory

Корпорация Microsoft рекомендует строить Active Directory в виде одного домена. Построение дерева, состоящего из многих доменов необходимо в следующих случаях:

- для децентрализации администрирования служб каталогов (например, в случае, когда компания имеет филиалы, географически удаленные друг от друга, и централизованное управление затруднено по техническим причинам);
- для повышения производительности (для компаний с большим количеством пользователей и серверов актуален вопрос повышения производительности работы контроллеров домена);
- для более эффективного управления репликацией (если контроллеры доменов удалены друг от друга, то репликация в одном может потребовать больше времени и создавать проблемы с использованием несинхронизированных данных);
- для применения различных политик безопасности для различных подразделений компании;
- при большом количестве объектов в БД Active Directory.

Наиболее крупная структура в Active Directory – лес. Лес объединяет деревья, которые поддерживают единую схему (*схема Active Directory* – набор определений типов, или классов, объектов в БД Active Directory). В лесу между всеми доменами установлены двухсторонние транзитивные доверительные отношения, что позволяет пользователям любого домена получать доступ к ресурсам всех остальных доменов, если они имеют соответствующие разрешения на доступ. По умолчанию, первый домен, создаваемый в лесу, считается его корневым доменом, в корневом домене хранится схема AD.

Новые деревья в лесу создаются в том случае, когда необходимо построить иерархию доменов с пространством имен, отличным от других пространств леса. В примере на рис. 7 российская компания могла открыть офис за рубежом и для своего зарубежного отделения создать дерево с доменом верхнего уровня company.com. При этом оба дерева являются частями одного леса с общим "виртуальным" корнем.

При управлении деревьями и лесами нужно помнить два очень важных момента:

- первое созданное в лесу доменов дерево является корневым деревом, первый созданный в дереве домен называется *корневым доменом дерева* (*tree root domain*);
- первый домен, созданный в лесу доменов, называется *корневым доменом леса* (*forest root domain*), данный домен не может быть удален (он хранит информацию о конфигурации леса и деревьях доменов, его образующих).

Организационные подразделения (*Organizational Units*, *OU*) – контейнеры внутри AD, которые создаются для объединения объектов в целях *делегирования административных прав* и *применения групповых политик* в

домене. ОП существуют *только внутри доменов* и могут объединять *только объекты из своего домена*. ОП могут быть вложенными друг в друга, что позволяет строить внутри домена сложную древовидную иерархию из контейнеров и осуществлять более гибкий административный контроль. Кроме того, ОП могут создаваться для отражения административной иерархии и организационной структуры компании.

Глобальный каталог является перечнем *всех объектов*, которые существуют в лесу Active Directory. По умолчанию, контроллеры домена содержат только информацию об объектах своего домена. Сервер Глобального каталога является контроллером домена, в котором содержится информация о каждом объекте (хотя и не обо всех атрибутах этих объектов), находящемся в данном лесу.

В службе каталогов должен быть механизм именования объектов, позволяющий однозначно идентифицировать любой объект каталога. В каталогах на базе протокола LDAP *для идентификации объекта в масштабе всего леса* используется механизм *отличительных имен* (*Distinguished Name, DN*). В Active Directory учетная запись пользователя с именем *User* домена company.ru, размещенная в стандартном контейнере *Users*, будет иметь следующее отличительное имя: "DC=ru, DC=company, OU=Users, CN=User".

В имени используются следующие обозначения:

- **DC** (Domain Component) – указатель на составную часть доменного имени;
- **OU** (Organizational Unit) – указатель на организационное подразделение (ОП);
- **CN** (Common Name) – указатель на общее имя.

Если отличительное имя однозначно определяет объект в масштабе всего леса, то для идентификации объекта относительно контейнера, в котором данный объект хранится, существует относительное отличительное имя (*Relative Distinguished Name, RDN*). Для пользователя *User* из предыдущего примера RDN-имя будет иметь вид " *CN=User* ".

Кроме имен DN и RDN, используется *основное имя* объекта (*User Principal Name, UPN*). Оно имеет формат <имя субъекта>@<суффикс домена>. Для того же пользователя из примера основное имя будет выглядеть как User@company.ru.

Имена DN, RDN могут меняться, если объект перемещается из одного контейнера AD в другой. Для того чтобы не терять ссылки на объекты при их перемещении в лесу, всем объектам назначается *глобально уникальный идентификатор* (*Globally Unique Identifier, GUID*), представляющий собой 128-битное число.

Планирование пространства имен и структуры AD – очень ответственный момент, от которого зависит эффективность функционирования будущей корпоративной системы безопасности. При этом надо иметь в виду, что созданную вначале структуру в процессе эксплуатации будет очень

трудно изменить (например, в Windows 2000 изменить имя домена верхнего уровня вообще невозможно, а в более поздних версиях операционной системы решение этой задачи требует выполнения жестких условий и тщательной подготовки данной операции). При планировании AD необходимо учитывать следующие моменты:

- тщательный выбор имен доменов верхнего уровня;
- качество коммуникаций в компании (связь между отдельными подразделениями и филиалами);
- организационная структура компании;
- количество пользователей и компьютеров в момент планирования;
- прогноз темпов роста количества пользователей и компьютеров.

3.3. Служба каталогов Active Directory: физическая и логическая структура, репликация данных

Служба каталогов Active Directory организована в виде иерархической структуры, построенной из различных компонентов, которые представляют элементы корпоративной сети. В этой структуре есть, например, пользовательские объекты, компьютерные объекты, и различные контейнеры. Способ организации этих элементов представляет собой логическую структуру Active Directory в корпоративной сети. Логическая структура Active Directory включает в себя уже упомянутые ранее леса, деревья, домены, а также организационные подразделения (ОП). Напомним, что означают эти понятия.

Домен – это логическая группа пользователей и компьютеров, которая поддерживает централизованное администрирование и управление безопасностью. Домен является единицей безопасности – это означает, что администратор для одного домена, по умолчанию, не может управлять другим доменом. Домен также является основной единицей для репликации – все контроллеры одного домена должны участвовать в репликации друг с другом. Домены в одном лесу имеют автоматически настроенные доверительные отношения, что позволяет пользователям из одного домена получать доступ к ресурсам в другом. Необходимо также знать, что можно создавать доверительные отношения с внешними доменами, не входящими в лес.

Дерево является набором доменов, которые связаны отношениями "дочерний"/"родительский", а также используют связанные (смежные, или прилегающие) пространства имен. При этом дочерний домен получает имя от родительского. Например, можно создать дочерний домен, называемый it, в домене company.com, тогда его полное имя будет it.company.com (рис. 8). Между доменами автоматически устанавливаются двухсторонние транзитивные доверительные отношения (домен it.company.com доверяет своему "родительскому" домену, который в свою очередь "доверяет" домену sales.company.com – таким образом, домен it.company.com доверяет домену

sales.company.com, и наоборот). Это означает, что доверительные отношения могут быть использованы всеми другими доменами данного леса для доступа к ресурсам данного домена. Заметим, что домен it.company.com продолжает оставаться самостоятельным доменом, в том смысле, что он остается единицей для управления системой безопасности и процессом репликации. Поэтому, например, администраторы из домена sales.company.com не могут администрировать домен it.company.com до тех пор, пока им явно не будет дано такое право.

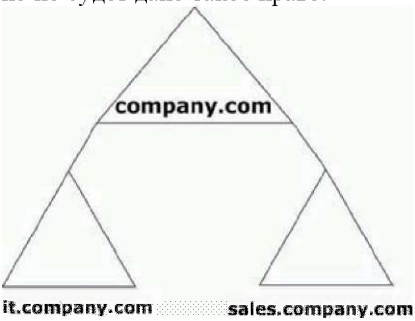


Рисунок 8 – Дерево доменов

Лес – это одно или несколько деревьев, которые разделяют общую *схему*, серверы *Глобального каталога* и *конфигурационную информацию*. В лесу все домены объединены транзитивными двухсторонними доверительными отношениями.

Каждая конкретная инсталляция Active Directory является *лесом*, даже если состоит всего из одного домена.

Организационное подразделение (ОП) является контейнером, который помогает группировать объекты для целей администрирования или применения групповых политик. ОП могут быть созданы для организации объектов в соответствии с их функциями, местоположением, ресурсами и так далее. Примером объектов, которые могут быть объединены в ОП, могут служить учетные записи пользователей, компьютеров, групп и т.д. Напомним, что ОП может содержать только объекты из того домена, в котором они расположены.

Подводя итог, можно сказать, что *логическая структура* Active Directory позволяет организовать ресурсы корпоративной сети таким образом, чтобы они отражали структуру самой компании.

Физическая структура Active Directory служит для связи между логической структурой AD и *топологией* корпоративной сети.

Основные элементы физической структуры Active Directory – *контроллеры домена* и *сайты*.

Контроллеры домена были подробно описаны в предыдущем разделе.

Сайт – группа IP-сетей, соединенных быстрыми и надежными коммуникациями. Назначение сайтов – управление процессом репликации между контроллерами доменов и процессом аутентификации пользователей. Понятие "быстрые коммуникации" очень относительное, оно зависит не только от качества линий связи, но и от объема данных, передаваемых по этим линиям.

Структура сайтов никак не зависит от структуры доменов. Один домен может быть размещен в нескольких сайтах, и в одном сайте могут находиться несколько доменов (рис. 8).

Поскольку сайты соединяются друг с другом медленными линиями связи, механизмы репликации изменений в AD внутри сайта и между сайтами различные. Внутри сайта контроллеры домена соединены линиями с высокой пропускной способностью. Поэтому репликация между контроллерами производится каждые 5 минут, данные при передаче не сжимаются, для взаимодействия между серверами используется технология вызова удаленных процедур (RPC). Для репликации между сайтами кроме RPC может использоваться также протокол SMTP, данные при передаче сжимаются (в результате сетевой трафик составляет от 10 до 40% от первоначального значения), передача изменений происходит по определенному расписанию. Если имеется несколько маршрутов передачи данных, то система выбирает маршрут с наименьшей стоимостью.

Кроме управления репликацией, сайты используются при аутентификации пользователей в домене. Процесс аутентификации может вызвать заметный трафик, особенно если в сети имеется большое количество пользователей (особенно в начале рабочего дня, когда пользователи включают компьютеры и регистрируются в домене). При входе пользователя в сеть его аутентификация осуществляется *ближайшим контроллером домена*. В процессе поиска "ближайшего" контроллера в первую очередь используется информация о сайте, к которому принадлежит компьютер, на котором регистрируется пользователь. Ближайшим считается контроллер, расположенный в том же сайте, что и регистрирующийся пользователь. Поэтому рекомендуется в каждом сайте установить, как минимум, один контроллер домена.

В процессе аутентификации большую роль играет также сервер глобального каталога (при использовании универсальных групп). Поэтому в каждом сайте необходимо также размещать как минимум один сервер глобального каталога (или на одном из контроллеров домена в каждом сайте настроить кэширование членства в универсальных группах). Пользователи сети (в том числе компьютеры и сетевые службы) используют серверы глобального каталога для поиска объектов. В случае, если доступ к серверу глобального каталога осуществляется через линии связи с низкой пропускной способностью, многие операции службы каталога будут выполняться медленно. Это обстоятельство также стимулирует установку сервера гло-

бального каталога в каждом сайте (более подробно о серверах глобального каталога будет рассказано ниже).

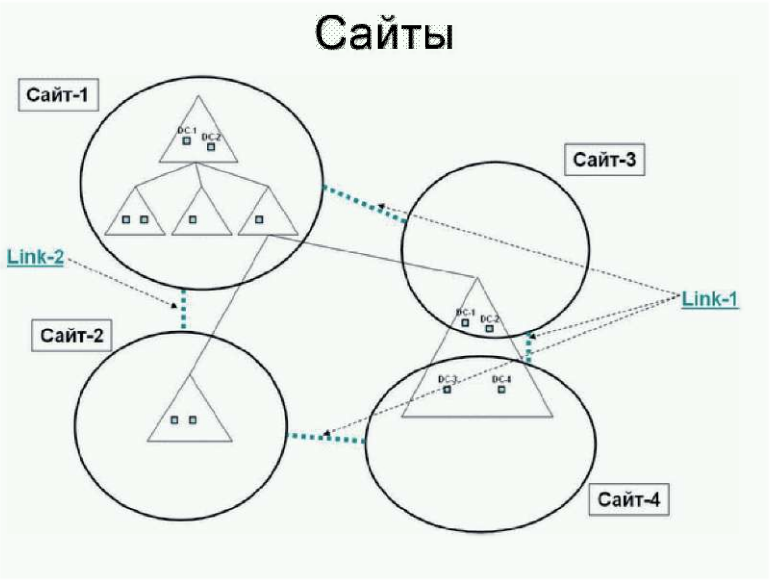


Рисунок 9 – Структура сайтов

В самом начале создания леса автоматически создается сайт по умолчанию с именем *Default-First-site-Name*. В дальнейшем сетевой администратор должен сам планировать и создавать новые сайты и определять входящие в них подсети, а также перемещать в сайты соответствующие контроллеры доменов. При создания *нового* контроллера на основании выделенного ему IP-адреса служба каталога *автоматически* отнесет его к соответствующему сайту.

Репликация изменений в AD между контроллерами домена происходит автоматически, каждые 5 минут. Топологию репликации, т.е. порядок, в котором серверы опрашивают друг друга для получения изменений в базе данных, серверы строят автоматически (эту задачу выполняет компонента служб каталогов, называемая *Knowledge Consistency Checker*, или *KCC*, вариант перевода данного термина – «*наблюдатель показаний целостности*»). При достаточно большом количестве контроллеров KCC строит кольцевую топологию репликации, причем для надежности образует несколько колец, по которым контроллеры передают данные репликации. Наглядно увидеть топологию репликации можно с помощью административной консоли "*Active Directory – сайты и службы*. Если в этой консоли раскрыть последовательно контейнеры *Sites*, *Defauit-First-site-Name*,

Servers, далее – конкретный сервер (например, DC1) и установить на узле *NTDS Settings*, то в правой половине окна видно, что сервер DC1 запрашивает изменения с сервера DC2.

Возможностей управления репликацией у администратора сети в данном случае немного. Можно лишь вызвать принудительную репликацию в той же консоли " *Active Directory – сайты и службы* ".

Разбивать большую корпоративную сеть на отдельные сайты необходимо по следующим причинам: отдельные подсети корпоративной сети, расположенные в удаленных офисах, могут быть подключены друг к другу медленными каналами связи, которые сильно загружены в течение рабочего дня; поэтому возникает необходимость осуществления репликации в те часы, когда сетевой трафик минимален, и передавать данные репликации со сжатием.

С точки зрения механизма репликации Active Directory представляет собой не цельную иерархическую структуру, а отдельные фрагменты. Каждый фрагмент, являясь частью каталога, представляет собой самостоятельное дерево. В терминологии службы Active Directory подобная совокупность ветвей называется прилегающим поддеревом (contiguous subtree) или контекстом имен (naming context).

Разделение пространства имен каталога на фрагменты позволяет оптимизировать процесс синхронизации копий каталога между множеством его носителей. Это достигается за счет того, что в каждом контексте имен хранится определенного вида информация. По умолчанию каталог Active Directory поделен на три контекста имен, которые называются разделы каталога (directory partition):

- доменный раздел каталога (Domain partition) используется для размещения информации о сетевых ресурсах, принадлежащих к определенному домену. Реплики доменного раздела располагаются на всех контроллерах указанного домена. Соответственно изменения, происходящие в этом разделе, реплицируются только на эти реплики;
- раздел схемы (Schema partition). Понятие схемы каталога было дано в начале главы. Для ее хранения используется специальный раздел каталога. Поскольку схема является общей для всех доменов леса, изменения в ней распространяются на все носители копии каталога;
- раздел конфигурации (Configuration partition) содержит информацию, используемую различными системными службами, в том числе и самой службой каталога. В частности, в разделе конфигурации хранится информация, описывающая топологию репликации между контроллерами домена. Эта информация необходима для успешного функционирования службы каталога в целом, поэтому изменения в данном разделе реплицируются на все носители каталога в лесе доменов.

Реплики трех указанных разделов каталога присутствуют в обязательном порядке на всех контроллерах домена. Доменный раздел каталога ин-

дивидуален для каждого домена. Реплики раздела схемы и раздела конфигурации одинаковы для всех контроллеров домена в лесу.

На серверах глобального каталога присутствует еще один раздел – содержащий подмножество атрибутов объектов всех доменных разделов каталога. При этом данный раздел доступен только для чтения информации.

Любой контроллер домена Active Directory может производить изменения в собственных репликах в любой момент времени. При этом все произведенные изменения будут синхронизированы с другими репликами. Подобная модель репликации получила название репликация с множеством равноправных участников (multimaster replication).

3.4. Управление учётными записями и группами в операционной системе Windows Server

Учетные записи (*account*) пользователей, компьютеров и групп – один из главных элементов управления доступом к сетевым ресурсам, а значит, и всей системы безопасности сети в целом.

В Active Directory существует 3 главных типа пользовательских учётных записей:

- Локальные учётные записи пользователей. Эти учетные записи существуют в локальной базе данных *SAM* (*Security Accounts Manager*) на каждой системе, работающей под управлением Windows Server. Эти учётные записи создаются с использованием инструмента *Local Users and Groups* (*Локальные пользователи и группы*) консоли *Computer Management* (*Управление компьютером*). Заметим, что для входа в систему по локальной учётной записи эта учетная запись обязательно должна присутствовать в базе данных SAM на системе, в которую Вы пытаетесь войти. Это делает локальные учёные записи непрактичными для больших сетей, вследствие больших накладных расходов по их администрированию.
- Учётные записи пользователей домена. Эти учётные записи хранятся в Active Directory и могут использоваться для входа в систему и доступа к ресурсам по всему лесу AD. Учетные записи этого типа создаются централизованно при помощи консоли "*Active Directory Users and Computers*" ("*Active Directory – пользователи и компьютеры*").
- Встроенные учётные записи. Эти учётные записи создаются самой системой и не могут быть удалены. По умолчанию любая система, будь то изолированная (отдельно стоящая) или входящая в домен, создает две учётные записи – *Administrator* (*Администратор*) и *Guest* (*Гость*). По умолчанию учётная запись Гость отключена.

Сосредоточим внимание на учётных записях пользователей домена. Эти учётные записи хранятся на контроллерах домена, хранящих копию базы данных Active Directory.

Существуют различные форматы, в которых могут быть представлены имена для входа пользователей в систему, потому что они могут отличать-

ся для целей совместимости с клиентами, работающими под управлением более ранних версий Windows. Два основных вида имён входа – это с использованием суффикса *User Principal Name* (*основного имени пользователя*) и имя входа пользователя в системах пред-Windows 2000.

Основное имя пользователя (*UPN*, *User Principle Name*) имеет такой же формат, как и электронный адрес. Он включает в себя имя входа пользователя, затем значок "@" и имя домена. По умолчанию доменное имя корневого домена выделено в выпадающем окне меню, независимо от того, в каком домене учетная запись была создана (выпадающий список будет также содержать имя домена, в котором вы создали эту учетную запись).

Также можно создавать дополнительные доменные суффиксы (та часть имени, которая стоит после знака @), которые будут появляться в выпадающем списке и могут быть использованы при образовании UPN, если вы их выберете (это делается при помощи консоли "*Active Directory – домены и доверие*" ("*Active Directory Domain and Trusts*").

Существует только одно обязательное условие при этом – все UPN в лесу должны быть уникальными, т.е. не повторяться. Если учётная запись входа пользователя использует UPN для входа в систему Windows Server, вам необходимо только указать UPN и пароль – б лее нет нужды помнить и указывать доменное имя. Другое преимущество данной системы именования состоит в том, что UPN часто соответствует электронному адресу пользователя, что также уменьшает количество информации о пользователе, которую необходимо запоминать.

Каждый компьютер с операционными системами Windows (если это не сервер, являющийся контроллером домена) имеет локальную базу данных учётных записей, называемую базой данных SAM. Эти базы обсуждались при описании модели безопасности "Рабочая группа". Локальные пользователи и особенно группы используются при назначении прав доступа к ресурсам конкретного компьютера даже в доменной модели безопасности. Общие правила использования локальных и доменных групп для управления доступом будут описаны ниже.

Доменные учётные записи пользователей (а также компьютеров и групп) хранятся в специальных контейнерах AD. Это могут быть либо стандартные контейнеры *Users* для пользователей и *Computers* для компьютеров, либо созданное администратором Организационное подразделение (ОП). Исключение составляют учетные записи контроллеров домена, они всегда хранятся в ОП с названием *Domain Controllers*.

При выборе пароля для учётной записи необходимо придерживаться следующих правил:

- длина пароля — не менее 7 символов;
- пароль не должен совпадать с именем пользователя для входа в систему, а также с его обычным именем, фамилией, именами его родственников, друзей и т.д.;

• пароль не должен состоять из какого-либо слова (чтобы исключить возможность подбора пароля по словарю);

• пароль не должен совпадать с номером телефона пользователя (обычного или мобильного), номером его автомобиля, паспорта, водительского удостоверения или другого документа;

• пароль должен быть комбинацией букв в верхнем и нижнем регистрах, цифр и спецсимволов (типа @#$%^*&()_+ и т.д.).

Еще одно правило безопасности – регулярная смена пароля (частота смены зависит от требований безопасности в каждой конкретной компании или организации). В доменах Windows существует политика, определяющая срок действия паролей пользователей.

Свойства учётной записи пользователя содержат большой набор различных параметров, размещенных на нескольких закладках при просмотре в консоли "*Active Directory – пользователи и компьютеры*", причем при установке различных программных продуктов набор свойств может расширяться.

Рассмотрим наиболее важные с точки зрения администрирования свойства. Откроем консоль "*Active Directory – пользователи и компьютеры*" и посмотрим свойства какого-нибудь пользователя. Меню свойств пользователя откроет нам гораздо больше закладок, чем мы обычно видим в клиентских операционных системах.

1. Закладка "*Общие*". На данной закладке содержатся в основном справочные данные, которые могут быть очень полезны при поиске пользователей в лесу AD. Наиболее интересные из них:

• "*Имя*";
• "*Фамилия*";
• "*Выводимое имя*";
• "*Описание*";
• "*Номер телефона*";
• "*Электронная почта*".

2. Закладка "*Адрес*" – справочная информация для поиска в AD.

3. Закладка "*Учётная запись*" – очень важный набор параметров:

• кнопка "*Время входа*" – дни и часы, когда пользователь может войти в домен;

• кнопка "*Вход на...*" – список компьютеров, с которых пользователь может входить в систему (регистрироваться в домене);

• Поле типа чек-бокс "*Заблокировать учетную запись*" – этот параметр недоступен, пока учетная запись не заблокируется после определенного политиками некоторого количества неудачных попыток входа в систему (попытки с неверным паролем), служит для защиты от взлома пароля чужой учетной записи методом перебора вариантов; если будет сделано определенное количество неудачных попыток, то учетная запись пользователя автоматически заблокируется, поле станет доступным и в нем

будет установлена галочка, снять которую администратор может вручную, либо она снимется автоматически после интервала, заданного политиками паролей;

- "*Параметры учетной записи*":
  - "*Требовать смену пароля при следующем входе в систему*"
  - "*Запретить смену пароля пользователем*"
  - "*Срок действия пароля не ограничен*"
  - "*Отключить учетную запись*" – принудительное отключение учётной записи (пользователь не сможет войти в домен);
  - "*Для интерактивного входа в сеть нужна смарт-карта*" – вход в домен будет осуществляться не при помощи пароля, а при помощи смарт-карты (для этого на компьютере пользователя должно быть устройство для считывания смарт-карт, смарт-карты должны содержать сертификаты, созданные Центром выдачи сертификатов);
- "*Срок действия учетной записи*" – устанавливает дату, с которой данная учётная запись не будет действовать при регистрации в домене (этот параметр целесообразно задавать для сотрудников, принятых на временную работу, людей, приехавших в компанию в командировку, студентов, проходящих практику в организации и т.д.)

4. Закладки "*Телефоны*", "*Организация*" – справочная информация о пользователе для поиска в AD.

5. Закладка "*Профиль*". *Профиль* (*profile*) – это настройки рабочей среды пользователя. Профиль содержит: настройки рабочего стола (цвет, разрешение экрана, фоновый рисунок), настройки просмотра папок компьютера, настройки обозревателя Интернета и других программ (например, размещение папок для программ семейства Microsoft Office). Профиль автоматически создается для каждого пользователя при первом входе на компьютер. Различают следующие виды профилей:

- *локальные* – хранятся в папке "*Documents and Settings*" на том разделе диска, где установлена операционная система;
- *перемещаемые* (сетевые, или *roaming*) – хранятся на сервере в папке общего доступа, загружаются в сеанс пользователя на любом компьютере, с которого пользователь вошел (зарегистрировался) в домен, давая возможность пользователю иметь одинаковую рабочую среду на любом компьютере (путь к папке с профилем указывается на данной закладке в виде адреса \\server\share\%username%, где server – имя сервера, share – имя папки общего доступа, %username% - имя папки с профилем; использование переменной среды системы Windows с названием %username% позволяет задавать имя папки с профилем, совпадающее с именем пользователя);
- *обязательные* (*mandatory*) – настройки данного типа профиля пользователь может изменить только в текущем сеансе работы в Windows, при выходе из системы изменения не сохраняются.

Параметр "Сценарий входа" определяет исполняемый файл, который при входе пользователя в систему загружается на компьютер и исполняет-

ся. Исполняемым файлом может быть пакетный файл (.bat, .cmd), исполняемая программа (.exe, .com), файл сценария (.vbs, js).

6. Закладка "*Член групп*" позволяет управлять списком групп, в которые входит данный пользователь.

7. Закладка " *Входящие звонки* ". Управление доступом пользователя в корпоративную систему через средства удаленного доступа системы Windows Server (например, через модем или VPN-соединение). В *смешанном режиме* домена Windows доступны только варианты "*Разрешить доступ*" и "*Запретить доступ*", а также параметры обратного дозвона ("*Ответный вызов сервера*"). В режимах "*Windows 2000 основной*" и " *Windows 20xx*" доступом можно управлять с помощью политик сервера удаленного доступа (не путать с групповыми политиками).

8. Закладки "*Профиль служб терминалов*", "*Среда*", "*Сеансы*", "*Удаленное управление*" – данные закладки управляют параметрами работы пользователя на сервере терминалов:

- управление разрешением пользователя работать на сервере терминалов;
- размещение профиля при работе в терминальной сессии,
- настройка среды пользователя в терминальной сессии (запуск определенной программы или режим рабочего стола, подключение локальных дисков и принтеров пользователя в терминальную сессию);
- управление сеансом пользователя на сервере терминалов (длительность сессии, тайм-аут бездействия сессии, параметры повторного подключения к отключенной сессии);
- разрешение администратору подключаться к терминальной сессии пользователя

Учётные записи групп, как и учетные записи пользователей, могут быть созданы либо в локальной базе SAM компьютера (сервера или рабочей станции), либо в доменной базе данных Active Directory.

Локальные группы простого сервера-члена домена или рабочей станции могут включать в себя и локальные учётные записи данного компьютера, и глобальные учётные записи любого пользователя или компьютера всего леса, а также доменные локальные группы "своего" домена и глобальные и универсальные группы всего леса.

Рассмотрим подробнее, какие группы могут создаваться в Active Directory. В Active Directory группы различаются по типу (группы безопасности и группы распространения) и по области действия (локальные в домене, глобальные и универсальные).

Каждая группа безопасности, так же, как и каждая учётная запись пользователя, имеет *идентификатор безопасности* (*Security Identifier*, или *SID*), поэтому группы безопасности используются для назначения разрешений при определении прав доступа к различным сетевым ресурсам.

Группы распространения не имеют идентификатора безопасности, поэтому не могут использоваться для назначения прав доступа, их главное

назначение – организация списков рассылки для почтовых программ (например, для Microsoft Exchange Server).

Локальные доменные группы могут содержать глобальные группы из любого домена, универсальные группы, глобальные учётные записи пользователей из любого домена леса, и используются при назначении прав доступа только к ресурсам "своего" домена.

Глобальные группы могут содержать только глобальные учётные записи пользователей "своего" домена, и используются при назначении прав доступа к ресурсам любого домена в лесу.

Универсальные группы могут содержать другие универсальные группы всего леса, глобальные группы всего леса, глобальные учётные записи пользователей из любого домена леса, и используются при назначении прав доступа к ресурсам любого домена в лесу.

В смешанном режиме домена универсальные группы недоступны для использования. В основном режиме или режиме Windows Server можно создавать и использовать универсальные группы. Кроме того, в основном режиме и режиме Windows Server глобальные группы могут включаться в другие глобальные группы, а доменные локальные группы могут включаться в другие доменные локальные.

Специфика универсальных групп заключается в том, что эти группы хранятся в Глобальном каталоге. Поэтому, если пользователь является членом универсальной группы, то при регистрации в домене ему обязательно должен быть доступен контроллер домена, являющийся сервером глобального каталога, в противном случае пользователь не сможет войти в сеть. Репликация между простыми контроллерами домена и серверами глобального каталога происходит достаточно медленно, поэтому любое изменение в составе универсальной группы требует больше времени для репликации, чем при изменении состава групп с другими областями действия.

При регистрации в домене пользователю передается в его сессию на компьютере т.н. *маркер доступа* (*Access Token*), называемый иногда маркером безопасности. Маркер доступа состоит из набора идентификаторов безопасности: идентификатора безопасности (SID) самого пользователя и идентификаторов безопасности тех групп, членом которых он является. Впоследствии этот маркер доступа используется при проверке разрешений пользователя на доступ к различным ресурсам домена.

При создании и использовании групп следует придерживаться следующих правил:

1. Включать глобальные учетные записи пользователей (Accounts) в глобальные группы (Global groups). Глобальные группы формируются обычно *по функциональным обязанностям сотрудников*.

2. Включать глобальные группы в доменные локальные или локальные на простом сервере или рабочей станции (Local groups). Локальные группы формируются *на основе разрешений для доступа к конкретным*

ресурсам и используются преимущественно для включения в списки доступа.

3. Давать разрешения (Permissions) на доступ к ресурсам локальным группам.

Предложенные рекомендации позволяют избегать путаницы в назначении прав доступа пользователям, избавляют администратора от необходимости назначать права доступа каждой учётной записи в отдельности, а также помогают упростить контроль назначенных и снятых прав доступа в случае, если какой-либо учётной записью временно перестают пользоваться, например, в случае увольнения сотрудника, за которым она была закреплена.

По первым буквам английских слов эту стратегию часто обозначают сокращенно **AGLP**. В основном режиме и режиме Windows Server с использованием универсальных групп эта стратегия может быть в более общем виде представлена как аббревиатура **AGG…GULL…LP**. Такой подход облегчает управление доступом к ресурсам по сравнению с назначением разрешений напрямую учетным записям пользователей. Например, при переходе сотрудника с одной должности на другую или из одного подразделения в другое достаточно соответствующим образом поменять его членство в различных группах, и разрешения на доступ к сетевым ресурсам автоматически будут назначены уже исходя из его новой должности.

Кроме тех групп, которые создает администратор, на компьютерах локально или во всем домене существуют *встроенные группы*, созданные во время установки системы или создания домена. Кроме встроенных групп в процессе работы системы формируются *динамические группы*, состав которых меняется в зависимости от ситуации.

Перечислим наиболее часто используемые на практике встроенные и динамические группы.

Таблица 2 – Встроенные и динамические группы

Встроенные локальные группы (на рабочей станции или простом сервере)	
Название группы	Описание
Администраторы	Могут выполнять все административные задачи на данном компьютере. Встроенная учетная запись *Администратор*, которая создается при установке системы, является членом этой группы. Если компьютер является членом домена, то в эту группу включается глобальная группа *Администраторы домена*.
Операторы резервного копирования	Члены группы могут выполнять вход на данный компьютер, выполнять резервное копирование и восстановление данных на этом компьютере, а

Продолжение таблицы 2

	также завершать работу этого компьютера.
Администраторы DHCP (создается при установке службы DHCP Server)	Члены этой группы могут администрировать службу DHCP Server.
Операторы сетевой конфигурации	Члены группы могут изменять настройки TCP/IP, а также обновлять и освобождать IP-адреса, назначаемые автоматически.
Пользователи монитора производительности	Члены группы могут следить за счетчиками производительности на конкретном сервере локально или удаленным образом.
Пользователи журнала производительности	Члены группы могут администрировать журналы производительности, счетчики и оповещения на конкретном сервере локально или удаленным образом.
Опытные пользователи	Члены группы могут создавать и модифицировать учетные записи пользователей, а также устанавливать программы на локальном компьютере, но не могут просматривать файлы других пользователей. Члены группы могут создавать и удалять локальные группы, а также добавлять и удалять пользователей в группах, которые они создали. Члены группы могут добавлять и удалять пользователей в группах *Опытные пользователи*, *Пользователи* и *Гости*.
Операторы печати	Члены группы могут управлять принтерами и очередями печати на конкретном сервере.
Пользователи удаленного рабочего стола	Членам группы разрешается выполнять подключение к удаленному рабочему столу компьютера.
Пользователи	Члены этой группы могут локально входить в систему на данном компьютере, работать с программами, сохранять документы и завершать работу данного компьютера. Они не могут устанавливать программы или вносить изменения в систему. Если компьютер является членом домена, то в эту группу включается глобальная группа *Пользователи домена*. В эту группу также включаются динамические группы *Интерактивные* и *Прошедшие проверку*.
Встроенные доменные локальные группы	

Продолжение таблицы 2

Название группы	Описание
Администраторы	Членам группы предоставляются права администратора на всех контроллерах домена и в самом домене. Учетная запись *Администратор*, группы *Администраторы предприятия* и *Администраторы домена* являются членами данной группы.
Операторы учетных записей	Члены группы могут создавать, удалять и управлять учетными записями пользователей и группами. Они не могут модифицировать группу *Администраторы*, *Администраторы домена*, *Контроллеры домена* или любую из групп *Операторы*.
Операторы резервного копирования	Члены группы могут выполнять резервное копирование и восстановление данных на всех контроллерах домена, а также могут выполнять вход на контроллеры домена и завершать их работу.
Администраторы DNS (создается при установке службы DNS)	Члены группы имеют административный доступ к серверам DNS.
Операторы сетевой конфигурации	Члены группы могут изменять настройки TCP/IP на контроллерах доменов.
Пользователи монитора производительности	Члены группы могут следить за счетчиками производительности на контроллерах домена.
Пользователи журнала производительности	Члены группы могут управлять журналами производительности, счетчиками и оповещениями на контроллерах домена.
Операторы печати	Члены группы могут управлять работой принтеров домена
Операторы сервера	Члены группы могут выполнять большинство административных задач на контроллерах домена, за исключением изменения параметров безопасности.
Пользователи	Члены этой группы локально могут входить в систему на данном компьютере, работать с программами, сохранять документы и завершать работу данного компьютера. Они не могут устанавливать программы или вносить изменения

	в систему. Группа *Пользователи домена* является по умолчанию членом данной группы.
Встроенные глобальные группы	
Название группы	Описание
Администраторы домена	Эта группа автоматически включается в локальную в домене группу *Администраторы*, поэтому члены группы *Администраторы домена* могут выполнять административные задачи на любом компьютере данного домена. Учетная запись *Администратор* включается в эту группу по умолчанию.
Компьютеры домена	Все контроллеры, серверы и рабочие станции домена являются членами этой группы.
Контроллеры домена	Все контроллеры домена являются членами этой группы.
Пользователи домена	Все глобальные учетные записи домена и входят в эту группу. Эта группа автоматически включается в локальную доменную группу *Пользователи*.
Администраторы предприятия (создается только в корневом домене леса)	Эта группа предназначена для пользователей, которые должны иметь права администратора в масштабах всего леса. *Администраторы предприятия* автоматически включается в группу *Администраторы* на всех контроллерах домена в данном лесу.
Администраторы схемы (создается только в корневом домене леса)	Члены этой группы могут изменять схему Active Directory.
Динамические группы	
Название группы	Описание
Интерактивные	В эту группу включается учетная запись любого пользователя, который локально вошел в систему на данном компьютере.
Прошедшие проверку	Любой пользователь, зарегистрировавшийся в данном домене или домене, имеющим с данным доменом доверительные отношения.
Все	Любая учетная запись, включая те, которые не прошли проверку на контроллерах доменов.

Назначением *организационных подразделений* (*ОП*, *Organizational Units*, *OU*) является организация иерархической структуры объектов AD внутри домена. Как правило, иерархия ОП в домене отражает организационную структуру компании.

На практике использование ОП (кроме иерархической организации объектов) сводится к двум задачам:

- делегирование административных полномочий на управление объектами ОП какому-либо пользователю или группе пользователей;
- применение групповых политик к объектам, входящим в ОП.

Делегирование административных полномочий на управление объектами ОП какому-либо пользователю или группе позволяет в больших организациях распределить нагрузку по администрированию учетными записями между различными сотрудниками, не увеличивая при этом количество пользователей, имеющих административные права на уровне всего домена.

3.5. Методы обеспечения безопасности в Active Directory, аутентификация Kerberos

Протокол *Kerberos* был создан в Массачусетском технологическом институте в рамках проекта *Athena*. Однако общедоступным этот протокол стал, начиная с версии 4. После того, как специалисты изучили новый протокол, авторы разработали и предложили очередную версию – Kerberos 5, которая была принята в качестве стандарта IETF. Требования реализации протокола изложены в документе RFC 1510, кроме того, в спецификации RFC 1964 описывается механизм и формат передачи жетонов безопасности в сообщениях Kerberos.

Протокол Kerberos предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними, причём в протоколе учтён тот факт, что начальный обмен информацией между клиентом и сервером происходит в незащищённой среде, а передаваемые пакеты могут быть перехвачены и модифицированы. Другими словами, протокол идеально подходит для применения в Интернет и аналогичных сетях.

Основная концепция протокола Kerberos очень проста. Если есть секрет, известный только двоим, то любой из его хранителей может с лёгкостью удостовериться, что имеет дело со своим напарником. Для этого ему достаточно проверить, знает ли его собеседник общий секрет.

Протокол Kerberos решает эту проблему средствами криптографии с секретным ключом. Вместо того, чтобы сообщать друг другу пароль, участники сеанса связи обмениваются криптографическим ключом, знание которого подтверждает личность собеседника. Но чтобы такая технология оказалась работоспособной, необходимо, чтобы общий ключ был симметричным, т.е., он должен обеспечивать как шифрование, так и дешифрова-

ние информации. Тогда один из участников использует его для шифрования данных, а другой с помощью этого ключа извлекает их.

Простой протокол аутентификации с секретным ключом вступает в действие, когда кто-то стучится в сетевую дверь и просит впустить его. Чтобы доказать своё право на вход, пользователь предъявляет *аутентификатор* (*authenticator*) в виде набора данных, зашифрованного секретным ключом. Получив аутентификатор, "привратник" расшифровывает его и проверяет полученную информацию, чтобы убедиться в успешности дешифрования. Разумеется, содержание набора данных должно постоянно меняться, иначе злоумышленник может просто перехватить пакет и воспользоваться его содержимым для входа в систему. Если проверка прошла успешно, то это значит, что посетителю известен секретный код, а так как этот код знает только он и привратник, следовательно, пришелец на самом деле тот, за кого себя выдаёт.

При использовании простых протоколов, типа, описанного выше, возникает одна важная проблема. Если каждому клиенту для поддержания связи с каждой службой требуется индивидуальный ключ, и такой же ключ нужен каждой службе для каждого клиента, то проблема обмена ключами быстро приобретает предельную остроту. Необходимость хранения и защиты такого множества ключей на огромном количестве компьютеров создаёт невероятный риск для всей системы безопасности.

Само название протокола Kerberos говорит о том, как здесь решена проблема управления ключами. Цербер или Кербер – персонаж греческой мифологии. Этот свирепый пёс о трёх головах, по поверьям греков, охраняет врата подземного царства мёртвых. Трём головам Цербера в протоколе Kerberos соответствуют три участника безопасной связи:

- клиент – система (пользователь), делающий запрос;
- сервер – система, которая обеспечивает сервис для систем, чью подлинность нужно подтвердить.
- центр распределения ключей (*Key Distribution Center*, *KDC*) – сторонний посредник между клиентом и сервером, который ручается за подлинность клиента. В среде Windows, начиная с Windows 2000, в роли KDC выступает контроллер домена со службой каталогов Active Directory.

В среде Kerberos для входа в систему пользователь должен предоставить свое имя пользователя, пароль и имя домена, часто упоминаемое как *Realm*, или "*Сфера*", в словаре Kerberos, в который он хочет войти. Эта информация посылается KDC, который устанавливает подлинность пользователя. Если пользователь подлинный, ему предоставляется нечто, называемое "*билет на получение билета*" (*ticket-granting ticket*, *TGT*).

Однако, если вам необходим доступ к конкретному серверу, вам также необходим билет для этого сервера или вы не сможете создать сеанс связи с ним.

Когда Вы хотите получить доступ к серверу, Вы сначала должны обратиться к KDC, предъявить свой билет TGT, как подтверждение своей

подлинности, а затем уже запросить "*билет сеанса*" для сервера, с которым вам необходим контакт. Если вы аутентифицированы, Вы сможете получить доступ к серверу в соответствии с правами, которыми обладаете. Билет сеанса и TGT, которые Вы получаете, имеют ограниченное время действия, которое может настраиваться в групповой политике. Значения по умолчанию составляют для TGT (также упоминаемого как билет пользователя) – 7 дней, а для билета сеанса (также упоминаемого как билет службы) – 10 часов.

В среде с одним доменом аутентификация Kerberos осуществляется очень просто. Однако в среде со многими доменами, этот процесс происходит в несколько этапов. Причина в том, что, когда Вы пытаетесь получить билет сессии для сервера, он должен быть получен от KDC того домена, в котором расположен сервер. Поэтому Вы должны будете получить несколько билетов сессии, для прохождения цепочки доверительных отношений по пути к KDC, к которому вам нужно получить доступ.

Пример, приведенный ниже, демонстрирует шаги, необходимые для того, чтобы клиент, расположенный в домене it.company.ru, получил доступ к серверу в домене sales.company.ru:

- клиент входит в систему как пользователь в домене it.company.ru и получает соответствующий TGT;
- клиент хочет взаимодействовать с сервером в домене sales.company.ru, он контактирует с KDC в домене it.company.ru и запрашивает билет сеанса для KDC в домене company.ru;
- после получения этого билета он контактирует с KDC в домене company.ru и запрашивает билет сеанса для KDC в домене sales.company.ru;
- после получения этого билета он контактирует с KDC в домене sales.company.ru и запрашивает билет для сервера, к которому ему необходим доступ;
- получив билет сессии для доступа к серверу, клиент имеет доступ к нему в соответствии с имеющимися у него разрешениями.

3.6. Групповые политики и управление ими

Управление рабочими станциями, серверами, пользователями в большой организации – очень трудоемкая задача. Механизм *Групповых политик* (*Group Policy*) позволяет автоматизировать данный процесс управления. С помощью групповых политик (*ГП*) можно настраивать различные параметры компьютеров и пользовательской рабочей среды сразу в масштабах сайта AD, домена, организационного подразделения (детализацию настроек можно проводить вплоть до отдельного компьютера или пользователя). Настраивать можно широкий набор параметров – сценарии входа в систему и завершения сеанса работы в системе, параметры *Рабочего стола* и *Панели управления*, размещения личных папок пользователя, на-

стройки безопасности системы (политики паролей, управления учетными записями, аудита доступа к сетевым ресурсам, управления сертификатами и т.д.), развертывания приложений и управления их жизненным циклом. Совокупность этих параметров и называется *групповой политикой*.

Каждый *объект групповых политик* (*GPO*, *Group Policy Object*) состоит из двух частей: *контейнера групповых политик* (GPC, *Group Policy Container*) хранящегося в БД Active Directory, и *шаблона групповых политик* (*GPT*, *Group Policy Template*), хранящегося в файловой системе контроллера домена, в подпапках папки SYSVOL. Место, в котором хранятся шаблоны политик, - это папка %systemroot%\SYSVOL\sysvol\<имя домена>\Policies, и имя папки шаблона совпадает с глобальным уникальным идентификатором (GUID) объекта Групповая политика.

Каждый объект политик содержит два раздела: конфигурация компьютера и конфигурация пользователя. Параметры этих разделов применяются соответственно либо к настройкам компьютера, либо к настройкам среды пользователя.

По умолчанию в GPT содержатся подкаталоги *User* (конфигурация пользователя) и *Machine* (конфигурация компьютера) и файл *gpt.ini*. По мере настройки объекта групповой политики в папке GPT появляются дополнительные файлы и папки.

В общем случае структура GPT содержит следующие папки, перечисленные в таблице 3.

Таблица 3 – папки шаблона групповой политики

Папка	Содержимое
\Adm	Файлы административных шаблонов .adm, формируемые на основе информации в GPC и GPT. Эти файлы обрабатываются Windows Server для внесения изменений в реестр
\User	Файл registry.pol с параметрами, заносимыми в реестр для пользователя
\User\Applications	Файл оповещения .aas, используемые компонентом Windows Installer для публикации пакетов установки ПО для пользователя
\User\Documents & Settings	Любые файлы, добавляемые на рабочий стол, в меню Пуск, папку Мои документы и другие папки профиля пользователя

Продолжение таблицы 3

\User\Scripts\Logon	Сценарии входа и дополнительные файлы, используемые этими сценариями
\User\Scripts\Logoff	Сценарии выхода и дополнительные файлы, используемые этими сценариями
\Machine	Файл registry.pol с параметрами, заносимыми в реестр компьютера
\Machine\Applications	Файл оповещения .aas, используемые компонентом Windows Installer для публикации пакетов установки ПО для всех пользователей компьютера
\Machine\Documents & Settings	Любые файлы, добавляемые на рабочий стол, в меню Пуск, папку Мои документы и другие папки профиля всех пользователя компьютера
\Machine\Microsoft\WindowsNT\SecEdit	Файл GptTmpl.ini, используемый редактором Security Editor
\Machine\Scripts\Startup	Сценарии запуска и дополнительные файлы, используемые этими сценариями
\Machine\Scripts\Shutdown	Сценарии выключения и дополнительные файлы, используемые этими сценариями

В корневой папке каждого GPT должен содержаться файл Gpt.ini, используемый для задания основных параметров объекта групповой политики. Файл является обычным текстовым файлом и может содержать следующие записи:

1. Version. Номер текущей версии объекта групповой политики. При создании объекта групповой политики номер версии равен 0 и увеличивается на единицу при каждом изменении объекта групповой политики.

2. Disabled. Может принимать значения 0 или 1 и используется только для локальных объектов групповой политики, указывая активен локальный объект групповой политики или нет. В доменных объектах групповой политики этот параметр не используется, т.к. информация об активности объекта групповой политики хранится в каталоге Active Directory в GPC.

Задание параметров групповых политик производится Редактором групповых политик, который можно открыть в консоли управления соответствующим объектом AD.

Каждый объект политик может быть привязан к тому или иному объекту AD – сайту, домену или организационному подразделению (а также к нескольким объектам одновременно).

Задание параметров групповых политик производится Редактором групповых политик, который можно открыть в консоли управления соответствующим объектом AD ("*Active Directory – сайты и службы*", "*Active Directory – пользователи и компьютеры*", локальная политика компьютера редактируется консолью gpedit.msc, запускаемой из командной строки).

При загрузке компьютера и аутентификации в домене к нему применяются компьютерные разделы всех привязанных политик. При входе пользователя в систему к пользователю применяется пользовательский раздел всех групповых политик. Политики, привязанные к некоторому уровню иерархии объектов AD (сайта, домена, подразделения) наследуются всеми объектами AD, находящимися на более низких уровнях. Порядок применения политик:

- локальная политика;
- политики сайта Active Directory;
- политики домена;
- политики организационных подразделений.

Если в процессе применения политик какие-либо параметры определяются в различных политиках, то действующими значениями параметров будут значения, определенные позднее.

Имеются следующие методы управления применением групповых политик:

- блокировка наследования политик на каком-либо уровне иерархии AD;
- запрет блокировки конкретного объекта групповых политик;
- управление приоритетом применения политик на конкретном уровне AD;
- разрешение на применение политик (чтобы политики какого-либо объекта ГП применялись к пользователю или компьютеру, данный пользователь или компьютер должен иметь разрешения на этот объект ГП "*Чтение*" и "*Применение групповой политики*").

Кроме применения политик в момент загрузки компьютера или входа пользователя в систему, каждый компьютер постоянно запрашивает обновленные политики на контроллерах домена, загружает их и применяет обновленные параметры (и к пользователю, и к компьютеру). Рабочие станции домена и простые серверы запрашивают обновления каждые 90 ± 30 минут, контроллеры домена обновляют свои политики каждые 5 минут. Обновить набор политик на компьютере можно принудительно из командной строки командой gpupdate или командами " secedit /refreshpolicy

machine_policy " и " secedit /refreshpolicy user_policy " (на компьютерах с системой Windows 2000).

Рассмотрим подробнее использование групповых политик для развертывания приложений в сетях под управлением Active Directory.

Групповые политики могут использоваться для установки прикладных программ в масштабах всего домена или отдельного организационного подразделения.

Используются следующие способы управления установкой приложений:

- *назначение приложений компьютерам* – при данном способе приложение, назначенное компьютеру, автоматически устанавливается при загрузке компьютера;
- *назначение приложений пользователям* – приложение устанавливается при первом вызове данного приложения – при открытии ярлычка приложения или файла, соответствующего данному приложению;
- *публикация приложений пользователям* – название приложения добавляется к списку доступных для установки программ в окне "*Установка и удаление программ*" в *Панели управления*.

С помощью политик можно управлять установкой приложений, которые устанавливаются с помощью компоненты *Windows Installer*, т.е. для них установочный пакет должен быть создан в формате файла с расширением ".msi ". Если приложение можно установить только с помощью установочной программы типа setup.exe или install.exe, то такие приложения *могут быть опубликованы*, но не назначены, после создания файла типа ".zap", в котором заданны соответствующие параметры, необходимые для публикации средствами ГП.

Контрольные вопросы

1. Что такое служба каталогов?
2. Какие возможности обеспечивает служба каталогов?
3. Что такое пространство имён X.500?
4. Что такое LDAP?
5. Что такое репликация?
6. Что такое домен?
7. Что такое контроллер домена?
8. Что такое дерево доменов?
9. Что такое лес доменов?
10. Когда необходимо построение дерева?
11. Когда в лесу доменов строится более одного дерева?
12. Что такое подразделение в AD?
13. Какие составные части имеет доменное имя?
14. Что такое глобальный каталог?
15. Что необходимо учитывать при планировании пространства имён?

16. Что входит в физическую структуру службы каталогов?
17. Что входит в логическую структуру службы каталогов?
18. Какие контексты имён используются в каталоге?
19. Для чего используется доменный раздел?
20. Что содержится в разделе схемы?
21. Что содержится в разделе конфигурации?
22. Что такое учётная запись пользователей?
23. Какие виды учётных записей могут храниться в базе?
24. Каких правил нужно придерживаться при выборе пароля учётной записи?
25. Что такое идентификатор безопасности?
26. Что такое группа безопасности?
27. Какова область действия локальной доменной группы?
28. Какие объекты могут входить в локальную доменную группу?
29. Какова область действия глобальной группы?
30. Какие объекты могут входить в глобальную группу?
31. Какова область действия универсальной группы?
32. Какие объекты могут входить в универсальную группу?
33. Какая группа обычно используется для назначения прав доступа?
34. Что такое Kerberos?
35. Что такое аутентификатор?
36. Что такое центр распределения ключей?
37. Опишите схему работы протокола Kerberos.
38. Что такое групповая политика?
39. Что такое объект групповой политики?
40. Что такое шаблон групповой политики?
41. Каков порядок применения групповых политик?
42. Назовите методы управления применением групповых политик.
43. Перечислите способы управления установкой приложений.

Задание

Установите на виртуальную машину на бесплатной прикладной платформе виртуализации, например, VirtualBox, операционную систему Windows Server с графическим интерфейсом и добавьте в диспетчере серверов следующие роли и компоненты:

1. Доменные службы Active Directory.
2. DNS-сервер.

Обратите внимание, что операционная система Windows Server, начиная с Windows Server 2012, имеет только 64-битную архитектуру, поэтому физический процессор Вашего компьютера обязательно должен быть 64-разрядным. В противном случае можно установить систему не старше Windows Server 2008.

Повысьте роль сервера с добавленными ролями до уровня контроллера домена при помощи диспетчера серверов.

С помощью оснастки Active Directory «Пользователи и компьютеры» создайте новую учётную запись пользователя и папку на рабочем столе. Создайте Подразделение для групп пользователей и Подразделение для настройки прав доступа. В первом создайте глобальную группу и добавьте в неё созданного пользователя. Во втором создайте локальную группу и добавьте в неё ранее созданную группу. После этого назначьте локальной группе права доступа к папке на рабочем столе.

4. СИСТЕМА ДОМЕННЫХ ИМЁН DNS И СЛУЖБА DHCP

4.1. Основные понятия, назначение и характеристики DNS

DNS (*Domain Name System* – система доменных имён) – это компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

Смысл системы DNS состоит в следующем. Как известно, компьютеры и мобильные устройства, интегрированные во Всемирную паутину, связываются друг с другом по сетевым адресам, то есть IP-адресам. А IP-адрес, как мы уже знаем – это некая числовая структура, которая простому пользователю совершенно ни о чём не говорит, поэтому пользователя требуется избавлять от необходимости оперировать какими-то наборами чисел при работе в интернете, которые он никогда не запомнит. Так в сети Интернет и происходит – в адресной строке браузера обычно содержатся символьные имена ресурсов, а никак не IP-адреса серверов, на которых они находятся.

Но теперь уже компьютеру символьное имя ресурса ни о чём не говорит, ведь его «не интересует», как «зовут» поисковик Яндекса или Google, его «интересует», где «живёт» поисковик Яндекса, то есть ему нужен IP-адрес – та самая числовая структура, в которой пользователь ничего не понимает. И чтобы разрешать эту ситуацию, в глобальной сети функционирует система DNS, задачей которой и является отвечать клиенту, какой IP-адрес имеет компьютер с указанными доменным именем.

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения – другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Начиная с 2010 года в систему DNS внедряются средства проверки целостности передаваемых данных, называемые DNS Security Extensions (DNSSEC). Передаваемые данные не шифруются, но их достоверность проверяется криптографическими способами. Внедряемый стандарт DANE обеспечивает передачу средствами DNS достоверной криптографической информации (сертификатов), используемых для установления безопасных и защищённых соединений транспортного и прикладного уровней.

Дерево DNS принято делить по уровням: первый, второй, третий и так далее. При этом начинается система с единственного корневого домена

(нулевой уровень). Интересно, что о существовании корневого домена сейчас помнят только специалисты, благодаря тому, что современная DNS позволяет не указывать этот домен в адресной строке. Впрочем, его можно и указать. Адресная строка с указанием корневого домена выглядит, например, так: «site.test.ru.» – здесь корневой домен отделен последней, крайней справа, точкой. Как несложно догадаться, адреса с использованием DNS записываются в виде последовательности, отражающей иерархию имен. Чем «выше» уровень домена, тем правее он записывается в строке адреса. Разделяются домены точками. Разберем, например, строку www.site.nic.ru. Здесь домен www – это домен четвертого уровня, а другие упомянутые в этой строке домены расположены в домене первого уровня RU. Например, site.nic.ru – это домен третьего уровня. Очень важно понимать, что привычный адрес веб-сайта, скажем, www.test.ru, обозначает домен третьего уровня (www), расположенный внутри домена второго уровня test.ru.

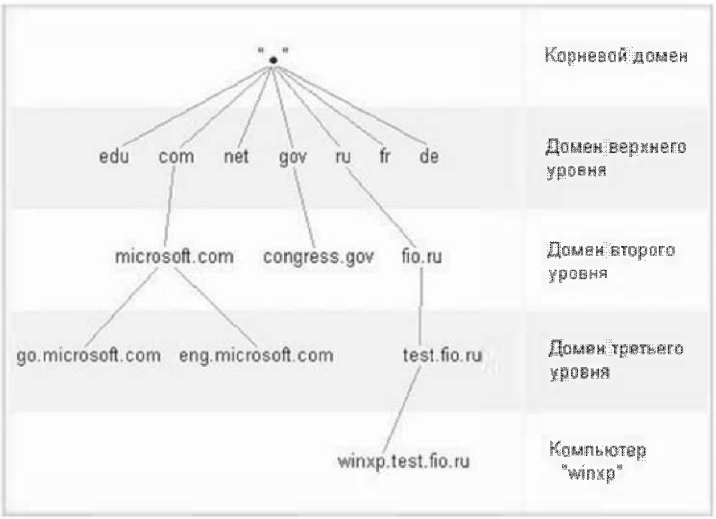


Рис. 10 – Пример структуры доменного имени

DNS обладает следующими характеристиками:

- *Распределённость администрирования.* Ответственность за разные части иерархической структуры несут разные люди или организации.
- *Распределённость хранения информации.* Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности, и (возможно) адреса корневых DNS-серверов.
- *Кеширование информации.* Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.

• *Иерархическая структура*, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

• *Резервирование*. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Ключевыми понятиями DNS являются:

1. Домен – узел в дереве имён, вместе со всеми подчинёнными ему узлами (если таковые имеются), то есть именованная ветвь или поддерево в дереве имён. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего уровня (в порядке повышения значимости)

2. Поддомен – подчинённый домен.

3. Ресурсная запись – единица хранения и передачи информации в DNS. Каждая ресурсная запись имеет имя, то есть привязана к определённому доменному имени, узлу в дереве имён, тип и поле данных, формат и содержание которого зависит от типа.

4. Зона – часть дерева доменных имён, включая ресурсные записи, размещаемая как единое целое на некотором сервере доменных имён, а чаще одновременно на нескольких серверах. Целью выделения части дерева в отдельную зону является передача ответственности за соответствующий домен другому лицу или организации. Это называется делегированием. Как связная часть дерева, зона внутри тоже представляет собой дерево. Если рассматривать пространство имён DNS как структуру из зон, а не отдельных узлов/имён, тоже получается дерево; оправданно говорить о родительских и дочерних зонах, о старших и подчинённых. На практике большинство зон 0-го и 1-го уровня ('.', ru, com, …) состоят из единственного узла, которому непосредственно подчиняются дочерние зоны. В больших корпоративных доменах (2-го и более уровней) иногда встречается образование дополнительных подчинённых уровней без выделения их в дочерние зоны.

5. Делегирование – операция передачи ответственности за часть дерева доменных имён другому лицу или организации. За счёт делегирования в DNS обеспечивается распределённость администрирования и хранения. Технически делегирование выражается в выделении этой части дерева в отдельную зону, и размещении этой зоны на DNS-сервере, управляемом этим лицом или организацией. При этом в родительскую зону включаются «склеивающие» ресурсные записи (NS и A), содержащие указатели на DNS-сервера дочерней зоны, а вся остальная информация, относящаяся к дочерней зоне, хранится уже на DNS-серверах дочерней зоны.

6. DNS-сервер – специализированное ПО для обслуживания DNS, а также компьютер, на котором это ПО выполняется. DNS-сервер может

быть ответственным за некоторые зоны и/или может перенаправлять запросы вышестоящим серверам.

7. DNS-клиент – специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

8. Авторитетность – признак размещения зоны на DNS-сервере. Ответы DNS-сервера могут быть двух типов: авторитетные (когда сервер заявляет, что сам отвечает за зону) и неавторитетные, когда сервер обрабатывает запрос, и возвращает ответ других серверов. В некоторых случаях вместо передачи запроса дальше DNS-сервер может вернуть уже известное ему (по запросам ранее) значение (режим кеширования).

9. DNS-запрос – запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или нерекурсивным.

Система DNS содержит иерархию DNS-серверов, соответствующую иерархии зон. Каждая зона поддерживается как минимум одним авторитетным сервером DNS, на котором расположена информация о домене.

Имя и IP-адрес не тождественны – один IP-адрес может иметь множество имён, что позволяет поддерживать на одном компьютере множество веб-сайтов (это называется виртуальный хостинг). Обратное тоже справедливо - одному имени может быть сопоставлено множество IP-адресов: это позволяет создавать балансировку нагрузки.

Для повышения устойчивости системы используется множество серверов, содержащих идентичную информацию, а в протоколе есть средства, позволяющие поддерживать синхронность информации, расположенной на разных серверах. Существует 13 корневых серверов, их адреса практически не изменяются.

Протокол DNS использует для работы TCP- или UDP-порт 53 для ответов на запросы. Традиционно запросы и ответы отправляются в виде одной UDP-датаграммы. TCP используется, когда размер данных ответа превышает 512 байт, и для AXFR-запросов

Итак, DNS важна для работы Интернета, так как для соединения с узлом необходима информация о его IP-адресе, а для людей проще запоминать буквенные (обычно осмысленные) адреса, чем последовательность цифр IP-адреса. В некоторых случаях это позволяет использовать виртуальные серверы, например, HTTP-серверы, различая их по имени запроса. Первоначально преобразование между доменными и IP-адресами производилось с использованием специального текстового файла hosts, который составлялся централизованно и автоматически рассылался на каждую из машин в своей локальной сети. С ростом Сети возникла необходимость в эффективном, автоматизированном механизме, которым и стала DNS. DNS была разработана Полом Мокапетрисом в 1983 году; оригинальное описание механизмов работы содержится в RFC 882 и RFC 883. В 1987 публикация RFC 1034 и RFC 1035 изменила спецификацию DNS и отменила RFC 882, RFC 883 и RFC 973 как устаревшие.

Когда сервис DNS-сервера запускается, то в оперативную память помещаются данные из всех зон. В памяти будет также храниться кэш DNS запросов. Полезно знать системные требования для DNS серверов:

1. DNS сервер без зон занимает порядка 4 Мб в оперативной памяти
2. При добавлении зон данные загружаются в оперативную память
3. Каждая запись занимает порядка 100 байт. Так если у вас 1000 записей это займет еще 100 кб.

4.2. DNS-запросы и разрешение имён

Система DNS функционирует по схеме рекурсивных или итеративных запросов, когда компьютер-клиент обращается за IP-адресом указанного пользователем доменного имени к DNS-серверу, явно указанному в свойствах его подключения к компьютерной сети или сообщаемому провайдером. В результате поиска DNS-сервер должен разрешить, то есть преобразовать доменное имя в IP-адрес и вернуть его клиенту.

При итеративном методе разрешения имён DNS-сервер выступает в роли клиента и опрашивает другие DNS-сервера в порядке убывания, начиная от корневых DNS-серверов и заканчивая последним, авторитетным за нужную DNS-зону. Рассмотрим, как работает данный метод:

1. Пользователь хочет получить доступ по имени www.inadmin.ru и отправляет запрос на свой DNS-сервер.
2. DNS сервер видит, что пришёл запрос, и у него в кэше нет ответа, какой IP-адрес у www.inadmin.ru.
3. Так как сервер не знает, где находится www.inadmin.ru, он обращается к корневому DNS-серверу, и спрашивает, где находится www.inadmin.ru.
4. Корневой DNS-сервер не знает, где хранятся записи для домена www.inadmin.ru, но знает, кто ответственный за домен первого уровня .ru и возвращает нашему DNS серверу его IP, например, 193.232.142.17.
5. Наш DNS сервер обращается к 193.232.142.17 с просьбой сообщить IP для www.inadmin.ru. Но этот DNS тоже не знает ничего про наш адрес. Но знает, что есть DNS-сервер, который отвечает за inadmin.ru и возвращает его IP, например, 195.128.64.3.
6. Наш DNS сервер обращается к 195.128.64.3 с просьбой сообщить IP для www.inadmin.ru. А вот этот сервер уже имеет нужную нам ресурсную запись, в которой указан IP-адрес, который мы ищем, и возвращает его нашему DNS-серверу.
7. Наш DNS сервер возвращает данный IP-адрес клиенту. Теперь клиент может подключиться по имени к серверу www.inadmin.ru.

При рекурсивном методе DNS-сервер просто пересылает данные другому DNS-серверу, чтобы тот выполнил всю работу (рекурсивно или итеративно) и вернул искомые данные, то есть возлагает задачу «хождения» по авторитетным DNS-серверам на своего «коллегу».

Кроме того, существует прямой и обратный DNS-запрос. Система DNS преобразовывает имена в IP-адреса и обратно. Обратное преобразование и осуществляется по обратному DNS-запросу. Для этого зарезервирован специальный домен in-addr.arpa, в котором хранятся PTR-записи. Октеты IP адреса хранятся в обратном порядке. Так для ip 1.2.3.4 будет создана запись вида 4.3.2.1.in-addr.arpa.

При запросе имени происходит несколько важных процедур, которые необходимо учитывать. Во-первых, данные о связке *имя - IP адрес* может храниться в нескольких местах (Hosts, DNS Cash, Lmhosts, DNS Server и др). Для того что бы полностью понимать принцип работы – нужно знать порядок, в котором Windows пытается разрешить любое имя.

1. При разрешении имени сверяется с локальным именем компьютера.
2. Если локальное имя не совпадает с запрашиваемым, то выполнятеся поиск в DNS Cash. В DNS-кэш динамически загружаются данные из файла HOSTS, поэтому поиск по файлу hosts не происходит, его данные всегда в памяти ПК, что ускоряет обработку. *Файл Hosts расположен в %systemroot%\System32\Drivers\Etc*
3. Если имя не разрешилось в IP адрес, то пересылается на DNS сервер, который задан в сетевых настройках.
4. Если имя сервера плоское (к примеру: server1) и не может быть разрешено с помощью DNS, то имя конвертируется в NetBIOS имя и ищется в локальном кэше.
5. Если имя не может разрешиться, то ищется на WINS серверах.
6. Если имя не может быть определено и на WINS сервере, то ищется с помощью BROADCAST запроса в локальной подсети.
7. Если имя не определилось, то ищется в файле LMHOSTS.

Поиск по всем 7-ми шагам прекращается, как только находится первое вхождение, удовлетворяющие условиям. Посмотреть кэш можно по команде ipconfig /displaydns. Очистить кэш можно по команде ipconfig /flushdns.

4.3. Ресурсные записи и DNS-зоны

Ресурсная запись является главной структурной единицей системы DNS, с помощью которой система выполняет свои функции. Фактически в ресурсных записях DNS-сервера содержатся все сведения, которые необходимы, чтобы дать ответ на поступивший DNS-запрос [9]. Рассмотрим основные виды ресурсных записей.

Запись A (address) – это главная адресная запись, необходимая для связи домена и IP-адреса сервера. Проще говоря, для работы сайта и всех поддоменов. Для протокола IPv4 используется запись A, для протокола IPv6 – запись AAAA.

Когда вы вводите название сайта в адресную строку браузера, именно по записи A служба доменных имён определяет, с какого сервера нужно открывать ваш сайт.

Примеры записи A:

Имя записи	Тип записи	Значение
site.ru	A	123.123.123.123
shop.site.ru	A	123.123.123.123

CNAME (Canonical name) — каноническое имя для псевдонима. Запись CNAME чаще всего используется для переадресации поддомена на другой домен. Примеры записи CNAME:

Имя записи	Тип записи	Значение
www.site.ru	CNAME	site.ru
webmail.site.ru	CNAME	webmail.hosting.reg.ru

В данном примере, если вы введете доменное имя webmail.site.ru в строку браузера, вы будете переадресованы на сайт webmail.hosting.reg.ru.

Одновременно добавить запись CNAME и запись A для одного и того же поддомена невозможно, т.е. нельзя добавить и запись A и запись CNAME для поддомена webmail.site.ru.

MX (Mail Exchanger) – адрес почтового шлюза для домена. Состоит из двух частей – приоритета и адреса узла. Записи MX критически важны для работы почты. Благодаря им отправляющая сторона «понимает» на какой сервер нужно отправлять почту для вашего домена.

Пример записи MX:

Имя записи	Тип записи	Приоритет	Значение
site.ru	MX	10	mx1.hosting.reg.ru
site.ru	MX	15	mx2.hosting.reg.ru

Записей MX может быть несколько. Делается это для того, чтобы в случае недоступности одного из почтовых серверов, почта была все же отправлена на другой. Приоритет записи определяет, на какой сервер нужно отправлять почту в первую очередь. Если приоритет одинаковый, сервер выбирается случайным образом.

NS (Authoritative name server) – адрес узла, отвечающего за доменную зону. Проще говоря, запись NS указывает, какие DNS-серверы хранят информацию о домене. Критически важна для работы службы DNS.

Обратная DNS-запись PTR связывает IP-адрес сервера с его каноническим именем (доменом). PTR-запись широко применяется в фильтрации почты. Для всех серверов виртуального хостинга REG.RU обратные DNS-записи уже прописаны. Если у вас виртуальный сервер VPS или выделенный сервер, прописать PTR-запись можно по инструкции

SOA (Start of Authority) – указывает, на каком сервере хранится эталонная информация о доменном имени. Критически важна для работы службы DNS.

SPF (Sender Policy Framework) – указывает сервера, которые могут отправлять почту от имени домена. Запись SPF вносят в TXT-запись домена.

Пример записи SPF:

Имя записи	Тип записи	Значение
site.ru	TXT	v=spf1 include:_spf.hosting.reg.ru ip4:37.140.192.92 a mx ~all

В данном примере:

- v=spf1 – определяет версию используемой записи SPF;
- include:_spf.hosting.reg.ru – включает в запись SPF значение SPF-записи другого домена. Т.е. для домена будут действовать в том числе все значения записи SPF для домена «_spf.hosting.reg.ru»;
- ip4:37.140.192.92 – разрешает прием почты с IP-адреса 37.140.192.92;
- a – разрешает приём почты с сервера, IP-адрес которого стоит в ресурсной A-записи домена. Проще говоря с сервера, где размещен сайт;
- mx – разрешает приём почты, если отправляющий сервер указан в одной из записей MX для домена;
- ~all – если письмо пришло с сервера, который не входит в вышеперечисленный список, его стоит проанализировать более тщательно. Также иногда используется –all – в этом случае письмо не проходит дополнительных проверок и сразу отвергается.

TXT (Text string) – содержит любую текстовую запись. Широко применяется для проверок на право владения доменом при подключении дополнительных сервисов, а также для записи SPF и ключа DKIM.

Записей TXT может быть сколько угодно. Вам может потребоваться добавить записи TXT при подключении бесплатной Яндекс.Почты и при получении бесплатного SSL-сертификата, и они не будут друг другу мешать, так как необходимы для подтверждения владения доменом в разных сервисах, в первом случае - в Яндекс, во втором - в сертификационном центре Global Sign.

Зоной (zone) называется часть пространства имен DNS со всеми ресурсными записями, за управление которой отвечает определенный сервер или группа серверов DNS. Она является в DNS основным механизмом для делегирования полномочий и применяется для установки границ, в пределах которых определенному серверу разрешено выполнять запросы. Любой сервер, который обслуживает какую-то определенную зону, считается полномочным или ответственным за эту зону.

Зона – это “зона ответственности” конкретного сервера доменных имен, т.е. понятие домена шире, чем понятие зоны. Если домен разбивается

на поддомены, то у каждого из них может появиться свой сервер, отвечающий за свою зону – в данном случае поддомен и поддомены более высокого уровня.

При этом зоной ответственности сервера более высокого уровня будет только та часть описания домена, которая не делегирована другим серверам. Разбиение домена на поддомены и организация сервера для каждого из них называется делегирование прав управления зоной соответствующему серверу доменных имен, или просто делегированием зоны.

Помимо того, что существуют зоны прямого и обратного просмотра для прямых и обратных DNS-запросов соответственно, существует 3 класса зон: основные, второстепенные и зоны-заглушки.

Основная зона используется в большинстве случаев и даёт возможность читать и создавать ресурсные записи. Обычно основные зоны передаётся на сервера второстепенных зон в первый раз целиком, а затем только изменения после последней синхронизации. Основная зона может храниться как в файле, так и в Active Directory.

Дополнительная зона не может храниться в Active Directory, в неё нельзя делать запись, и она используется для повышения отказоустойчивости DNS-зоны. Зона предоставляет возможность снизить объем трафика запросов DNS в областях сети, где происходит интенсивное запрашивание и использование данных зоны. Кроме того, в случае недоступности сервера, который управляет основной зоной, дополнительная зона может обеспечивать разрешение имен до тех пор, пока основной сервер снова не станет доступным. Наконец, зона-заглушка имеет только записи NS и SOA и служит для повышения эффективности разрешения имён.

Мастер создания новой зоны в Windows Server содержит следующие страницы конфигурации:

– тип зоны (Zone Type);

– область репликации зоны, интегрированной в Active Directory (Active Directory Zone Replication Scope);

– зона прямого или обратного просмотра (Forward or Reverse Lookup Zone);

– имя зоны (Zone Name);

– динамическое обновление (Dynamic Update).

На странице Область репликации зоны, интегрированной в Active Directory (Active Directory Zone Replication Scope) мастера создания новой зоны (New Zone Wizard) можно выбрать контроллеры домена в сети для сохранения данных зоны. Эта страница, появляется только при выборе опции сохранения зоны и Active Directory. Опции выбора области репликации зон определяют контроллеры домена, среди которых будет выполниться репликация данных зон.

Па этой странице представлены такие опции:

– сохранение зоны на всех контроллерах домена, которые также являются DNS-серверами во всем лесе Active Directory;

– сохранение зоны на всех контроллерах домена, которые также служит DNS-серверами и локальном домене Active Directory;

– сохранение зоны на всех контроллерах домена и локальном домене Active Directory (используется для совместимости с Windows 2000);

– Сохранение зоны на всех контроллерах домена, указанных и области настраиваемого раздела каталога Active Directory.

На странице Зона прямого или обратного просмотра (Forward or Reverse Lookup Zone) мастера создания новой зоны (New Zone Wizard) необходимо выбрать тип создаваемой зоны; зона прямого просмотра (Forward Lookup Zone) или зона обратного просмотра (Reverse Lookup Zone).

В зонах прямого просмотра DNS-серверы сопоставляют полные доменные имена FQDN с IP-адресами. В зонах обратного просмотра DNS-серверы сопоставляют IP-адреса именам FQDN. Таким образом, зоны прямого просмотра отвечают на запросы разрешения имен FQDN в IP-адреса, а зоны обратного просмотра отвечают на запросы разрешения IP-адресов в имена FQDN. Отметим, что зоны прямого просмотра получают имя в соответствии с доменными именами DNS, для которых выполняется разрешение, например, google.com. Зоны обратного просмотра именуются и обратном порядке первых трех октетов адресного пространства, для которого обеспечивается разрешение имен, плюс, дополнительный тег in-addr.arpa. Например, при разрешении имен для подсети 192.168.1.0/24 зона обратного просмотра получит имя 1.168.192.in-addr.arpa. В зоне прямого просмотра отдельная запись базы данных, сопоставляющая имя узла с адресом, называется записью *узел* (A). В зоне обратного просмотра отдельная запись базы данных, сопоставляющая IP-адрес, с именем узла, называется *указателем* или PTR-записью.

Для одновременного создания зон прямого и обратного просмотра можно использовать мастер настройки DNS-сервера (Configure A DNS Server Wizard).

На странице Имя зоны (Zone Name) мастера создания новой зоны (New Zone Wizard) можно выбрать имя создаваемой зоны прямого просмотра, Зоны обратного просмотра получают особые имена в соответствии с диапазоном IP-адресов, для которых являются полномочными.

Если зона создается для разрешения имен в домене Active Directory, лучше всего указать имя зоны, соответствующее имени домена Active Directory. Например, если организация содержит два домена Active Directory, с именами google.ru и translate.google.ru, инфраструктура разрешения имен должна включать две зоны с именами, соответствующими именам этих доменов.

В случае создания зоны для пространства имен DNS не в среде Active Directory, нужно указать имя Интернет-домена организации, например, wikipedia.org.

Чтобы добавить DNS-сервер на существующий контроллер домена, обычно добавляется копия основной зоны, обеспечивающая разрешение имен в локальном домене Active Directory. Для этого нужно просто создать зону, имя которой соответствует имени существующей зоны в локальном домене Active Directory. Новая зона будет заполнена данными с других DNS-серверов в домене.

Клиентские компьютеры DNS могут регистрировать и динамически обновлять свои записи ресурсов с помощью DNS-сервера. По умолчанию DNS-клиенты со статическими IP-адресами обновляют записи узлов (A или AAAA) и указателей (PTR), а DNS-клиенты, являющиеся DHCP-клиентами, - лишь записи узлов. В среде рабочей группы DHCP-сервер обновляет записи указателя от лица DHCP-клиента при каждом обновлении конфигурации IP.

Для успешного динамического обновления DNS зона, в которой клиенты регистрируют или обновляют записи, должна быть отконфигурирована для приема динамических обновлений. Существует два типа такого обновления:

– *безопасное обновление (Secure updates)* – позволяет выполнять регистрацию только с компьютеров домена Active Directory и обновление лишь с того компьютера, который изначально выполнял регистрацию.

– *небезопасные обновления (Nonsecure updates)* – позволяет выполнять обновление с любого компьютера.

На странице Динамическое обновление (Dynamic Update) мастера создания новой зоны (New Zone Wizard) для создаваемой зоны можно разрешить безопасные, небезопасные динамические обновления или вообще запретить обновление.

4.4. Роли DNS-серверов, уровни безопасности. Планирование пространства имён в корпоративной сети

DNS-сервера могут выполнять следующие виды ролей:

1. Cashing-only – не хранят на себе никаких зон, являются только серверами, где хранится кэш запросов. Поэтому они не создают трафик зоны. Такие сервера можно использовать в филиальном офисе для уменьшения DNS трафика между ним и главным офисом.

2. Non-recursive - сервера, на которых хранится DNS-зона, и у которых отключена возможность рекурсивного разрешения имени. Это приводит к тому, что если сервер не может разрешить имя (не имеет ресурсной записи), то DNS запрос будет не разрешён. Такие сервера можно ставить в роли внешних DNS серверов компаний. Также это защитит от использования внешними пользователями ваших DNS серверов для разрешения DNS имен в интернете.

3. Forward-only – сервера, которые занимаются только пересылкой запросов на другие сервера (обычный рекурсивный запрос отключён). В та-

ком случае, если сервер не получит ответа от других, то запрос будет не разрешён. Такие сервера можно использовать для управления DNS-трафиком между корпоративной сетью и интернетом. В таком сценарии все внутренние сервера будет обращаться к Forward-only серверу с просьбой разрешить внешние имена. Пятно контакта с интернет уменьшится до одного DNS сервера.

4. Conditional forwards – отличается от Forward-only тем, что задаётся связка какой домен на какой IP нужно пересылать.

Выделяют 3 уровня безопасности в DNS:

• Низкий уровень безопасности

1. Ваша DNS инфраструктура полностью выставлена в интернет.
2. Обычное разрешение имен DNS выполняют все сервера в вашей сети.
3. Все DNS сервера сконфигурированы на использование корневых серверов.
4. Все DNS сервера позволяют перемещение зоны на любые сервера.
5. Все DNS сервера «слушают» на всех своих IP.
6. Отключена очистка от старых записей в кэше.
7. Динамическое обновление разрешено для всех зон.
8. На пограничном брандмауэре пропускается DNS трафик в обе стороны.

• Средний уровень безопасности

1. Ваша DNS инфраструктура имеет ограниченный доступ в интернет
2. Все DNS сервера настроены на использование пересылки запросов на специальные серверы, когда они не могут разрешить имя локально.
3. Перемещении зоны разрешено только для своих NS серверов.
4. Сервера настроены «прослушивать» только на определенных IP.
5. Включена очистка «загрязнений» в кэше.
6. Общение между внутренним и внешними серверами происходит через брандмауэр, который частично ограничивает запросы. *Существует жёсткий список, от кого и кому разрешены DNS запросы.*
7. Внешние DNS серверы настроены на использование корневых серверов.

• Высокий уровень безопасности предполагает полное отсутствие взаимодействия с интернетом. Это не стандартная конфигурация, но она идеальна, если не нужен доступ в интернет.

1. Ваша DNS инфраструктура полностью не доступна из интернета.
2. Внутри сети используются DNS сервера, которые являются корневыми и хранят все адресное пространство.
3. Сервера, настроенные для пересылки запросов, используют только внутренние IP DNS серверов.
4. Перемещение зоны жёстко ограничено IP-адресами.
5. Сервера настроены прослушивать только на определенных IP.
6. Включена очистка загрязнений в DNS кэше.

7. Внутренние DNS сервера настроены на использование корневых серверов, прикреплённых к корневым внутренним DNS, на которых хранится корневая зона для вашего пространства имен.

8. Все DNS сервера хранятся на контроллерах домена и имеют ограниченный доступ (DACL).

9. Все зоны хранятся в Active Directory и имеют ограниченный доступ (DACL).

10. Безопасные динамические обновления разрешены за исключением верхнего уровня корневых зон.

При правильном планировании пространства DNS имен не будет проблем с разрешением этих имен. В текущее время каждая компания нуждается в связи с внешним миром. Что это означает для нас?

1. Внутренние имена DNS серверов и служб не должны быть доступны из интернета.

2. Внутренние сервера должны уметь разрешать внешние (интернет) имена.

3. Внешние пользователи должны иметь возможность разрешать внешние имена (к примеру, имя сайта, точка подключения VPN, Exchange OWA и т.д.).

Правильным решением будет расщепить структуру DNS на области действия (локальная сеть и интернет). Есть несколько типовых решений. Давайте их рассмотрим и решим, какие же выбрать.

• Одно пространство имен. К преимуществам можно отнести одно пространство имен для локальной сети и интернет. При этом разные DNS сервера отвечают за разные ресурсные записи. Если внутренний DNS используется для Active Directory и подобных ресурсов, то внешний DNS используется для WWW сайтов, VPN точки вхождения и т.д.

• Поддомен. Случай, когда для внутренней инфраструктуры мы выделяем из основного домена поддомен, что, в свою очередь

1. Проще в администрировании.

2. Делает понятной топологию сети

3. Внутреннее пространство имен остаётся невидимым для внешних запросов.

• Отдельное пространство имен. Похоже на второй случай, мы тоже разделяем пространство имен. Но в данном случае есть необходимость не разделять публичный домен. В таком случае создаётся отдельный домен для внутренних нужд, и отдельный для внешних.

4.5. Служба DHCP и технология NAT

Служба DHCP (Dynamic Host Configuration Protocol) – это одна из служб поддержки протокола TCP/IP, разработанная для упрощения администрирования IP-сети за счет использования специально настроенного сервера для централизованного управления IP-адресами и другими параметрами протокола TCP/IP, необходимыми сетевым узлам [10].

Работа протокола DHCP базируется на классической схеме клиент-сервер. В роли клиентов выступают компьютеры сети, стремящиеся получить IP-адреса в так называемую аренду (lease), а DHCP-серверы выполняют функции диспетчеров, которые выдают адреса, контролируют их использование и сообщают клиентам требуемые параметры конфигурации. Параметры выделяются клиенту на определенный срок, после чего считается свободным и может быть выдан другому клиенту.

Сервер поддерживает пул свободных адресов и, кроме того, ведет собственную регистрационную базу данных. Взаимодействие DHCP-серверов со станциями-клиентами осуществляется путем обмена сообщениями.

Для взаимодействия DHCP-сервера и DHCP-клиента используется специальный протокол DHCP, который является расширением протокола BOOTP (Bootstrap Protocol). DHCP устраняет определенные ограничения, которые BOOTP имел в качестве службы настройки узла.

Сервер DHCP избавляет сетевого администратора от необходимости ручного выполнения таких операций, как:

- автоматическое назначение сетевым узлам IP-адресов и прочих параметров протокола TCP/IP (например, маска подсети, адрес основного шлюза подсети, адреса серверов DNS и WINS);
- недопущение дублирования IP-адресов, назначаемых различным узлам сети;
- освобождение IP-адресов узлов, удаленных из сети;
- ведение централизованной БД выданных IP-адресов.

Во взаимодействии по протоколу DHCP принимают участие две или три стороны:

1. DHCP-клиент – тот, кто хочет получить параметры настройки TCP/IP;

2. DHCP-сервер – тот, кто выдаёт эти параметры;

3. DHCP-ретранслятор (relay agent) – вспомогательный участник, который может играть роль посредника между клиентом и сервером. Он используется в тех случаях, когда у клиента нет возможности обратиться к серверу напрямую, в частности, в том случае, если они находятся в разных широковещательных доменах. DHCP-ретранслятор обрабатывает стандартный широковещательный DHCP-запрос и перенаправляет его на DHCP-сервер в виде целенаправленного (unicast) пакета, а полученный от DHCP-сервера ответ, в свою очередь, перенаправляет DHCP-клиенту.

Как правило, DHCP-сервер выделяет IP-адреса (и прочие параметры TCP/IP) одним из двух способов:

1. Случайным образом из предопределённого пула (в том случае, если клиенту ранее уже выдавался какой-то адрес, он может попробовать получить его вновь);

2. Жёстко зафиксированным образом, исходя из MAC-адреса клиента.

В роли DHCP сервера может выступать сервер под управлением серверной ОС семейства Linux или Windows, некоторые модели коммутаторов и даже обычные компьютеры с клиентскими операционными системами, в случае если на них установлено специализированное программное обеспечение.

Протокол DHCP поддерживает три механизма выделения адресов: автоматический, динамический и ручной. В первом случае клиент получает постоянный IP-адрес, в последнем DHCP используется только для уведомления клиента об адресе, который администратор присвоил ему вручную.

Механизм получения динамического IP адреса клиентом DHCP клиентом достаточно прост, но требует более детального рассмотрения.

При включении компьютера, настроенного на автоматическое получение параметров сети, он выполняет широковещательный запрос на IP адрес 255.255.255.255, а в качестве своего IP адреса указывает 0.0.0.0 (так как у него еще нет IP адреса). В ходе данного широковещательного запроса рассылается сообщение DHCPDISCOVER, данное сообщение содержит в себе информацию, позволяющую отличить его от других типов запросов/сообщений (то-есть указывает на то, что это сообщение предназначено для DHCP сервера, для получения IP адреса), MAC адрес устройства, сформировавшего запрос, а также предыдущий IP адрес устройства (если он у него был).

Так как сообщение DHCPDISCOVER рассылается широковещательным способом, оно попадает не только на DHCP сервер, но и на другие устройства данного сегмента сети, но так как в сообщение DHCPDISCOVER указывается, что оно предназначено только для DHCP сервера, остальные устройства сети отвергают данное сообщение.

При получении сообщения DHCPDISCOVER DHCP сервером, он анализирует его содержание и в соответствии со своими настройками выбирает подходящую конфигурацию для запросившего компьютера и отправляет ее обратно в сообщении DHCPOFFER. Обычно сообщение DHCPOFFER отсылается только на MAC адрес компьютера, который был указан в сообщении DHCPDISCOVER, но иногда оно может рассылаться и методом широковещательной рассылки.

В случае если в сети существует несколько DHCP серверов компьютер может получить в ответ на сообщение DHCPDISCOVER несколько сообщений DHCPOFFER от разных DHCP серверов. Из них компьютер выбирает одно, обычно полученное первым. И отвечает на него сообщением DHCPREQUEST, которое содержит в себе всю туже информацию, что и

сообщение DHCPDISCOVER + IP адрес выбранного DHCP сервера. Сообщение DHCPREQUEST рассылается широковещательным методом, для того чтобы его могли получить все DHCP сервера сети, если их несколько.

Все устройства сети, не являющиеся DHCP серверами, игнорируют сообщение DHCPREQUEST. DHCP сервера, IP адрес которых не содержится в сообщении DHCPREQUEST понимают, что их не выбрали в качестве DHCP сервера. DHCP сервер IP адрес которого указан в сообщении DHCPREQUEST получает его и понимает, что именного его выбрали в качестве DHCP сервера для нового компьютера, на что он отвечает сообщением DHCPACK, которое как бы подтверждает данный выбор. Сообщение DHCPACK отправляется на MAC адрес компьютера, указанного в сообщении DHCPREQUEST.

Компьютер, запрашивающий конфигурацию, получает сообщения DHCPACK. И применяет конфигурацию, которая была получена в сообщении DHCPOFFER. Вот так путем несложного обмена сообщениями функционирует протокол DHCP.

DHCP сервер может быть настроен по-разному, и в зависимости от его конфигурации он будет выдавать IP адреса, запрашивающим компьютерам разными способами. Если на момент получения запроса DHCPDISCOVER сервер не располагает свободными IP-адресами, он может направить уведомление о возникшей проблеме администратору. В противном случае при выборе адреса обычно применяется следующий алгоритм. Клиенту выделяется адрес, записанный за ним в данный момент. Если это невозможно, сервер предложит адрес, которым пользовался клиент до окончания срока последней аренды (при условии, что данный адрес свободен), либо адрес, запрошенный самим клиентом при помощи соответствующей опции (опять же, если адрес не занят). Наконец, в том случае, когда все предыдущие варианты не проходят, новый адрес выбирается из пула доступных адресов с учетом подсети, из которой поступил клиентский запрос.

Исходя из определенной сетевым администратором политики и, соответственно, настроек сервер может выдать клиенту адрес, отличающийся от запрошенного (даже при доступности последнего), вообще отказать в предоставлении адреса или предложить адрес, относящийся к другой подсети. Более того, DHCP-сервер вообще не обязан реагировать на каждый поступивший запрос DHCPDISCOVER. Это предоставляет администратору возможность контролировать доступ к сети, например, разрешив серверу отвечать только тем клиентам, которые предварительно зарегистрировались с помощью специальной процедуры.

Спустя примерно половину этого срока клиент пытается возобновить его. Если клиент не может обновить аренду, он будет пытаться сделать это снова до окончания срока аренды. Если эти попытки не принесут успеха, клиент будет пытаться обратиться к другому DHCP-серверу. При обновлении аренды клиент проходит два состояния – обновления ад-

реса (RENEWING) и обновления конфигурации (REBINDING). Первое наступает примерно на половине срока аренды адреса (так называемый момент T1), второе – по истечении приблизительно 7/8 полного времени аренды (момент T2); для рассинхронизации процессов реконфигурирования разных клиентов значения этих временных меток рандомизируются с помощью случайной добавки.

В момент T1 клиент оправляет DHCP-серверу, выдавшему адрес, сообщение DHCPREQUEST с просьбой продлить срок аренды. Получив положительный ответ (DHCPACK), клиент пересчитывает срок аренды и продолжает работу в обычном режиме. Клиент ожидает прихода ответа от сервера в течение (T2 – t)/2 с (при условии, что это значение не меньше 60 с), где t – время отсылки последнего сообщения DHCPREQUEST, после чего отправляет данное сообщение повторно.

Если ответ от сервера не поступил к моменту T2, клиент переходит в состояние REBINDING и передает уже широковещательное сообщение DHCPREQUEST со своим текущим сетевым адресом. В этом случае моменты повторных выдач запросов DHCPREQUEST рассчитываются аналогично предыдущему случаю, только вместо T2 фигурирует время окончания срока аренды.

Не исключено, однако, что ответ DHCPACK не придет до окончания срока аренды. Тогда клиент обязан немедленно прекратить выполнение любых сетевых операций и заново начать процесс инициализации. Если запоздавший ответ DHCPACK все-таки поступит, клиенту рекомендуется сразу же возобновить работу под прежним адресом.

Невозобновленный IP-адрес возвращается в пул адресов. Если клиент связался с сервером, но текущий IP-адрес не может быть возобновлен, DHCP-сервер присваивает клиенту новый IP-адрес.

DHCP-сервер обычно не влияет на процедуру загрузки или входа в есть. Загрузка клиентов и регистрация пользователей возможна даже при нефункционирующем сервере DHCP.

Но самый главный плюс DHCP вовсе не в том, что с его помощью можно автоматически раздавать IP-адреса. На этом функционал протокола не заканчивается. Основная его ценность в другом: с его помощью вы можете назначать хостам и другие, не менее важные настройки. Например:

- Шлюзы по умолчанию. Если в вашей сети имеется несколько Интернет-каналов (для обеспечения бесперебойной работы), вы можете назначить хостам несколько шлюзов и порядок их предпочтения. В случае выхода одного из каналов из строя, переключение на резервный канал произойдет автоматически, без вашего вмешательства. Это же дает возможность организовать простейшую балансировку нагрузки между каналами, назначив по DHCP одной группе хостов один маршрутизатор в качестве шлюза, а другой группе – второй.
- Статические маршруты. Если в вашей сети есть несколько подсетей, соединенных маршрутизаторами, то при помощи DHCP можно авто-

матически оповещать хосты о наличии маршрутов в другие подсети. Причем это, по желанию, можно сделать только для избранных – например, используя привязку к MAC. Эта же опция полезна при организации VPN-доступа к корпоративной сети – VPN-клиентам можно сообщить маршруты лишь к нужным им подсетям, оставив другие подсети недоступными для подключающихся по VPN пользователей.

- Смещение времени. Если ваши пользователи часто бывают в различных временных поясах (например, мотаются из Питера во Владивосток и обратно), то можно заставить системные часы их ноутбука адаптироваться к вашему местному времени при помощи DHCP.
- Сервер синхронизации времени. Поскольку часы компьютеров славятся своей неточностью, их желательно синхронизировать с какими-то эталонными часами. Для этого используется служба NTP. Информацию о сервере NTP можно раздавать хостам при помощи DHCP.
- DNS-серверы. С помощью этой опции вы можете назначать вашим клиентам DNS-серверы как внутри сети, так и за ее пределами. Причем, в отличие ручной настройки интерфейса, вы можете передать хосту обширный список доступных DNS-серверов.
- Настройки сервера загрузки – настройки протокола TFTP/BOOTP, необходимые для бездисковой загрузки хостов. Эта возможность востребована при наличии в сети бездисковых терминалов, загружающихся по сети, и при организации дистанционной автоматической установки ОС на компьютеры пользователей (об этом поговорим отдельно)
- Списки доступных SMTP и POP серверов.
- Настройки WINS и Netbios
- Размер MTU, время жизни кэша ARP, размер TTL и др.

Если у вас сеть разбита на несколько подсетей, разделенных маршрутизаторами, то одним DHCP-сервером вам ограничиться не получится. DHCP-запросы и ответы не маршрутизируются между подсетями и распространяются в пределах лишь одного сегмента. Это связано в первую очередь с тем, что протокол DHCP не использует для передачи данных IP-адресацию, а работают на более низком уровне. Следовательно, в каждом из сегментов сети, имеющем свой диапазон IP-адресов, вам потребуется отдельная DHCP- служба.

Трансляция сетевых адресов (NAT) – технология преобразования адресов и/или портов источника и/или получателя IP-пакета. Эти преобразования отслеживаются в соответствующих таблицах, что позволяет при получении ответного пакета выполнить обратную трансляцию адреса/порта [11]. Технология NAT имеет два важных достоинства:

1. При NAT-адресации внутренние хосты могут совместно использовать один или несколько зарегистрированных внешних IP-адресов. При этом требуется относительно немного внешних адресов для поддержки большого числа внутренних хостов, что экономит IP-адреса.

2. NAT позволяет маскировать (скрыть) внутреннюю структуру локальной сети. За счёт выполнения трансляции адресов запросы компьютеров локальной сети к внешним хостам выглядят так, будто выполняются с одного и того же компьютера. Также можно с помощью трансляции адреса совместно с портом осуществлять сокрытие серверной инфраструктуры.

Важно отметить, что некоторые протоколы не поддерживают трансляцию адресов (например, PPTP), либо использование трансляции накладывает ограничения на использование некоторых служб. Для обхода таких проблем NAT-маршрутизатор должен иметь возможность разбирать пакеты, вносить изменения и собирать их снова

Различают два способа трансляции адресов:

1. Network Address Translation (NAT) – замена адреса источника на адрес маршрутизатора. При этом порт остаётся неизменным.

2. Static Address Translation (SAT) – замена адрес источника или приёмника на некоторый адрес, при этом возможна одновременная замена порта.

SAT подразделяется на два типа:

1. Source SAT – трансляция внутреннего адреса источника (*Inside local*) в зарегистрированный адрес источника (*Inside global*).

2. Destination SAT – трансляция внешнего адреса назначения (*Outside local)* в адрес назначения (*Outside global).*

Адресация NAT обычно функционирует на маршрутизаторе Cisco, соединяя две сети и транслируя частные локальные адреса внутренней сети в открытые зарегистрированные адреса внешней сети и обратно. Как показано на рисунке 11, внутреннему узлу требуется обменяться данными с внешним узлом (128.23.2.2).

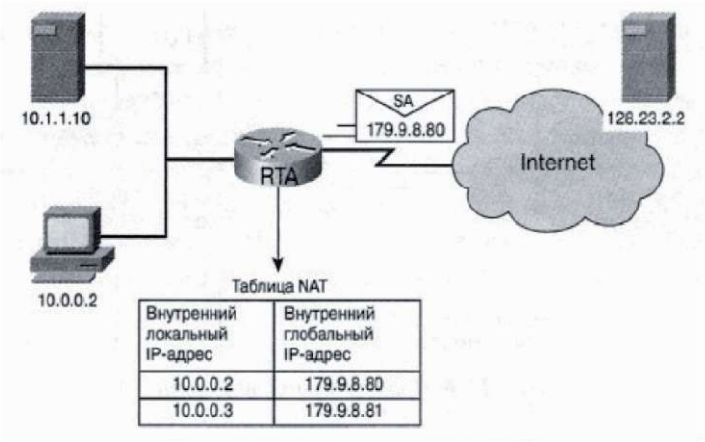


Рисунок 11 – Адресная таблица NAT

NAT выполняет преобразование локального адреса (10.0.0.2) в глобальный (179.9.8.80) и сохраняет это в своей NAT-таблице. Аналогично внутренний адрес (10.0.0.3) преобразуется в глобальный (179.9.8.81).

NAT принимает ответный пакет, направленный из внешней сети во внутреннюю, просматривает свою адресную таблицу для нахождения преобразования данного глобального адреса в локальный. Такая статическая NAT-адресация взаимного однозначного преобразования локальных и глобальных адресов (адрес - адрес) обычно используется для внутренних IP-узлов, которые должны быть постоянно доступны из внешей сети (например, Internet), таких как сервер DNS или сервер электронной почты (e-mail server).

Адресация NAT может быть сконфигурирована для представления только одного внешнего адреса для всей внутренней сети с помощью однозначного преобразования номеров портов (много адресов – один адрес с назначенным портом). Эта функция адресации NAT называется PAT (Port Address Translation). Такой способ эффективно скрывает внутреннюю структуру сети от внешней сети и повышает уровень безопасности.

Использование адресации NAT позволяет выполнить трансляцию ряда внутренних адресов, в то время как PAT может транслировать лишь один или несколько внешних адресов. Как показано на рисунке 12, хосты 10.0.0.2 и 10.0.0.3 посылают пакеты во внешнюю сеть, используя один IP-адрес 179.9.8.80 и разные порты.

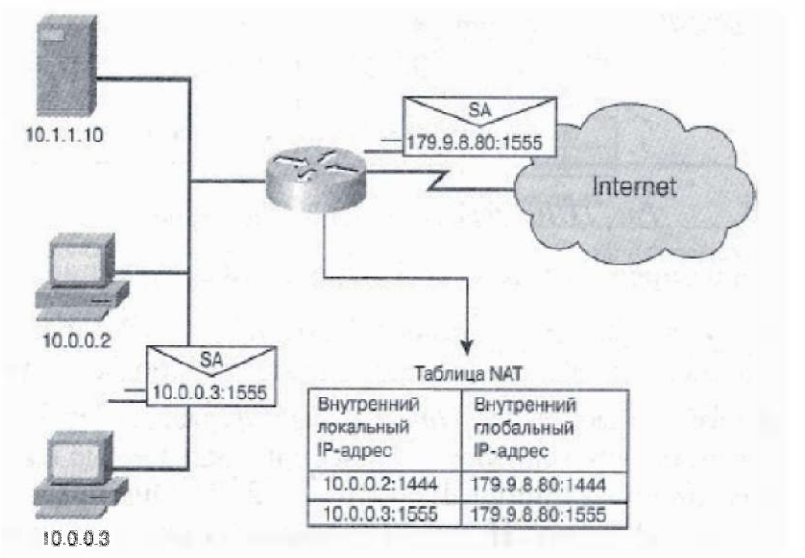


Рисунок 12 – Таблица NAT/PAT

Поскольку номер порта записывается двумя байтами, общее количество внутренних адресов, которые могут быть транслированы в один внешний адрес, при использовании PAT теоретически может достигать 65 536 для каждого IP-адреса. PAT пытается сохранить первоначальный порт источника. Если порт источника уже присвоен, то адресация PAT пытается найти первый доступный номер порта в соответствующей группе портов 0-511, 512-1023 или 1024-65535. Если в соответствующей группе нет доступных портов и конфигурируется более одного IP- адреса, то PAT переходит к следующему IP-адресу и пытается вновь найти первоначальный порт источника. Это процесс продолжается до тех пор, пока PAT не исчерпает доступные порты и внешние IP-адреса.

Известная транснациональная компания по разработке и продаже сетевого оборудования Cisco определила для NAT-адресации следующие термины.

Внутренние локальные адреса (Inside local address) – IP-адреса, назначенные хосту во внутренней сети, соответствующие RFC 1918.

Внутренние глобальные адреса (Inside global address) – зарегистрированные IP-адреса, назначаемые провайдером службы или выделяемые из регионального регистра Internet (Regional Internet Registries, RIR). Они предоставляют один или более внутренних локальных IP-адресов для связи с внешней сетевой средой.

Внешние локальные адреса (Outside local address) – IP-адреса внешних узлов, в том виде как они известны узлам внутренней сети.

Внешние глобальные адреса (Outside global address) – IP-адрес, назначаемый владельцем узла, этому узлу для использования во внешней сети.

Вопросы конфигурирования NAT включают статическую и динамическую трансляцию, а также перезагрузку NAT (PAT). Под статической трансляцией понимается ручное конфигурирование адресов в просмотровой таблице. Для каждого внутреннего локального адреса при использовании статической NAT требуется внутренний глобальный адрес. Для того, чтобы сконфигурировать статическую трансляцию внутреннего адреса, требуется выполнить действия, описанные ниже.

1. Задать статическую трансляцию внутреннего локального адреса во внутренний глобальный адрес.

Router (config) #ip nat inside source static *local-ip global-ip*

2. Задать внутренний интерфейс и указать его, как принадлежащий к внутренней сети

Router (config) #interface *type number*

Router (config-if) #ip nat inside

3. Задать выходной интерфейс и указать его, как подсоединенный извне

Router (config) #interface *type number*

Router(config-if) #ip nat outside

Пример статической NAT-адресации показан на рис. 13.

При использовании динамической трансляции адресов преобразования адресов не существуют в NAT-таблице до тех пор, пока маршрутизатор не получит данные, для которых такая трансляция требуется. Динамические преобразования адресов являются временными и в конечном итоге устаревают и удаляются. Для того, чтобы сконфигурировать трансляцию внутренних адресов, следует выполнить действия, описанные ниже.

```
hostname CM
!
ip nat inside source static 10.1.1.2 192.168.1.2
!
interface Ethernet0
ip address 10.1.1.1 255.255.255.0
ip nat inside
!
interface Serial0
ip address 192.168.1.1 255.255.255.0
ip nat outside
!
ip nat inside source static 10.1.1.2 192.168.1.2
```

Рисунок 13 – Статическая NAT-адресация

1. Задать пул глобальных адресов, которые будут использоваться по мере необходимости.

Router (config)#ip nat pool *имя нач-ip конеч-ip* {netmask *маска* | prefix-length *длина-префикса*}

2. Создать стандартный список доступа для идентификации тех адресов, которые нужно будет транслировать.

Router (config)#access-list *номер-списка* permit *источник* [шаблон-источник]

3. Сконфигурировать динамический NAT на основе адресов источника.

Router (config)#ip nat inside source list *номер-списка-дост* pool *имя*

4.Указать внутренний интерфейс

Router (config)#interface *type number*

Router (config-if)#ip nat inside

5. Указать внешний интерфейс

Router (config)#interface *type number*

Router(config-if)#ip nat outside

Список доступа должен определять только те адреса, которые следует транслировать. Следует помнить о том, что неявная команда deny all присутствует в каждом списке доступа. Недостаточно строгий список доступа может привести к непредсказуемым результатам. Рекомендуется не конфигурировать списки доступа, на которые ссылаются команды NAT с permit

any (т.е. разрешить трансляцию для всех). Использование permit any может привести к тому, что NAT будет потреблять слишком много ресурсов маршрутизатора, что может вызвать проблемы в сети.

Приведенные ниже команды конфигурируют соответствующие интерфейсы для выполнения внутренних и внешних функций.

Пример динамической NAT-адресации:

```
ip nat pool nat-pool1 179.9.8.80 179.9.8.95 netmask 255.255.255.0
ip nat inside source list 1 pool nat-pool1
!
interface fastethernet0/0
ip address 10.1.1.1 255.255.255.0
ip nat inside
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.0
ip nat outside
!
```

Рисунок 14 – Динамическая NAT-адресация

В примере происходит трансляция всех адресов, проходящих через список доступа 1 (имеющие адрес источника от 10.0.0.0/16) в адрес из пула с именем *nat-pool*. Этот пул содержит адреса из диапазона от 179.9.8.80/28 до 179.9.8.95/28.

Одной из наиболее мощных функций NAT является способность использовать PAT. Это иногда называется NAT-адресацией "много-в-один" или перегрузкой адреса. При использовании перегрузки *(overloading)* сотни узлов с частными адресами могут получать доступ к внешней сети (Internet), используя лишь один глобальный адрес. NAT-маршрутизатор отслеживает различные сеансы связи, устанавливая соответствие TCP и UDP номеров портов в таблице трансляции.

Для того, чтобы сконфигурировать перегрузку внутренних глобальных адресов, следует выполнить действия, описанные ниже.

1. Определить стандартный список доступа, разрешающий те адреса, которые должны транслироваться: Router (config)#access-list *номер-списка* permit *источник* [шаблон-источник]

2. Сконфигурировать динамический NAT на основе адресов источников, указать список доступа, определенный на предыдущем этапе. Router

(config)#ip nat inside source list *номер-списка-дост* interface *интерфейс* overload

3. Задать набор глобальных адресов, которые будут использоваться для перезагрузки.

Router (config)#ip nat pool *имя ip-адрес* {netmask *маска* | prefix-length *длина-префикса*}

4. Задать трансляцию с перезагрузкой.

Router (config)#ip nat inside source list *номер-списка-дост* pool *имя* overload

5.Указать внутренний интерфейс

Router (config)#interface *type number*

Router (config-if)#ip nat inside

6. Указать внешний интерфейс

Router (config)#interface *type number*

Router(config-if)#ip nat outside

Пример перезагрузки NAT

1. Перезагрузка на интерфейсе.

```
Router(config)#access-list 1 permit 10.0.0.0 0.0.255.255
Router(config)#ip nat inside source list 1 interface serial0/0 overload
Router(config)#interface s0
Router(config-if)#ip nat outside
Router(config-if)#interface ethernet 0
Router(config-if)#ip nat inside
```

2. Перезагрузка с использованием пула.

```
Router(config)#access-list 1 permit 10.0.0.0 0.0.255.255
Router(config)#ip nat pool nat-pool2 179.9.8.20 netmask
255.255.255.240
Router(config)#ip nat inside source list 1pool nat-pool2 overload
Router(config)#interface s0
Router(config-if)#ip nat outside
Router(config-if)#interface ethernet 0
Router(config-if)#ip nat inside
```

Для тестирования функций NAT используется команда debug ip nat, которая отображает информацию обо всех пакетах, транслируемых маршрутизатором. По команде debug ip nat *detailed* выводится описание каждого пакета, для которого предполагается трансляция. При этом также выводится информация об определенных ошибках или исключительных условиях, таких, например, как невозможность выделить глобальный адрес. Команда debug сильно нагружает маршрутизатор – не злоупотребляйте ей и отключайте её, как только закончили поиск ошибок.

Контрольные вопросы

1. Что такое DNS?
2. В чём смысл использования DNS?
3. Назовите основные характеристики DNS.
4. Что такое поддомен?
5. Что такое ресурсная запись?
6. Что такое зона?
7. Что означает авторитетность?
8. Что такое делегирование?
9. Какое имя имеет корневой домен?
10. Что такое разрешение имён?
11. Что такое рекурсивный DNS-запрос?
12. Что такое итеративный DNS-запрос?
13. Опишите схему работы итеративного запроса.
14. Назовите назначение записи A.
15. Назовите назначение записи CNAME.
16. Назовите назначение записи NS.
17. Назовите назначение записи MX.
18. Назовите назначение записи PTR.
19. Назовите назначение записи SOA.
20. Что такое зона?
21. Чем отличается зона прямого просмотра от зоны обратного просмотра?
22. Чем отличается основная зона от дополнительной?
23. Чем отличается зона-заглушка от основной и дополнительной?
24. Какие роли может выполнять DNS-сервер?
25. Опишите характеристики низкого уровня безопасности в DNS.
26. Опишите характеристики среднего уровня безопасности в DNS.
27. Опишите характеристики высокого уровня безопасности в DNS.
28. Какие существуют правила планирования пространства имён в корпоративной сети?
29. Что такое DHCP?
30. Опишите схему работы DHCP при запросе аренды IP-адреса.
31. Что такое NAT?
32. Какие вы знаете способы трансляции IP-адресов?
33. Что такое внутренние локальные адреса?
34. Что такое внутренние глобальные адреса?
35. Что такое внешние локальные адреса?
36. Что такое внешние глобальные адреса?

Задание

В консоли управления DNS-сервером в диспетчере серверов откройте зону прямого просмотра и создайте псевдоним для сервера. Затем создайте зону обратного просмотра и добавьте запись PTR для сервера. Откройте Windows PowerShell и проверьте при помощи утилиты nslookup, «отзывается» ли сервер на созданный псевдоним, и работает ли обратный dns-запрос?

Добавьте роль DHCP-сервера и создайте произвольный пул из 100 IP-адресов, в который входит IP-адрес сервера. При этом выдачу адреса, используемого сервером, заблокировать.

5. УДАЛЁННОЕ АДМИНИСТРИРОВАНИЕ

Удаленное администрирование избавляет от необходимости постоянного присутствия в непосредственной близости от настраиваемого оборудования, тем самым снижая накладные расходы на содержание штатного системного администратора.

Управление компьютером или сервером производится по сети, в том числе и через интернет. Доступ к компьютеру, или серверу осуществляется с помощью специального программного обеспечения по защищенным каналам связи, гарантирующим безопасность передаваемой информации между компьютером или сервером клиента и администратора из службы поддержки. Общение администратора с клиентом происходит по Skype, или телефону.

Удаленное администрирование – легко реализуемая задача, которая позволяет решить большинство возникающих вопросов в процессе использования компьютеров, серверов и программного обеспечения. Основное условие, соблюдение которого необходимо для успешного удаленного администрирования - это отсутствие проблем со связью. Если нарушена работа сети, соответственно не будет возможности подключиться к удаленной машине клиента.

Вопросы, которые решают, используя удалённое администрирование:

- установка, обновление и настройка программного обеспечения;
- настройка рабочего окружения пользователя и параметров ОС;
- проведение антивирусной профилактики, удаление вирусов и шпионского (вредоносного и нежелательного) ПО;
- администрирование бухгалтерских программ, в том числе специализированных (1С, Банк-Клиент и пр.);
- проведение консультаций по работе в прикладных программах;
- осуществление диагностики и устранение программных сбоев операционной системы и программного обеспечения
- решения других срочных вопросов.

Программы удалённого администрирования – это программы или функции операционных систем, позволяющие получить удалённый доступ к компьютеру через Интернет или ЛВС и производить управление и администрирование удалённого компьютера в реальном времени. Программы удалённого администрирования предоставляют почти полный контроль над удалённым компьютером: они дают возможность удалённо управлять рабочим столом компьютера, возможность копирования или удаления файлов, запуска приложений и т. д.

Существует множество реализаций программ удалённого администрирования. Все реализации отличаются по интерфейсу и используемым протоколам. Интерфейс может быть визуальный или консольный. Одними из самых популярных и распространённых программ являются, например, компонент Windows Remote Desktop Services с клиентом Remote Desktop

Connection, Radmin, DameWare, PuTTy, VNC, UltraVNC, Apple Remote Desktop, Hamachi, LiteManager, TeamViewer, AnyDesktop, Remote Manipulator System, Ammyy Admin и др.

Собственно, для цели передачи команд администрирования и вывода экрана используются протоколы удалённого администрирования: RDP, VNC, X11, Telnet, Rlogin, RFB, ARD, ICA, ALP и собственные. Для шифрования трафика в программах удалённого администрирования используются протоколы SSH, SSL, TLS и др.

Главное преимущество удаленного администрирования – это увеличение скорости реакции обслуживающего ИТ персонала на проблемы пользователей. ИТ специалисту не нужно идти в другой конец офиса, или ехать в другой конец города (если офис распределен по нескольким точкам). Для решения поставленной задачи достаточно подключиться к серверу, или компьютеру пользователя и авторизоваться в системе. Вся эта процедура занимает не больше тридцати секунд.

Наиболее популярной прикладной программой удалённого администрирования является пакет TeamViewer, работающий на платформах Windows, Linux, Mac OS X, а также мобильных платформах iOS и Android. Согласно пресс-релизам компании-разработчика «TeamViewer GmbH», TeamViewer используется более чем на 15 000 000 компьютеров, работающих в пятидесяти странах мира.

TeamViewer может работать с установкой или без неё – в последнем случае программа работает без администраторских прав доступа. Для установления связи TeamViewer должен быть запущен на обеих машинах. При запуске TeamViewer создаётся ID компьютера и пароль. Чтобы установить связь между компьютерами, клиент-оператор должен связаться с удаленным оператором и узнать его ID и пароль, а затем ввести их в клиент-TeamViewer.

TeamViewer также может установить связь с удалённым компьютером, используя браузер с технологией Flash.

TeamViewer позволяет устанавливать VPN (Virtual Private Network) соединения между клиентом и сервером. Есть возможность скачать с сайта производителя отдельные модули программы (клиентский и серверный). Можно также на сайте производителя сконфигурировать клиентский модуль с заранее предустановленным паролем доступа и собственным логотипом, скомпилировать и сразу скачать его. Однако без лицензии связь в этом случае возможна не более 5 минут за сеанс. Предлагаемые модули без собственных предустановок не имеют таких ограничений.

Модули не требуют инсталляции и просты в использовании. Возможен видео-, голосовой, и текстовый чат между компьютерами. Последняя на данный момент 14 версия программы работает быстрее, чем раньше, поскольку оптимизирована для работы с низкой пропускной способностью. Программа автоматически обнаруживает медленные соединения с

помощью интеллектуального адаптивного сжатия, что повышает скорость и надежность работы.

Таким образом, пользователю больше не нужно быстрое подключение к Интернету, чтобы начать использовать программу. Эта версия обещает улучшенное качество соединения с уменьшенной задержкой.

Контрольные вопросы

1. Что подразумевается под удалённым администрированием?
2. Перечислите вопросы, которые решаются с помощью удалённого администрирования?
3. Какие Вы знаете программы удалённого администрирования?

Задание

Установите на стационарный компьютер и на ноутбук или другое мобильное устройство программу TeamViewer и настройте удалённое управление компьютером с мобильного устройства.

6. АДМИНИСТРИРОВАНИЕ СЕРВЕРА БАЗ ДАННЫХ

6.1. Задачи администрирования баз данных. Платформа MS SQL Server и её инструменты

Администрирование баз данных (БД) – одна из функций системного администратора, на выполнение которой часто назначается свой администратор. У такого сотрудника свои функции и свои основные задачи:

1. Проектирование базы данных.
2. Оптимизация производительности базы данных.
3. Обеспечение безопасности в базе данных.
4. Резервное копирование и Восстановление базы данных.
5. Обеспечение целостности баз данных.
6. Обеспечение перехода на новую версию СУБД.

По действующему стандарту для администратора базы данных задачи и должностные обязанности определяются, в зависимости от уровня квалификации, из следующего списка:

1. Обеспечение функционирования БД, в которое входит:

- Резервное копирование БД
- Восстановление БД
- Управление доступом к БД
- Установка и настройка программного обеспечения (ПО) для обеспечения работы пользователей с БД
- Установка и настройка ПО для администрирования БД
- Мониторинг событий, возникающих в процессе работы БД
- Протоколирование событий, возникающих в процессе работы БД

Оптимизация функционирования БД

- Мониторинг работы БД, сбор статистической информации о работе БД
- Оптимизация распределения вычислительных ресурсов, взаимодействующих с БД
- Оптимизация производительности БД
- Оптимизация компонентов вычислительной сети, взаимодействующих с БД
- Оптимизация выполнения запросов к БД
- Оптимизация управления жизненным циклом данных, хранящихся в БД

2. Предотвращение потерь и повреждений данных, включающее следующие подзадачи:

- Разработка регламентов резервного копирования БД
- Контроль выполнения регламента резервного копирования
- Разработка стратегии резервного копирования БД
- Разработка регламентов восстановления БД

- Разработка автоматических процедур для создания резервных копий БД
- Проведение процедуры восстановления данных после сбоя
- Контроль соблюдения регламента восстановления
- Анализ сбоев в работе БД и выявление их причин
- Разработка методических инструкций по сопровождению БД
- Мониторинг работы программно-аппаратного обеспечения БД
- Настройка работы программно-аппаратного обеспечения БД
- Подготовка предложений по модернизации программно-аппаратных средств поддержки БД
- Прогнозирование и оценка рисков сбоев в работе БД
- Разработка автоматических процедур для горячего резервирования БД
- Выполнение процедур по вводу в рабочий режим ресурсов горячей замены
- Подготовка отчетов о функционировании БД
- Консультирование пользователей в процессе эксплуатации БД
- Подготовка предложений по повышению квалификации сотрудников

3. Обеспечение информационной безопасности на уровне БД, состоящее в следующем:

- Разработка политики информационной безопасности на уровне БД
- Контроль соблюдения регламентов по обеспечению безопасности на уровне БД
- Оптимизация работы систем безопасности с целью уменьшения нагрузки на работу БД
- Разработка регламентов и аудит системы безопасности данных
- Подготовка отчетов о состоянии и эффективности системы безопасности на уровне БД
- Разработка автоматизированных процедур выявления попыток несанкционированного доступа к данным

4. Управление развитием БД

- Анализ системных проблем обработки информации на уровне БД, подготовка предложений по перспективному развитию БД
- Разработка регламентов обновления версий программного обеспечения БД
- Разработка регламентов по миграции БД на новые платформы и новые версии ПО
- Изучение, освоение и внедрение в практику администрирования новых технологий работы с БД
- Контроль обновления версий БД
- Контроль миграции БД на новые платформы и новые версии ПО

• Планирование организационной структуры подразделения и развития кадрового потенциала

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Платформа Microsoft SQL Server обладает следующими основными инструментами [12]:

- Аналоги SSMS
- Работа с бэкапами
- Средства мониторинга и оповещений
- Встроенные инструменты SQL Server (например, bcp)
- Инструменты тестирования
- Генерация тестовых данных
- Средства по созданию документации
- Поисковые утилиты
- Инструменты по сравнению данных
- Инструменты по сравнению структуры базы данных
- Управление заданиями
- Управление индексами
- Работа со статистикой
- Проектирование баз данных

MS SQL Server включает средства управления для развитого управления и настройки баз данных, так же, как и тесную интеграцию с такими инструментами, как Microsoft Operations Manager (MOM) и Microsoft Systems Management Server (SMS).

Стандартные протоколы доступа к данным существенно уменьшают время, необходимое для интеграции данных SQL Server с существующими системами.

Поддержка Web-служб позволяет обеспечить взаимодействие с другими приложениями и платформами.

MS SQL Server предлагает интегрированные инструменты разработки для ядра базы данных, извлечения, трансформации и загрузки данных, извлечения информации, OLAP и отчётности, которые тесно интегрированы с Microsoft Visual Studio® для предоставления сквозных возможностей разработки приложений.

6.2. Обеспечение отказоустойчивости сервера баз данных

Зеркальное отображение базы данных – это решение, нацеленное на повышение доступности базы данных SQL Server. Зеркальное отображение каждой базы данных осуществляется отдельно и работает только с теми базами данных, которые используют модель полного восстановления.

Зеркальное отображение базы данных — это простая стратегия обеспечения надежности, имеющая следующие преимущества:

- Повышает доступность базы данных.
- Повышает защиту данных.
- Повышает доступность рабочей базы данных при обновлениях.

Большинству предприятий требуются решения, обеспечивающие высокую доступность всего экземпляра SQL Server, а не только отдельных БД. Чтобы удовлетворить это требование, можно включать экземпляры SQL Server 2008 в кластеры Microsoft Cluster Service. Клиенты воспринимают кластер с восстановлением после сбоев как одиночный экземпляр SQL Server 2008, однако при отказе одного из серверов такого кластера осуществляется восстановление после сбоя, и нагрузка распределяется по другим серверам кластера.

Некоторые ограничения прежних версий SQL Server не позволяли в полной мере воспользоваться преимуществами кластерных решений. В частности, кластерные решения должны были использовать одну букву диска для каждого экземпляра SQL Server, а все узлы кластера должны были относиться к одной и той же подсети. В кластерных серверах SQL Server 2008, созданных на базе Windows Server 2008 (кодовое имя «Longhorn»), эти ограничения отсутствуют, делая возможной более гибкую конфигурацию кластеров.

Кроме того, для обслуживания критически важных приложений и очень больших сред допускается создавать кластеры SQL Server 2008 с числом узлов до 16 (при использовании поддержки кластеров Windows Server 2008).

Отказоустойчивый кластер (англ. High-Availability cluster, HA cluster — кластер высокой доступности) – кластер (группа серверов), спроектированный в соответствии с методиками обеспечения высокой доступности и гарантирующий минимальное время простоя за счёт аппаратной избыточности. Без кластеризации сбой сервера приводит к тому, что поддерживаемые им приложения или сетевые сервисы оказываются недоступны до восстановления его работоспособности. Отказоустойчивая кластеризация исправляет эту ситуацию, перезапуская приложения на других узлах кластера без вмешательства администратора в случае обнаружения аппаратных или программных сбоев. Процесс перезапуска известен как аварийное переключение. В рамках этого процесса программное обеспечение кластеризации может дополнительно настроить узел перед запуском приложения на нём (например, импортировать и смонтировать соответствующие файло-

вые системы, переконфигурировать сетевое оборудование или запустить какие-либо служебные приложения).

Отказоустойчивые кластеры широко используются для поддержки важных баз данных, хранения файлов в сети, бизнес-приложений и систем обслуживания клиентов, таких как сайты электронной коммерции.

Реализации HA-кластеров представляют собой попытки достигнуть отказоустойчивости кластера в целом путём исключения критических точек отказа, в том числе за счёт резервирования вычислительных мощностей, сетевых подключений и хранилищ данных, объединённых в избыточную сеть хранения данных.

Моментальный снимок базы данных является статичным, доступным только для чтения представлением базы данных SQL Server (базы данных-источника). Моментальный снимок базы данных согласуется на уровне транзакций с базой данных-источником в момент создания моментального снимка. Моментальный снимок базы данных всегда находится на том же экземпляре сервера, что и база данных-источник. При обновлении базы данных-источника обновляется и моментальный снимок базы данных. Это означает, что чем дольше существует моментальный снимок базы данных, тем больше вероятность того, что он израсходует все доступное место на диске.

Может существовать несколько моментальных снимков одной и той же базы данных-источника. Каждый моментальный снимок базы данных существует до тех пор, пока он не будет явно удален владельцем базы данных.

Индекс является структурой на диске, которая связана с таблицей или представлением и ускоряет получение строк из таблицы или представления. Индекс содержит ключи, построенные из одного или нескольких столбцов в таблице или представлении. Эти ключи хранятся в виде структуры сбалансированного дерева, которая поддерживает быстрый поиск строк по их ключевым значениям в SQL Server.

Опция оперативного индекса позволяет одновременно производить модификацию (вставки, изменения, удаления) таблицы или данных кластерного индекса и любых связанных индексов во время выполнения команды DDL.

В MS SQL Server существует возможность выполнять операцию восстановления во время работы экземпляра SQL Server.

Возможность оперативного восстановления улучшает доступность SQL Server, так как недоступны только восстанавливаемые данные. Остальная часть базы данных остаётся доступной.

Компонент SQL Server Database Engine автоматически изменяет индексы при вставке, обновлении или удалении базовых данных. Со временем эти изменения могут привести к тому, что данные в индексе окажутся разбросанными по базе данных (фрагментированными). Фрагментация имеет место в тех случаях, когда в индексах содержатся страницы, для ко-

торых логический порядок, основанный на значении ключа, не совпадает с физическим порядком в файле данных. Существенно фрагментированные индексы могут серьезно снижать производительность запросов и служить причиной замедления отклика приложения, особенно операций сканирования.

Можно устранить фрагментацию путем реорганизации или перестроения индекса. Для секционированных индексов, построенных на основе схемы секционирования, можно использовать любой из этих методов для всего индекса или отдельной его секции.

При перестроении старый индекс удаляется, и создается новый. Таким образом, устраняется фрагментация, восстанавливается место на диске путем сжатия страниц с учетом указанного или существующего коэффициента заполнения, переупорядочиваются индексные строки в последовательных страницах. Если ALL указано, то все индексы в таблице удаляются и перестраиваются в ходе одной транзакции.

6.3. Интегрированная платформа для работы с интеллектуальными ресурсами предприятия

В современном мире, данные и системы, управляющие данными, должны быть постоянно защищены и доступны пользователям. MS SQL Server включает главные улучшения управления данными предприятия в следующих областях:

- Управляемость
- Доступность
- Масштабируемость
- Безопасность

MS SQL Server предоставляет единую консоль управления, которая позволяет администраторам данных, отслеживать, управлять и настраивать все базы данных и связанные службы по всему предприятию.

Серверное ПО предоставляет расширяемую инфраструктуру управления средствами SQL Management Objects (SMO), позволяя пользователям переделывать и расширять их среду управления.

SQL Server упрощает управление средствами единой интегрированной консоли управления для мониторинга и управления реляционной базой данных SQL Server, IntegrationServices, AnalysisServices, ReportingServices, NotificationServicesи SQL Mobilена большом числе распределённых серверов и баз данных.

Администратор баз данных может выполнять следующие задачи:

- создание и выполнение запроса;
- просмотр серверных объектов;
- управление объектом;
- отслеживание активности системы;
- просмотр оперативной справки.

Возможности работы с интеллектуальными ресурсами предприятия в MS SQL Server улучшены в следующих областях:

- Integration Services
- Analysis Services
- Reporting Services
- Интеграция с Microsoft Office System

Эти возможности реализованы в интегрированной платформе для работы с интеллектуальными ресурсами предприятия.

Набор BI инструметов SQL Server предоставляет сквозную интеграцию BI приложений. Business Intelligence Development Studio является первой интегрированной средой разработки, созданной для разработчиков BI.

SQL Server Integration Services (SSIS) позволяют выполнять сложную интеграцию данных, преобразования и синтеза на высокой скорости для очень больших объёмов данных.

Модули Integration Services, Analysis Services и Reporting Services взаимодействуют для предоставления цельного вида данных, полученных из разнородных источников.

Data Mining включает новые алгоритмы, включая правило ассоциации, временные ряды, регрессионные деревья, кластеризацию последовательностей, нейронные сети, простой Байес.

Reporting Services расширяют платформу BI Microsoft до уровня потребителей, которые используют результаты анализа. Reporting Services являются управляемой средой отчётов предприятия, встроенной и управляемой через Web службы. Отчёты могут быть персонализированы и доставлены во множестве форматов, с диапазоном интерактивных опций и опций печати.

SQL Server Management Studio объединяет в себе функции управления всеми компонентами SQL Server.

6.4. Обеспечение безопасности данных

MS SQL Server имеет улучшенную модель безопасности платформы БД, с возможностью предоставить более точный и гибкий контроль для обеспечения безопасности данных, характеризующуюся:

- применением политик для паролей учётных записей SQL Server в области аутентификации;
- обеспечением большей модульности для указания разрешений на различных уровнях в области авторизации;
- разделением владельца и схемы в области управления безопасностью.

Прежде чем перейти к средствам обеспечения отказоустойчивости, перечислим основные причины повреждений баз данных:

- отключение питания сервера;
- дефекты оборудования.

- сбои самого сервера;
- повреждения индексов;
- повреждения таблиц;
- стихийные и техногенные бедствия;
- вредоносные программы;
- человеческий фактор.

Рассмотрим существующие средства обеспечения отказоустойчивости. Oracle RAC – технология, обеспечивающая работу нескольких серверов Oracle с одной базой данных, при этом на каждом сервере функционирует отдельный экземпляр СУБД, однако все экземпляры работают с одним и тем же файлом данных. Такая структура позволяет обеспечить высокую доступность и повысить производительность за счет того, что запросы распределяются по нескольким серверам (узлам кластера). Выход из строя одного узла не приводит к недоступности базы данных, а наличие нескольких параллельно работающих узлов с динамическим распределением нагрузки позволяет горизонтально масштабировать производительность СУБД;

Oracle Data Guard представляет собой технологию синхронизации данных из основной базы в резервную базу-копию, при этом может быть обеспечено оперативное отражение изменений в режиме реального времени. Данная технология защищает от выхода из строя дискового массива основной базы данных и позволяет быстро переключиться на резервную базу данных, без существенных потерь времени. В Oracle Database 11g, в технологию Oracle Data Guard внесен ряд усовершенствований, в частности: возможность запускать запросы на чтение в режиме реального времени на резервной системе, возможность проводить обновления баз данных в онлайн режиме, временно переводить резервную базу данных в режим snapshot standby для проведения на ней тестов;

IBM DB2 pureScale – технология, предусматривающая доступ нескольких серверов СУБД (аппаратных или виртуальных)к одной базе данных IBM DB2, за счет чего достигается распараллеливание операций и достигается высокий уровень отказоустойчивости, а также существенное увеличение скорости обработки запросов, в особенности при выполнении OLTP транзакций.Реализация технологии IBM DB2 pureScale предусматривает наличие сервера-координатора, выполняющего функции централизованного управления блокировками, централизованного глобального кэша для страниц данных (т.н. буферный пул группы group buffer pool) и т.д.;

DB2 HADR – технология, позволяющая создать высокодоступную архитектуру СУБД DB2. Суть технологии заключается в наличии резервной базы данных, и зеркалировании всех транзакций, выполняемых на основной базе данных в резервную. При этом резервная база не активна, и не может использоваться для работы одновременно с основной (за исключением операций «только на чтение», которые допустимо применять к ре-

зервной базе данных). В HADR есть 4 типа синхронизации между основной и резервной базами: SYNC, NEARSYNC, ASYNC, SYPERASYNC. В режиме SYNC за счет синхронного непрерывного отображения содержимого транзакционных логов из основной базы в резервную, ее состояние может практически точно соответствует основной базе, и в случае выхода ее из строя и переключения на резервную базу данных потери транзакций не произойдет;

SQL Server Always On – технология обеспечения высокого уровня доступности и отказоустойчивости для СУБД SQL Server, реализованная в ПО MS SQL, начиная с версии SQL Server 2012. Функциональность AlwaysOn реализуется на базе встроенных средств кластеризации Windows Server Failover Cluster и включает два варианта реализации: Availability Groups и Failover Cluster Instances. Группы доступности (Availability Groups) подразумевают реплики базы данных, функционирующие на отдельных узлах кластера, при этом выделяются первичные и вторичные реплики. Первичные реплики – базы данных с которыми непосредственно происходит работа в режиме «чтение/запись», вторичные реплики являются резервными, и предназначены для транслирования в них всех транзакций из журнала первичной реплики. Процесс применения транзакций к вторичным репликам выстроен таким образом, что отказоустойчивость базы данных при выходе из строя первичной реплики, осуществляется без потери данных. Всего предусматривается возможность для создания до четырех реплик базы данных. Механизм контроля сбоя реализован без применения ресурса-свидетеля (witness), вместо которого используется определения сбоя методом «голосования», то есть определение сбоя осуществляется на основании контрольных данных состояния от большинства узлов кластера. Кроме резервирования первичной реплики, вторичные реплики выполняют роль доступа к данным в режиме «только на чтение», что позволяет наряду с обеспечением отказоустойчивости также достичь повышения производительности при выборке данных из базы.

В режиме FCI также предусматривается использование встроенного механизма кластеризации Windows Server. Экземпляры базы данных также функционируют на разных узлах кластера, однако в отличие от варианта с Availability Groups, в режиме FCI существует одна общая база (вместо отдельных реплик у каждого экземпляра). Активный экземпляр базы данных, функционирующий на одном из узлов кластера принимает владение базой данных, и в случае его выхода из строя, владение базой данных перехватывает экземпляр, функционирующий на другом узле кластера.

Восстановление базы данных – функция СУБД, которая в случае логических и физических сбоев приводит базу данных в актуальное и консистентное состояние.

В случае логического отказа или сигнала отката одной транзакции журнал изменений сканируется в обратном направлении, и все записи отменяемой транзакции извлекаются из журнала вплоть до отметки начала

транзакции. Согласно извлеченной информации выполняются действия, отменяющие действия транзакции. Этот процесс называется откат (rollback).

В случае физического отказа, если ни журнал изменений, ни сама база данных не повреждены, то выполняется процесс прогонки (rollforward). Журнал сканируется в прямом направлении, начиная от предыдущей контрольной точки. Все записи извлекаются из журнала вплоть до конца журнала. Извлеченная из журнала информация вносится в блоки данных внешней памяти, у которых отметка номера изменений меньше, чем записанная в журнале. Если в процессе прогонки снова возникает сбой, то сканирование журнала вновь начнётся сначала, но восстановление фактически продолжится с той точки, где оно прервалось.

В случае физического отказа, если журнал изменений доступен, но сама база данных повреждена, то должен быть выполнен процесс восстановления базы из резервной копии. После восстановления база будет находиться в состоянии на момент выполнения резервной копии. Для восстановления базы данных на момент отказа необходимо выполнить прогонку всех изменений, используя журнал изменений.

В случае физического отказа, если журнал изменений недоступен, но сама база данных не повреждена, восстановление возможно только на момент предыдущей контрольной точки.

В случае физического отказа, если повреждены как журнал изменений, так и сама база данных, то восстановление возможно только на момент выполнения резервной копии.

6.5. Методы, модели и средства восстановления данных

SQL Server выполняются в контексте модели восстановления базы данных. Модели восстановления предназначены для управления обслуживанием журналов транзакций. Модель восстановления — это свойство базы данных, которое управляет процессом регистрации транзакций, определяет, требуется ли для журнала транзакций резервное копирование, а также определяет, какие типы операций восстановления доступны. Существует три модели восстановления: простая модель восстановления, модель полного восстановления и модель восстановления с неполным протоколированием. Обычно в базе данных используется модель полного восстановления или простая модель восстановления. Базу данных можно в любой момент переключить на использование другой модели восстановления.

Рассмотрим известные виды резервного копирования.

Полное копирование обычно затрагивает всю систему и все файлы. Еженедельное, ежемесячное и ежеквартальное резервное копирование подразумевает создание полной копии всех данных. Обычно оно выполняется тогда, когда копирование большого объёма данных не влияет на работу организации. Для предотвращения большого объёма использованных

ресурсов используют алгоритмы сжатия, а также сочетание этого вида с другими: дифференциальным или инкрементным. Полное резервное копирование незаменимо в случае, когда нужно подготовить резервную копию для быстрого восстановления системы с нуля.

При *дифференциальном* («разностном») резервном копировании каждый файл, который был изменён с момента последнего полного резервного копирования, копируется каждый раз заново. Дифференциальное копирование ускоряет процесс восстановления. Все копии файлов делаются в определённые моменты времени, что, например, важно при заражении вирусами.

При добавочном («*инкрементном*») резервном копировании происходит копирование только тех файлов, которые были изменены с тех пор, как в последний раз выполнялось полное или добавочное резервное копирование. Последующее инкрементное резервное копирование добавляет только файлы, которые были изменены с момента предыдущего. Инкрементное резервное копирование занимает меньше времени, так как копируется меньшее количество файлов. Однако процесс восстановления данных занимает больше времени, так как должны быть восстановлены данные последнего полного резервного копирования, а также данные всех последующих инкрементных резервных копирований. В отличие от дифференциального копирования, изменившиеся или новые файлы не замещают старые, а добавляются на носитель независимо.

Клонирование позволяет скопировать целый раздел или носитель (устройство) со всеми файлами и каталогами в другой раздел или на другой носитель. Если раздел является загрузочным, то клонированный раздел тоже будет загрузочным.

Резервное копирование в виде образа предполагает создание точной копии всего раздела или носителя (устройства), хранящаяся в одном файле.

Резервное копирование в режиме реального времени позволяет создавать копии файлов, каталогов и томов, не прерывая работу, без перезагрузки компьютера.

При холодном резервировании база данных выключена или закрыта для потребителей. Файлы данных не изменяются, и копия базы данных находится в согласованном состоянии при последующем включении.

При горячем резервировании база данных включена и открыта для потребителей. Копия базы данных приводится в согласованное состояние путём автоматического приложения к ней журналов резервирования по окончании копирования файлов данных.

В рамках вашей стратегии резервного копирования вы, возможно, решите выполнять полное резервное копирование раз в неделю и дополнять его ежедневным, разностным или добавочным резервным копированием. Преимущество полных резервных копий состоит в том, что они содержат все выбранные вами файлы. Недостаток обычных резервных копий в том,

что они дольше создаются и занимают больше места, чем другие типы резервных копий. Добавочные и разностные копии, с другой стороны, занимают меньше места и создаются быстрее, так как они являются частичными. Их недостаток в том, что восстановление систем и файлов из добавочных и резервных копий выполняется медленнее, чем при использовании только обычной резервной копии. Чтобы понять, почему так происходит, рассмотрим следующие примеры резервного копирования и восстановления:

Обычное резервное копирование с ежедневным добавочным копированием. Полное резервное копирование выполняется каждое воскресенье, а добавочное – с понедельника по субботу. Добавочная резервная копия, созданная в понедельник, содержит изменения, выполненные с воскресенья. Добавочная копия, созданная во вторник, содержит изменения, выполненные с понедельника, и т д. Если сервер выйдет из строя в четверг и вам будет необходимо восстановить его из резервной копии, вы сможете сделать это, восстановив обычную резервную копию, созданную в воскресенье, добавочную копию, созданную в понедельник, добавочную копию, созданную во вторник, и добавочную копию, созданную в среду, - именно в таком порядке.

Обычное резервное копирование выполняется каждое воскресенье, а разностное – с понедельника по субботу. Разностная копия, созданная в понедельник, содержит изменения, выполненные с воскресенья, так же, как и резервная копия, созданная во вторник, в ереду, и т. д. Если сервер выйдет из строя в четверг и вам надо будет восстановить его из резервной копии, вы сможете сделать это, восстановив обычную резервную копию, созданную в воскресенье, а затем – разностную копию, созданную в среду.

6.6. Технология RAID

Одним из распространённых средств обеспечения отказоустойчивости сервера баз данных является RAID (англ. redundant array of independent/inexpensive disks) – избыточный массив независимых/недорогих жёстких дисков – матрица из нескольких дисков управляемых контроллером, взаимосвязанных скоростными каналами и воспринимаемых как единое целое. В зависимости от типа используемого массива может обеспечивать различные степени отказоустойчивости и быстродействия. Служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи информации. RAID используется в нескольких вариантах.

RAID 0 («Striping») представляет собой дисковый массив из 2 или более дисков, в котором информация разбита на блоки и последовательно записана на жесткие диски. Соответственно информация записывается и читается одновременно, что увеличивает скорость.

К сожалению, при отказе одного из дисков информация необратимо теряется, поэтому применяется либо в домашних условиях, либо для хранения файла подкачки, своп файла.

RAID 1 (Mirroring - «зеркалирование»). В данном случае один диск полностью повторяет другой, что гарантирует работоспособность при поломке одного диска, но объем полезного пространства уменьшается вдвое. Поскольку диски покупаются одновременно, в случае бракованной партии возможен отказ обоих дисков. Скорость записи приблизительно равна скорости записи на один диск, возможно чтение сразу с двух дисков (если контроллер поддерживает данную функцию), что увеличивает скорость.

Применяется чаще всего в малых офисах под базы данных, либо для хранения операционной системы.

RAID 5. В данном случае все данные разбиваются на блоки и для каждого набора считается контрольная сумма, которая хранится на одном из дисков – циклически записывается на все диски массива (попеременно на каждый), и используется для восстановления данных. Устойчив к потере не более чем одного диска.

RAID 5 имеет высокие показатели чтения – информация считывается почти со всех дисков, но уменьшенную производительность при записи – требуется вычислять контрольную сумму. Но самая критичная операция перезапись, так как она проходит в несколько этапов:

1) чтение данных;

2) чтение контрольной суммы;

3) сравнение новых и старых данных;

4) запись новых данных;

5) запись новой контрольной суммы.

RAID5 применяется при необходимости большого объема, и высокой скорости чтения.

RAID 10 (RAID 1+0). Сочетает в себе принципы RAID 0 и RAID 1. При его применении каждый жесткий диск имеет свою «зеркальную пару», при это используется половина полезного объема. Работоспособен пока существует один рабочий диск из каждой пары. Наиболее высокие показатели записи/перезаписи, сопоставимы с RAID 5 по скорости чтения. Применяется для хранения баз данных, при высокой нагрузке.

RAID 6 (ADG). Логическое продолжение RAID 5. Отличие заключается в том, что контрольная сумма высчитывается 2 раза, и, как следствие имеет большую надежность (устойчив при поломке более 2 дисков), и меньшую производительность.

Организация работы RAID обеспечивается RAID-контроллерами, которые могут быть: встроенными в материнскую плату, внутренними (в виде платы) и внешними.

Контрольные вопросы

1. Сформулируйте задачи администратора баз данных.
2. Что входит в обеспечение информационной безопасности на уровне БД?
3. В чём состоит управление развитием БД?
4. Какие инструменты имеет платформа Microsoft SQL Server?
5. Что понимается под зеркальным отображением БД?
6. Что такое отказоустойчивый кластер?
7. Какие преимущества даёт зеркальное отображение БД?
8. Что такое дифференциальное копирование?
9. Что такое инкрементное копирование?
10. В чём состоит клонирование?
11. Что такое RAID?
12. Что представляет собой RAID0?
13. Что представляет собой RAID1?
14. Что представляет собой RAID5?
15. Что представляет собой RAID10?

Задание

Спроектировать базу данных в заданной предметной области и реализовать серверную часть на платформе Microsoft SQL Server.

7. ВЕБ-СЛУЖБЫ И СЕРВИСЫ. АДМИНИСТРИРОВАНИЕ ИНТЕРНЕТ-УЗЛОВ

7.1. *Понятие веб-службы, URI, URL. Структура URL*

Веб-служба, веб-сервис (англ. web service) – это идентифицируемая веб-адресом программная система со стандартизированными интерфейсами. Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (SOAP, XML-RPC и т. д.) и соглашениях (REST). Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения.

В обиходе веб-сервисами называют услуги, оказываемые в Интернете. В этом употреблении термин требует уточнения, идёт ли речь о поиске, веб-почте, хранении документов, файлов, закладок и т. п. Такими веб-сервисами можно пользоваться независимо от компьютера, браузера или места доступа в Интернет. В основе сервисов лежат понятия URI и URL.

URI – это символьная строка, позволяющая идентифицировать какой-либо ресурс: документ, изображение, файл, службу, ящик электронной почты и т. д. Прежде всего, речь идёт, конечно, о ресурсах сети Интернет и Всемирной паутины. URI предоставляет простой и расширяемый способ идентификации ресурсов. Расширяемость URI означает, что уже существуют несколько схем идентификации внутри URI, и ещё больше будет создано в будущем.

Для доступа к любым сетевым ресурсам необходимо знать где они размещены и как к ним можно обратиться. Во Всемирной паутине для обращения к веб-документам изначально используется стандартизованная схема адресации и идентификации, учитывающую опыт адресации и идентификации таких сетевых сервисов, как e-mail, telnet, ftp и т.п. – URL, Uniform Resource Locator.

URL (RFC 1738) – это унифицированный локатор (указатель) ресурсов, стандартизированный способ записи адреса ресурса в www и сети Интернет. Адрес URL имеет гибкую и расширяемую структуру для максимально естественного указания местонахождения ресурсов в сети. Для записи адреса используется ограниченный набор символов ASCII. Общий вид адреса можно представить так:

<схема>://<логин>:<пароль>@<хост>:<порт>/<полный-путь-к-ресурсу>. Под схемой здесь подразумевается схема обращения к ресурсу, под которой часто понимается протокол: http, ftp, gopher, mailto, news, telnet, file, man, info, whatis, ldap, wais и т.п. Логин: пароль есть данные для авторизации пользователя. Под хостом подразумевается доменное имя хоста или его IP-адрес. Порт – это порт хоста для подключения. Полный путь к ресурсу зависит от протокола.

7.2. Службы Интернет Windows Server. Возможности, режимы работы

Службы Интернета – это системы, предоставляющие услуги пользователям Интернета. Такие службы IIS (информационные службы Интернета) поставляются компанией Microsoft с операционными системами, начиная от Windows NT, и могут быть установлены как на сервер, так и на клиента. К услугам относятся: электронная почта, WWW, телеконференции, списки рассылки, FTP, IRC, а также другие продукты, использующие Интернет как среду передачи информации.

Услуги, предоставляемые Интернетом, можно разделить на две основные категории.

1. Отложенные (off-line). Основным признаком этой группы является наличие временного перерыва между запросом и получением информации.

2. Прямые (on-line). Характерны тем, что информация по запросу возвращается немедленно. Если от получателя информации требуется немедленная реакция на нее, то такая услуга носит интерактивный характер.

Самой первой и самой распространенной службой Интернета является электронная почта (e-mail). Эта служба предоставляет услуги отложенного чтения. Пользователь посылает сообщение, и адресат получает его на свой компьютер через некоторый промежуток времени. Электронное письмо состоит из заголовков, содержащих служебную информацию (об авторе письма, получателе, пути прохождения по сети и т. д.), и содержимого письма.

Электронное письмо можно снабдить цифровой подписью и зашифровать. Скорость пересылки составляет в среднем несколько минут. При этом стоимость электронной почты минимальна и не зависит от расстояния. Основными достоинствами электронной почты являются простота, дешевизна и универсальность.

Телеконференции – вторая по распространенности служба Интернета, предоставляющая отложенные услуги.

Служба телеконференций состоит из множества тематических телеконференций – групп новостей (newsgroup), поддерживаемых серверами новостей. Сервер новостей – это компьютер, который может содержать тысячи групп новостей самых разнообразных тематик. Каждый сервер новостей, получивший новое сообщение, передает его всем узлам, с которыми он обменивается новостями. Группа новостей – это набор сообщений по определенной теме. Новости разделены по иерархически организованным тематическим группам, и имя каждой группы состоит из имен подуровней. Например, конференция comp.sys.linux.setup принадлежит группе «компьютеры», подгруппе «операционные системы», конкретнее – системе Linux, а именно – её установке.

Существуют как глобальные иерархии, так и иерархии, локальные для какой-либо организации, страны или сети. Набор групп, получаемых сер-

вером телеконференций, определяется его администратором и их наличием на других серверах, с которыми данный сервер обменивается новостями.

Доступ к группам новостей осуществляется через процедуру подписки, которая состоит в указании координат сервера новостей и выбора интересующих пользователя групп новостей. Следует заметить, что каждый сервер новостей имеет определенный набор конференций, и, если интересующая тематика на нем не найдена, можно попробовать использовать другой сервер. Данная процедура, а также работа с группами новостей осуществляется с помощью программного обеспечения, поддерживающего эти функции, например, широко распространенным приложением компании Microsoft Outlook Express.

В обсуждении темы телеконференции может участвовать множество людей, независимо от того, где они находятся физически. Обычно, хотя это и не является правилом, за порядком в конференциях следят специальные люди, так называемые модераторы. В их обязанности входит поддержание порядка в конференции в соответствие с установленными в ней правилами поведения и ее тематикой.

Наряду с описанной формой служб телеконференции широкое распространение получили WWW-телеконференции, также называемые форумами. Отличие состоит в том, что они работают через web-интерфейс, и размещаются не централизованно на серверах новостей, а наweb-сайтах.

Списки рассылки (mail lists) – служба, не имеющая собственного протокола и программы-клиента и работающая исключительно через электронную почту.

Идея работы списка рассылки состоит в объединении под одним адресом электронной почты адресов многих людей – подписчиков списка рассылки. Когда письмо посылается на этот адрес, сообщение получают все подписчики данного списка рассылки. Ведущими списка рассылки, как правило, являются люди, хорошо владеющие его тематикой. Они отвечают за подготовку и рассылку очередных выпусков. Получателями писем являются люди, собственноручно подписавшиеся на список. Кроме того, у них есть право и возможность в любой момент отменить свою подписку.

Существуют открытые рассылки (для всех желающих), закрытые (для людей определенного круга), бесплатные (существующие за счет энтузиазма создателей, спонсорской поддержки, платных рекламодателей) и платные.

В зависимости от числа подписчиков список рассылки обслуживается на сервере программами различной сложности. Эти программы могут обеспечивать или не обеспечивать полную функциональность, которая заключается в автоматической подписке клиентов и приеме их отказа от подписки, проверке корректности электронных адресов, ведении архива сообщений, обработке почтовых ошибок, поддержке работы в режиме дайджеста (когда подписчик получает не каждое сообщение отдельным

письмом, а все сообщения за какой-то срок в одном письме), проверке сообщений администратором списка перед рассылкой и т. д.

Под словом чат (от английского chat) подразумеваются службы Интернета, позволяющие проводить текстовые дискуссии в режиме реального времени. От традиционной формы разговора их отличает то, что они ведутся в текстовом виде — путем набора текста на клавиатуре. Самым популярным открытым стандартом, лежащим в основе чатов, является IRC (InternetRelayChat).

IRC – это многопользовательская, предназначенная для чата многоканальная сеть, с помощью которой пользователи могут беседовать в режиме реального времени независимо от своего месторасположения.

Несмотря на то, что IRC существует достаточно много лет, в коммерческой деятельности современных компаний, например, в работе центров обслуживания потребителей, этот стандарт практически не применяется. Основным его предназначением остается обсуждение самого широкого круга вопросов между пользователями Интернета.

В свое время чаты, в основе которых лежал стандарт IRC, получили достаточно широкое распространение. Однако сегодня все более популярными становятся чаты, проводимые на отдельных web-сайтах и в социальных сетях, основывающиеся либо на языке HTML, либо на языке Java. Это позволяет пользователям Интернета участвовать в них без установки дополнительного программного обеспечения, используя только стандартный браузер, тем самым число потенциальных участников становится максимальным. С другой стороны, возможность установки на корпоративном сайте компании системы, обеспечивающей работу чата, позволяет широко использовать эту службу в коммерческих целях, например, для обсуждения с потребителями тех или иных вопросов деятельности предприятия, обсуждения продукции, системы обслуживания и т. д.

Промежуточное положение между электронной почтой и чатами по динамичности и интерактивности общения занимают Интернет-пейджеры или службы мгновенных сообщений. Интернет-пейджеры постепенно становятся одними из самых популярных средств общения в Сети и по широте использования скоро смогут достичь электронную почту. Службы мгновенных сообщений позволяют общаться в режиме реального времени, совмещая в себе преимущества электронной почты и телефона. Частью процесса обмена в подобных системах могут становиться текстовый диалог, передача графики, голосовая и видео связь, обмен файлами. Примером подобных программ служат Viber, WhatsApp, мобильные приложения крупных социальных сетей и др.

FTP (file transfer protocol) – это протокол передачи файлов, но при рассмотрении FTP как службы Интернета имеется в виду не просто протокол, а именно служба доступа к файлам в файловых архивах. Одна из причин достаточно высокой ее популярности объясняется огромным количеством информации, накопленной в FTP-архивах за десятилетия эксплуата-

ции компьютерных систем. Другая причина кроется в простоте доступа, навигации и передачи файлов по FTP. Кроме того, FTP также служба прямого доступа, требующая полноценного подключения к Интернету.

WWW (World Wide Web) – служба прямого доступа, требующая полноценного подключения к Интернету и позволяющая интерактивно взаимодействовать с представленной на web-сайтах информацией. Это самая современная и удобная служба Интернета. Она основывается на принципе гипертекста и способна представлять информацию, используя все возможные мультимедийные ресурсы: видео, аудио, графику, текст и т. д. Взаимодействие осуществляется по принципу клиент-сервер с использованием протокола передачи гипертекста (Hyper Text Transfer Protocol, HTTP). С помощью протокола HTTP служба WWW позволяет обмениваться документами в формате языка разметки гипертекста – HTML (Hyper Text Markup Language), который обеспечивает надлежащее отображение содержимого документов в браузерах пользователей.

Принцип гипертекста, лежащий в основе WWW, состоит в том, что каждый элемент HTML-документа может являться ссылкой на другой документ или его часть, при этом документ может ссылаться как на документы на этом же сервере, так и на других серверах Интернета. Ссылки WWW могут указывать не только на документы, свойственные службе WWW, но и на прочие службы и информационные ресурсы Интернета. Более того, большинство программ-клиентов WWW – браузеров (browsers), обозревателей, или навигаторов, не просто понимают такие ссылки, но и являются программами-клиентами соответствующих служб: FTP, сетевых новостей Usenet, электронной почты и т. д. Таким образом, программные средства WWW являются универсальными для различных служб Интернета, а сама информационная система WWW выполняет по отношению к ним интегрирующую функцию.

Необходимо подчеркнуть, что Интернет и WWW – это не тождественные понятия. Узкое определение Интернета представляет его как взаимосвязь компьютерных сетей на базе семейства протоколов TCP/IP, в пространстве которой становится возможным функционирование протоколов более высокого уровня, в том числе протокола передачи гипертекста (HTTP) – протокола WorldWideWeb, гипертекстового сервиса доступа к удаленной информации. Кроме WorldWideWeb, на этом уровне (он называется прикладным или уровнем приложений) действуют и другие протоколы, например, электронной почты (POP3, SMTP, IMAP), общения в режиме реального времени (IRC) и групп новостей (NNTP).

Таким образом, WorldWideWeb – это одна из служб Интернета, которая предлагает простой в использовании интерфейс и дает возможность пользователям, даже не слишком хорошо знающим компьютер, получать доступ к web-службам в любой части Интернета.

В отдельную группу можно выделить службы Интернета, не имеющие сегодня такого широкого распространения, как те, о которых было расска-

зано ранее и не имеющие всеми признанных единых стандартов. В их основе также лежит использование Интернета как среды передачи информации. В частности, к этой группе можно отнести:

- средства передачи голоса по каналам связи Интернета, предоставляющие услуги телефонной и факсимильной связи;
- программные средства для проведения видео- и аудио- конференций через Интернет;
- системы широковещательной передачи мультимедийной информации.

Особую группу составляют службы Интернета, поддерживаемые одной из групп его участников и причисляемые в данной категории благодаря глобальному характеру предоставляемых ими услуг по поиску информации. Поиск информации является сегодня одной из ключевых проблем Интернета, так как количество представленных в нем web-страниц сегодня оценивается более чем в несколько сотен миллионов. Кроме того, в основе проблем поиска информации лежат такие причины, как множественность и фрагментарность источников, большое количество различных способов хранения данных, дефицит времени на выборку и обработку информации, стоимость получения информации, ненадежность данных, постоянное обновление и добавление информации.

Ниже перечислены основные инструменты поиска информации в Интернете, которым удается в значительной степени преодолевать вышеназванные трудности:

Поисковые машины (spiders, crawlers). Основная функция поисковых машин состоит в исследовании Интернета с целью сбора данных о существующих в нем web-сайтах и выдаче по запросу пользователя информации oweb-страницах, наиболее полно удовлетворяющих введенному запросу.

Каталоги представляют собой иерархически организованную тематическую структуру, в которую, в отличие от поисковых машин, информация заносится по инициативе пользователей. Добавляемая страница жестко привязывается к принятым в каталоге категориям.

Мета-средства поиска позволяют усовершенствовать процесс путем запуска одновременно нескольких поисковых средств. Этот способ значительно повышает скорость, однако не позволяет воспользоваться возможностями построения сложных запросов, предлагаемыми большинством современных систем поиска.

7.3. Обеспечение безопасности в веб-службах

Одним из важнейших условий широкого применения Интернета было и остается обеспечение адекватного уровня безопасности для всех транзакций, проводимых через него. Это касается информации, передаваемой между пользователями, информации сохраняемой в базах данных торговых систем, информации, сопровождающей финансовые транзакции.

Понятие безопасности информации можно определить, как состояние устойчивости информации к случайным или преднамеренным воздействиям, исключающее недопустимые риски ее уничтожения, искажения и раскрытия, которые приводят к материальному ущербу владельца или пользователя информации. Поскольку Сеть полностью открыта для внешнего доступа, то роль этих методов очень велика. Большая значимость фактора безопасности также отмечается многочисленными исследованиями, проводимыми в Интернете.

Решить проблемы безопасности призвана криптография – наука об обеспечении безопасности данных. Криптография и построенные на ее основе системы призваны решать следующие задачи:

1. Конфиденциальность. Информация должна быть защищена от несанкционированного доступа как при хранении, так и при передаче. Доступ к информации может получить только тот, для кого она предназначена. Обеспечивается шифрованием.

2. Аутентификация. Необходимо однозначно идентифицировать отправителя, при однозначной идентификации отправитель не может отказаться от послания. Обеспечивается электронной цифровой подписью и сертификатом.

3. Целостность. Информация должна быть защищена от несанкционированного изменения как при хранении, так и при передаче. Обеспечивается электронной цифровой подписью.

В соответствии с названными задачами основными методами обеспечения безопасности выступают шифрование, цифровая подпись и сертификаты.

Осуществляя сделки в Сети, в первую очередь необходимо убедиться, что важная информация надежно скрыта от посторонних лиц. Этому служат технологии шифрования, преобразующие простой текст в форму, которую невозможно прочитать, не обладая специальным шифровальным ключом. Благодаря данным технологиям можно организовать безопасную связь по общедоступным незащищенным каналам Интернета.

Любая система шифрования работает по определенной методологии, включая в себя один или более алгоритмов шифрования (математических формул), ключи, используемые этими алгоритмами, а также систему управления ключами.

Согласно методологии шифрования, сначала к тексту применяются алгоритм шифрования и ключ для получения из него шифрованного текста. Затем шифрованный текст передается к месту назначения, где тот же самый алгоритм и ключ используются для его расшифровки, чтобы получить первоначальный текст. В методологию шифрования также входят процедуры создания ключей и их распространения.

Наиболее распространены алгоритмы шифрования, которые объединяют ключ с текстом. Безопасность систем такого типа зависит от конфиденциальности ключа, используемого в алгоритме шифрования, а не от

конфиденциальности самого алгоритма, который может быть общедоступен и благодаря этому хорошо проверен. Но основная проблема, связанная с этими методами, состоит в безопасной процедуре генерации и передачи ключей участникам взаимодействия.

В настоящее время существует два основных типа криптографических алгоритмов:

1. Классические, или симметричные алгоритмы, основанные на использовании закрытых, секретных ключей, когда и шифрование, и дешифрирование производятся с помощью одного и того же ключа.

2. Алгоритмы с открытым ключом, в которых используются один открытый и один закрытый ключ, то есть операции шифрования производятся с помощью разных ключей. Эти алгоритмы называются также асимметричными.

Каждая методология требует собственных способов распределения ключей и собственных типов ключей, а также алгоритмов шифрования и расшифровки ключей.

Технология шифрования с *секретным ключом* (*симметричный алгоритм*) требует, чтобы оба участника зашифрованной переписки имели доступ к одному и тому же ключу. Это необходимо, так как отправитель использует ключ для зашифровки сообщения, а получатель применяет его же для расшифровки. Как следствие, возникает проблема безопасной передачи этого ключа.

Алгоритмы симметричного шифрования используют ключи не очень большой длины и могут быстро шифровать большие объемы данных.

Порядок использования систем с симметричными ключами выглядит следующим образом:

1. Безопасно создается, распространяется и сохраняется симметричный секретный ключ.

2. Отправитель использует симметричный алгоритм шифрования вместе с секретным симметричным ключом для получения зашифрованного текста.

3. Отправитель передает зашифрованный текст. Симметричный секретный ключ никогда не передается по незащищенным каналам связи.

4. Для восстановления исходного текста, получатель применяет к зашифрованному тексту тот же самый симметричный алгоритм шифрования вместе с тем же самым симметричным ключом, который уже есть у него.

Для решения проблемы распространения ключей при использовании симметричных методов шифрования на основе результатов, полученных классической и современной алгеброй, были предложены системы с *открытым ключом*, или *асимметричные криптосистемы*. Суть их состоит в том, что каждым адресатом генерируются два ключа, связанные между собой по определенному правилу. Хотя каждый из пары ключей подходит как для шифрования, так и для дешифрирования, данные, зашифрованные одним ключом, могут быть расшифрованы только другим.

Один ключ объявляется открытым, а другой закрытым. Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Секретный ключ сохраняется в тайне. Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст не может быть расшифрован тем же открытым ключом. Дешифрирование сообщения возможно только с использованием закрытого ключа, известного лишь самому адресату.

Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции.

Понятие односторонней функции было введено в теоретическом исследовании о защите входа в вычислительные системы. Функция *f(x)* называется односторонней (one-way function), если для всех значений x из ее области определения легко вычислить значения *y=f(x)*, но вычисление обратного значения практически неосуществимо. То есть по заданному значению *y0* нельзя найти такое значение *x0*, для которого *f(x0)=y0*. «Практически неосуществимо» в данном случае означает, что требуется такой огромный объем вычислений, который при существующем уровне развития техники невозможно реализовать.

Множество классов необратимых функций порождает все разнообразие систем с открытым ключом.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Известно несколько криптосистем с открытым ключом. Наиболее развита на сегодня система RSA, предложенная еще в 1978 г. Алгоритм RSA назван по первым буквам фамилий его авторов: Р. Л. Райвеста (R. L. Rivest), А. Шамира (A. Shamir) и Л. Адлемана (L. Adieman). Этот алгоритм стал мировым фактически признанным стандартом для открытых систем и рекомендован МККТТ (Международный Консультативный Комитет по телефонии и телеграфии). Также используются алгоритмы: ECC (криптосистема на основе эллиптических кривых), алгоритм Эль-Гамаля.

Следует отметить, что алгоритмы систем шифрования с открытым ключом можно использовать в качестве следующих инструментов: · как самостоятельные средства защиты передаваемых и хранимых данных; · как средства для распределения ключей (алгоритмы систем шифрования с открытым ключом более трудоемки, чем традиционные криптосистемы, поэтому на практике часто бывает рационально передать ключи, объем информации в которых незначителен с их помощью, а потом с помощью обычных алгоритмов осуществлять обмен большими информационными потоками); как средства аутентификации пользователей (для создания электронной цифровой подписи).

Все асимметричные криптосистемы являются объектом атак, в которых применяется прямой перебор ключей, поэтому для обеспечения эквивалентного уровня защиты в них должны использоваться гораздо более

длинные ключи, чем в симметричных криптосистемах. Можно привести следующие приблизительные данные об эквивалентности длин ключей.

Для того чтобы избежать низкой скорости алгоритмов асимметричного шифрования, методы шифрования с открытым ключом часто используют для шифрования небольших объемов информации, например, для шифрования секретного ключа, на основе которого далее производится криптографическое закрытие информации симметричными методами.

Шифрование передаваемых через Интернет данных позволяет защитить их от посторонних лиц. Однако для полной безопасности должна быть уверенность в том, что второй участник транзакции является тем лицом, за которое он себя выдает. В бизнесе наиболее важным идентификатором личности заказчика является его подпись. В электронной коммерции применяется электронный эквивалент традиционной подписи – *цифровая подпись*. С ее помощью можно доказать не только то, что транзакция была инициирована определенным источником, но и то, что информация не была испорчена во время передачи.

Как и в шифровании, технология электронной подписи использует либо секретный ключ (в этом случае оба участника сделки применяют один и тот же ключ), либо открытый ключ (при этом требуется пара ключей – открытый и личный). И в данном случае более просты в использовании и более популярны методы с открытым ключом (такие, как RSA)

Хэш-функции являются одним из важных элементов криптосистем на основе ключей и используются для обнаружения факта модификации сообщения, то есть для электронной подписи. Их относительно легко вычислить, но почти невозможно расшифровать. Хэш-функция имеет исходные данные переменной длины и возвращает строку (иногда называемую дайджестом сообщения – MD) фиксированного размера, обычно 128 бит.

Существует несколько защищенных хэш-функций: Message Digest 5 (MD-5), Secure Hash Algorithm (SHA) и др. Они гарантируют, что разные документы будут иметь разные электронные подписи, и что даже самые незначительные изменения документа вызовут изменение его дайджеста.

Рассмотрим, как работает технология цифровой подписи, использующая алгоритм RSA. Предположим, вы хотите послать сообщение. В этом случае порядок работы следующий:

1. При помощи хеш-функции вы получаете дайджест – уникальным образом сжатый вариант исходного текста.

2. Получив дайджест сообщения, вы шифруете его с помощью личного ключа RSA, и дайджест превращается в цифровую подпись.

3. Вы посылаете вместе с самим сообщением цифровую подпись.

4. Получив послание, получатель расшифровывает цифровую подпись с помощью вашего открытого ключа и извлекает дайджест сообщения.

5. Получатель, применяя для сообщения ту же хэш-функцию, что и вы, получает свой сжатый вариант текста и сравнивает его с дайджестом,

восстановленным из подписи. Если они совпадают, то это значит, что подпись правильная и сообщение действительно поступило от вас. В противном случае сообщение либо отправлено из другого источника, либо было изменено после создания подписи.

При аутентификации личности отправителя открытый и личный ключи играют роли, противоположные тем, что они выполняли при шифровании. Так, в технологии шифрования открытый ключ используется для зашифровки, а личный – для расшифровки. При аутентификации с помощью подписи все наоборот. Кроме того, подпись гарантирует только целостность и подлинность сообщения, но не его защиту от посторонних глаз. Для этого предназначены алгоритмы шифрования. Например, стандартная технология проверки подлинности электронных документов DSS (Digital Signature Standard) применяется в США компаниями, работающими с государственными учреждениями. Однако у технологии RSA более широкие возможности в силу того, что она служит как для генерации подписи, так и для шифрования самого сообщения. Цифровая подпись позволяет проверить подлинность личности отправителя: она основана на использовании личного ключа автора сообщения и обеспечивает самый высокий уровень сохранности информации.

Как было сказано выше, основной проблемой криптографических систем является распространение ключей. В случае симметричных методов шифрования эта проблема стоит наиболее остро, поэтому при шифровании данных для передачи ключей через Интернет чаще всего используются асимметричные методы шифрования.

Асимметричные методы более приспособлены для открытой архитектуры Интернета, однако и здесь использование открытых ключей требует их дополнительной защиты и идентификации для определения связи с секретным ключом. Без такой дополнительной защиты злоумышленник может выдать себя за отправителя подписанных данных или за получателя зашифрованных данных, заменив значение открытого ключа или нарушив его идентификацию. В этом случае каждый может выдать себя за другое лицо. Все это приводит к необходимости верификации открытого ключа. Для этих целей используются *электронные сертификаты*.

Электронный сертификат представляет собой цифровой документ, который связывает открытый ключ с определенным пользователем или приложением [13]. Для заверения электронного сертификата используется электронная цифровая подпись доверенного центра – ЦС (Центра Сертификации). Исходя из функций, которые выполняет ЦС, он является основным компонентом всей инфраструктуры открытых ключей (ИОК или PKI – Public Key Infrastructure). Используя открытый ключ ЦС, каждый пользователь может проверить достоверность электронного сертификата, выпущенного ЦС, и воспользоваться его содержимым.

Для того чтобы сертификатам можно было доверять, независимая организация, выполняющая функции ЦС и являющаяся их источником,

должна быть достаточно авторитетной. В настоящее время наиболее известным источником сертификатов являются компании Thawte (www.thawte.com) и VeriSign (www.verisign.com), однако существуют и другие системы, такие как World Registry (IBM), Cyber Trust (GTE) и Entrust (Nortel). В России дистрибьютором сертификатов SSL компании Thawte сегодня является «РосБизнесКонсалтинг» (www.rbc.ru).

Технология цифровых сертификатов работает следующим образом. Чтобы воспользоваться сертификатом, потенциальный покупатель должен, прежде всего, получить его в надежном источнике. Для этого ему необходимо каким-то образом доказать подлинность своей личности, возможно, явившись в эту организацию и предъявив соответствующий документ, а также передать источнику сертификатов копию своего открытого ключа. После этого при желании купить что-либо через Интернет, ему будет достаточно добавить к заказу свою электронную подпись и копию сертификата. Отдел обслуживания покупателей фирмы, в которой он совершил покупку, проверяет сертификат, чтобы убедиться, что к заказу приложен подлинный открытый ключ, а также выясняет, не аннулирован ли сертификат.

Следует отметить, что технология цифровых сертификатов является двунаправленной. Это значит, что не только фирма может проверить подлинность заказа покупателя, но и сам покупатель имеет возможность убедиться, что он имеет дело именно с той фирмой, за которую она себя выдает. Осуществив взаимную проверку, обе стороны спокойно заключают сделку, так как обладают подлинными открытыми ключами друг друга и, соответственно, могут шифровать передаваемые данные и снабжать их цифровой подписью. Такой механизм обеспечивает надежность сделки, ибо в этом случае ни одна из сторон не сможет отказаться от своих обязательств.

Протоколы и стандарты безопасности

Описанные выше методы обеспечения безопасности являются основой построения большинства Интернет-систем. Это могут быть системы обмена информацией или платежные системы. Важность вопросов безопасности для их организации очень велика. Так, согласно проводимым исследованиям, одной из основных причин медленного роста электронной коммерции сегодня остается озабоченность покупателей надежностью средств, применяемых при расчетах в Интернете. Основные причины обеспокоенности связаны со следующими факторами.

1. Отсутствие гарантии конфиденциальности – кто-либо может перехватить передаваемые данные и попытаться извлечь ценную информацию, например, данные о кредитных картах. Это может произойти как во время передачи информации, так и непосредственно после совершения покупки через торговые web-сайты.

2. Недостаточный уровень проверки (аутентификации) участников операции – покупатель, посещая электронный магазин, не уверен, что

представленная на нем компания именно та, за кого она себя выдает, а у продавца нет возможности проверить, что покупатель, сделавший заказ, является законным обладателем кредитной карты.

3. Нет гарантии целостности данных – даже если отправитель данных может быть идентифицирован, то третья сторона может изменить их во время передачи.

Наиболее распространенными механизмами, призванными устранить указанные факторы и обеспечить безопасность проведения электронных платежей через Интернет сегодня являются:

1. Протокол SSL (Secure Socket Layer), обеспечивающий шифрование передаваемых через Интернет данных;

2. Стандарт SET (Secure Electronic Transactions), разработанный компаниями Visa и MasterCard и обеспечивающий безопасность и конфиденциальность совершения сделок при помощи пластиковых карт.

Протокол SSL – один из существующих протоколов обмена данными, обеспечивающий шифрование передаваемой информации. В настоящее время это наиболее распространенный метод защиты электронных транзакций в Интернете.

Протокол SSL является стандартом, основанным на криптографии с открытыми ключами. Протокол обеспечивает защиту данных, передаваемых в сетях TCP/IP по протоколам приложений за счет шифрования и аутентификации серверов и клиентов. Это означает, что шифруется вся информация, передаваемая и получаемая web-браузером, включая URL-адреса, все отправляемые сведения (такие, как номера кредитных карт), данные для доступа к закрытымweb-сайтам (имя пользователя и пароль), а также все сведения, поступающие cweb-серверов.

Протокол SSL позволяет решить часть названных проблем безопасности, однако его роль в основном ограничивается обеспечением шифрования передаваемых данных. Поэтому для комплексного решения перечисленных выше проблем была разработана спецификация и создан набор протоколов, известные как стандарт SET (Secure Electronic Transaction) – безопасные электронные транзакции.

Официальной датой введения стандарта SET является 1 февраля 1996 г. В этот день Visa International и MasterCard International совместно с рядом технологических компаний объявили о разработке единого открытого стандарта защищенных расчетов через Интернет с использованием пластиковых карт [14]. В декабре 1997 г. была создана некоммерческая организация SETCo LLC, призванная координировать работы по развитию стандарта и осуществлять тестирование и сертификацию предлагаемого на рынке программного обеспечения для обеспечения контроля над соответствием этого программного обеспечения спецификациям SET.

Благодаря использованию цифровых сертификатов и технологий шифрования, SET позволяет как продавцам, так и покупателям производить аутентификацию всех участников сделки. Кроме того, SET обеспечи-

вает надежную защиту номеров кредитных карт и другой конфиденциальной информации, пересылаемой через Интернет, а открытость стандарта позволяет разработчикам создавать решения, которые могут взаимодействовать между собой. Также важным фактором, обеспечивающим продвижение SET, является его опора на существующие карточные системы, ставшие привычным финансовым инструментом с отлаженной технологией и правовым механизмом.

В основе системы безопасности, используемой SET, лежат стандартные криптографические алгоритмы DES и RSA. Инфраструктура SET построена в соответствии с инфраструктурой открытого ключа (Public Key Infrastructure, PKI) на базе сертификатов, соответствующих стандарту X.509, утвержденному организацией по стандартизации (ISO).

Главная особенность SET – регламентация использования системы безопасности, которая устанавливается международными платежными системами [15]. Требования Visa и Europay к центру обработки на основе SET включают, во-первых, традиционные требования к обработке пластиковых карт (защита помещений, контроль над доступом, резервное энергоснабжение, аппаратная криптография и т. п.), и, во-вторых, специфические дополнения – межсетевые экраны (firewalls) для защиты каналов Интернета. Такой подход позволяет использовать единые методики оценки рисков при проведении электронных платежей вне зависимости от способа аутентификации клиента (традиционная карта с магнитной полосой, смарт-карта или цифровой сертификат). Это позволяет участникам платежной системы разрешать спорные ситуации по отработанным механизмам и сконцентрироваться на развитии своего электронного бизнеса.

SET обеспечивает следующие требования защиты операций электронной коммерции:

- секретность данных оплаты и конфиденциальность информации заказа, переданной вместе с данными об оплате;
- сохранение целостности данных платежей, которая обеспечивается при помощи цифровой подписи;
- специальную криптографию с открытым ключом для проведения аутентификации;
- аутентификацию держателя кредитной карты, которая обеспечивается применением цифровой подписи и сертификатов держателя карты; аутентификацию продавца и его возможности принимать платежи по пластиковым картам с применением цифровой подписи и сертификатов продавца;
- подтверждение того, что банк продавца является действующей организацией, которая может принимать платежи по пластиковым картам через связь с обрабатывающей системой, что обеспечивается с помощью цифровой подписи и сертификатов банка продавца;

• готовность оплаты транзакций в результате аутентификации сертификата с открытым ключом для всех сторон;

• безопасность передачи данных посредством использования криптографии.

SET позволяет сохранить существующие отношения между банком, держателями карт и продавцами, и объединяется с действующими системами, опираясь на открытость, международные стандарты платежных систем, лежащие в его основе, а также технологии и правовые механизмы, существующие в финансовой отрасли.

Как известно, после геополитических событий 2014 года, когда иностранные платёжные системы поставили под угрозу использование денежных средств на банковских картах граждан России, российское правительство было вынуждено запустить работы по созданию национальной платёжной системы «МИР», независимой от иностранных компаний. На сегодняшний день платёжная система введена в действие и успешно функционирует.

После появления карты МИР за безопасность отвечала система MirAccept, которую разработали наши специалисты, но она базировалась на основе стандарта 3D Secure 1.0, созданного Visa [16]. Данная система безопасности платежа в интернете основывается на том, что держателю карты, который собирается оплатить товар или услугу онлайн, высылается на мобильный телефон одноразовый пароль для подтверждения транзакции.

Пока все кары МИР работали на основе системы безопасности MirAccept 1.0 версии, разработчики национальной платежной системы сообщили, что ими разработана новая платформа для безопасности платежей в интернете версии 2.0. Новая система также базировалась на протоколе 3D Secure только версии 2.0, а она принадлежит не Visa, а EMVCo являющейся международной организацией, определяющей стандарты в области электронной коммерции.

MirAccept версии 2.0 даст возможность владельцам карт МИР совершенно безопасно совершать платежи как на компьютерах, так и в мобильных телефонах, планшетах и так далее.

Также осталась идентификация плательщика по одноразовому паролю, высылаемого в СМС-сообщении, и добавилась функция более глубокого анализа таких параметров как частота транзакций держателя карты МИР, анализ устройств, с которых производится оплата и тому подобное. Другими словами, новая система учитывает множество рисков и дополнительно защищает клиента от кражи денег с карты.

Все банки, выпускающие карты на основе национальной платёжной системы МИР, уже переходят на новую систему безопасности MirAccept 2.0 и для тех, кто недавно оформил карты, первый год обслуживания будет бесплатным.

7.4. Администрирование веб-служб и веб-узлов, построение веб-представительства компании

Веб-узел – это информационный ресурс, дающий возможность предоставлять доступ к информации, организовать работу пользователей с информационной системой. Веб-узел обеспечивает хранение и упорядочивание информационных ресурсов (документов, графических объектов, ссылок, файлов мультимедиа).

Администрирование веб-узла предполагает решение следующих задач:

1. Управление содержимым веб-узла.
2. Обеспечение доступности и целостности хранимой информации.
3. Разграничение доступа к различным областям и объектам веб-узла.
4. Выполнение резервного копирования и восстановления данных.
5. Мониторинг производительности веб-приложения.
6. Аудит работы пользователей.

Построение веб-узла включает ряд подзадач:

- разработка информационного массива, хранящего данные;
- организация доступа к данным информационного ресурса;
- разработка наборов программных модулей для отбора и обработки информации;
- разработка интерфейсных форм для организации работы с системой.

Доступ к информационным ресурсам информационной системы организуется с использованием архитектуры двухуровневого клиент-сервера.

Серверное решение включает в себя компьютер с установленным программным обеспечением (платформой Microsoft):

- серверная операционная система, например, MS Windows Server;
- веб-сервер и необходимые серверные расширения, такие как ASP, службы Windows SharePoint Services – решение для совместной работы на основе веб-технологий;
- система управления базами данных, например, MS SQL Server;
- другие информационные ресурсы – гипертекстовые страницы, графические файлы, другие документы.

Клиентами выступают персональные компьютеры с установленным программным обеспечением – веб-клиентами (браузерами). Пользователи системы – это лица, участвующие в процессе движения информации, порождающие информационные потоки.

В качестве пользователей выступают такие категории лиц:

- администраторы системы, осуществляющий общее администрирование и поддержку информационной системы;
- руководители организации и подразделений, осуществляющие контроль за работой сотрудников, просмотр и корректировка информации, относящейся к его деятельности;

• сотрудники, выполняющие ввод оперативной и иной информации, просмотр результатов обработки данных в соответствии с правилами безопасности и должностными обязанностями;

• анонимные пользователи системы, просматривающие общедоступную информацию.

Для хранения и обработки информации используются различные средства управления, например, СУБД MS SQL Server, программные модули обработки информации и ее представления в удобном для пользователя виде (средства генерации отчетов). Информация может быть представлена как статическими файлами, так и динамическими структурами, генерируемыми в процессе работы информационной системы.

Интерфейсные формы информационной системы должны обеспечить возможности просмотра и изменения данных, хранящихся в базах данных информационной системы, различными группами пользователей. Для отображения информации в информационных системах, основанных на web-технологии, наиболее часто используются документы в формате html.

Использование языка HTML позволяет вывести в окне браузера разнообразную информацию, включающую форматированный текст, графическую информацию, мультимедиа-информацию и т.п.

Представление данных в информационной системе обеспечивается посредством набора интерфейсных формы. Возможным подходом к организации интерфейса пользователя является реализация в виде веб-приложений на основе различных технологий типа ASP.NET – Active Server Pages. Данный подход позволяет обеспечить гибкое управление содержимым портала и представлением данных. Технология ASP является ключевой технологией разработки динамических web-страниц в рамках решений Microsoft.

Технология ASP обладает гибкостью в использовании, позволяет создавать на веб-сервере динамические интерактивные страницы. Страницы могут формироваться с учетом того типа браузера, который установлен на машине пользователя. Сценарий ASP выполняется как внутренний процесс сервера. Обработка сценариев является многопотоковым процессом.

Процесс создания динамических страниц с помощью сценариев ASP может быть разделен на несколько этапов:

• браузер запрашивает ASP-страницу с web-сервера;

• ASP-файл загружается в машину обработки сценариев;

• выполняются команды файла сценария на стороне web-сервера;

• загружаются ADO-объекты, которые образуют интерфейс между web-страницами и различными типами источников данных (отдельные файлы, базы данных SQL-сервера);

• вызов баз данных и подстановка информации в web-страницы;

• отправка сгенерированных страниц в браузер клиента.

Системный администратор не всегда управляет веб-представительством на всех уровнях, включая физический. Платформа для корпоративного веб-сайта может предоставляться и сторонней организацией, и в этом случае задачи администратора ограничиваются логическим построением веб-узла и управлением им через консоль, предоставляемую разработчиком платформы, однако в этом случае следует учитывать, что контроль веб-узла не будет полностью находиться в руках отвечающего за него администратора, соответственно, и работоспособность такого веб-представительства будет во многом зависеть от компании, которая предоставляет программно-аппаратную платформу. Распространённым способом построения веб-представительства является применение систем управления контентом, о которых пойдёт речь в последнем параграфе.

7.5. Системы управления контентом

Любой Web-сайт состоит из набора страниц, а различия заключаются лишь в том, как они организованы. Существует два вида организации Web-сайта – статический и динамический. В первом случае специалисты, отвечающие за создание и поддержку сайта, пишут в HTML-форме каждую в отдельности страницу, включая ее оформление и контент. Во втором – в основе любой Web-страницы лежит шаблон, определяющий расположение в окне Web-браузера всех компонентов страницы, и вставка конкретной информации производится с использованием стандартных средств, не требующих от участника процесса знания языка HTML и достаточно сложных для неспециалиста процедур публикации Web-страницы.

Если сайт состоит из множества страниц или он должен часто обновляться, то преимущество динамической организации становится очевидным. Разработчикам Web-сайта не надо переписывать всю страницу при изменении ее информационного наполнения или дизайна. Страницы не хранятся целиком, а формируются динамически при обращении к ним.

Таким образом, отделение дизайна от контента является главной отличительной особенностью динамических сайтов от статических. На этой основе возможны дальнейшие усовершенствования структуры сайта, такие как определение различных пользовательских функций и автоматизация бизнес-процессов, а самое главное, контроль поступающего на сайт контента.

Для создания динамического сайта возможны два пути. Во-первых, это написание собственных программ, отвечающих за создание нужных шаблонов и поддерживающих необходимые функции. При этом созданная система будет полностью отвечать потребностям, однако возможно потребует больших программистских усилий и времени. Второй путь – это воспользоваться уже существующими системами, которые и называются системами управления Web-контентом. Преимуществом этого пути является уменьшение затрат времени и сил. К его недостаткам можно отнести сни-

жение гибкости, предоставление недостаточного или чрезмерного набора возможностей.

Под контентом (дословный перевод английского термина content, означающего содержание, содержимое) понимают информационное наполнение сайта – то есть все типы материалов, которые находятся на сервере: web-страницы, документы, программы, аудиофайлы, фильмы и так далее. Таким образом, управление контентом – это процесс управления подобными материалами. Он включает следующие элементы: размещение материалов на сервере, удаление материалов с сервера, когда в них больше нет необходимости, организацию (реорганизацию) материалов, возможность отслеживать их состояние.

Системы управления контентом (в английском языке существует устоявшийся термин – Content Management Systems или, сокращенно, CMS) – это программные комплексы, автоматизирующие процедуру управления контентом.

Функции систем управления контентом можно разделить на несколько основных категорий:

1. Создание – предоставление авторам удобных и привычных средств создания контента.

2. Управление – хранение контента в едином репозитории. Это позволяет следить за версиями документов, контролировать, кто и когда их изменял, убеждаться, что каждый пользователь может изменить только тот раздел, за который он отвечает. Кроме того, обеспечивается интеграция с существующими информационными источниками и ИТ-системами. CMS поддерживает контроль над рабочим потоком документов, т.е. контроль за процессом их одобрения. Таким образом, управление контентом включает в себя хранение, отслеживание версий, контроль за доступом, интеграцию с другими информационными системами и управление потоком документов.

3. Публикация – автоматическое размещение контента на терминале пользователя. Соответствующие инструменты автоматически адаптируют внешний вид страницы к дизайну всего сайта.

4. Представление – дополнительные функции, позволяющие улучшить форму представления данных; например, можно строить навигацию по структуре репозитория.

Системы управления контентом делятся на четыре основных категории, которые частично перекрываются:

1. Системы управления исходными кодами традиционно поддерживают управление исходными кодами программ, и часто предоставляют некоторый web-интерфейс, который может использоваться внутри корпоративной сети, а также вне ее для параллельной работы с исходными кодами.

2. Системы управления документами предназначены для организаций, оперирующих с большим количеством документов, например, офисы больших компаний, редакции и страховые компании.

3. Системы управления web-контентом представляют собой новую индустрию программных продуктов. Эти системы предназначены для разработки и управления Web-сайтами различной степени сложности. Обычно такие системы поддерживают и некоторый тип управления потоками работ.

4. Системы электронной коммерции – обеспечивают хранение и управление электронными каталогами товаров. По сути, эти системы незначительно отличаются друг от друга. Самое главное отличие этих систем – это люди, которые их используют.

Использование CMS предоставляет следующие преимущества:

1. Оперативное обновление информации - информацию публикует сотрудник, владеющий информацией, без дополнительных посредников в виде технических специалистов. CMS предназначены для автоматизации процесс публикации информации на web-сайте, предоставляя пользователям возможность самим публиковать материалы в WWW и определять их визуальное представление, используя для этого стандартные средства, не требующие знания языка HTML и достаточно сложных для неспециалиста процедур. С помощью CMS можно, не будучи профессиональным разработчиком, создавать и модифицировать информационное наполнение сайтов.

2. Снижение стоимости поддержки – обновление информации производится самостоятельно, нет необходимости оплачивать труд собственного или внешнего web-мастера. Снижение стоимости происходит за счет снижения потерь времени на поиски документов, пресечения дублирования и ошибок, увеличения скорости связи с партнерами и клиентами.

3. Предоставление дополнительных сервисов пользователю – часть сервисов – поиск, форумы, голосования и т.д., требуют интерактивного взаимодействия с пользователем. Они уже реализованы в рамках CMS.

4. Уменьшение сроков и стоимости разработки – наиболее востребованная функциональность уже реализована в CMS и может быть сразу использована.

5. Повышение качества разработки – при разработке полностью или частично используются готовые модули, которые уже прошли неоднократное тестирование.

6. Снижение стоимости дальнейших модификаций – CMS позволяют разделить данные и их представление. Это позволяет гораздо проще изменить внешний вид сайта, чем в случае со статическим сайтом.

Среди CMS-систем часто выделяют так называемые каркасы (content management framework, CMF) – инструментарии для создания системы.

Разработкой систем управления контентом занимаются многие компании, в том числе крупнейшие разработчики информационных технологий - IBM, Microsoft, Oracle, Macromedia.

В последнее время начали появляться организации, пытающиеся объединить разработчиков CMS, создать единую информационную среду для

потенциальных пользователей подобных систем, продвигать и утверждать единые стандарты. Прежде всего, это ассоциации OSCOM (Open Source Content Management), и CMSWatch.

OSCOM утвердила такие стандарты, как WebDav, RSS, ATOM и JSR-170.

В свою очередь, CMSWatch ежегодно выпускает отчет, включающий в себя обзор рынка CMS-систем, сравнение некоторых из них, описание жизненного цикла контента и управления им в CMS-системах.

Существует классификация CMS, основанная на модели представления данных – *объектной, сетевой или модульной* [18].

Объектная модель представления данных оперирует такими понятиями, как класс и объект. Классы определяют структуру данных и представляют собой набор атрибутов (текстовая строка, целое число, изображение и т.д.). Экземпляры класса (объекты) имеют определенную структуру и могут содержать другие объекты, образуя произвольную иерархическую структуру. Объекты могут наследовать свойства, содержание и поведение объектов, которые в них содержатся. Примерами объектов служат документы, картинки, папки и учетные записи пользователей. Класс контента не хранит в себе реальных данных – такую информацию содержат объекты (экземпляры класса). Определив один класс, можно создать множество его представителей (контент объектов).

В CMS-системах данные обычно хранятся в реляционной или объектной базе данных. В первом случае объектная модель данных отображается на реляционную модель базы данных.

Как правило, системы, основанные на объектно-ориентированной модели данных, наиболее функциональные, гибкие, но, в то же время, и наиболее сложные.

Сетевая модель представления данных опирается на теорию графов: структура информации представляется в виде узлов с помеченными связями между ними. Фундаментом системы может служить как сетевая, так и традиционная реляционная СУБД, на которую отображена сетевая модель описания данных. В реляционных таблицах хранится информация об узлах, их атрибутах и связях между ними. Связь отличается от атрибута тем, что в ней хранится ссылка на другой узел, а в атрибуте — собственно значение. Для извлечения данных из направленного графа обычно используются рекурсивные процедуры обработки, такие как составление списков узлов, определение атрибутов узла по атрибутам родителя и др.

В модульных системах контент разделен на отдельные модули по типам содержимого. Структура данных зависит от модуля, и вся работа с контентом сосредоточена внутри модуля. Модули независимы и полностью отвечают за работу с документами данного типа. Документы описываются с помощью фиксированного набора характеристик – типы документов строго фиксированы. Расширять функциональность можно за счет добавления нового модуля, замены или редактирования существующего

кода. Чаще всего нет никакой системы связей между документами разных модулей и между документами одного и того же модуля. Стандартный набор типов контента (модулей) таков: ссылки, статьи, файлы, новости, разделы, форум.

Несмотря на очевидную ограниченность модели данных, системы на ее основе наиболее популярны благодаря своей простоте. У модульных CMS-систем есть один общий недостаток – строго фиксированная в пределах модуля структура содержимого. Однако для расширения их функциональности можно воспользоваться внешними модулями, которых в Сети немало. Очевидное преимущество этих систем — возможность получения почти полностью готового к использованию портала за короткое время.

Рисунок 15 – Схема функционирования CMS

Основная идея систем управления контентом – разделение визуального дизайна сайта и его информационного наполнения. При создании сайта с помощью такой системы разрабатывается набор шаблонов страниц, в которых впоследствии размещается информация. В этом случае роль разработчиков (фактически это группа внедрения) ограничивается только созданием "начальной" информационной системы на основе системы управления контентом, затем пользователи сами публикуют требуемую информацию и определяют ее представление. Управление сайтом сводится к минимуму, – администратору остается только управлять пользователями.

Пользователи CMS делятся на две группы – создатели шаблонов страниц и авторы контента (информационного наполнения). Таким обра-

зом, одна группа пользователей создает структуру и оформление страниц, а другая наполняет его содержанием.

Функции систем управления контентом структурированы согласно жизненному циклу системы.

Сначала группа внедрения разворачивает ядро CMS и создает в СУБД информационное хранилище контента – БД. Далее администратор предоставляет доступ к системе различным пользователям, затем создается контент, он публикуется, и к нему применяются шаблоны оформления.

На первом этапе необходимо создать все типы контента и схемы их метаописаний, а также настроить систему на определенный поток работ (если система поддерживает создание потоков работ, а не использует единственный встроенный). Понятие типа контента аналогично понятию класса, а элементы контента представляют собой набор экземпляров таких "классов". Типами контента являются, например, текст и изображение; экземпляром контента конкретный документ или картинка.

Следующая важная возможность – хранение информации о версии контента. Это позволяет задать номер версии любых операций изменения контента и при необходимости восстановить его. В любой момент можно отказаться от изменений и, практически в режиме реального времени, откатиться на одну из предыдущих зафиксированных версий. Строгий контроль версий необходим для определения ответственности отдельных лиц, а также для резервного и аварийного восстановления системы.

Кроме управления контентом, система должна предоставлять возможность создавать метаданные о нем. *Метаданные* – это сведения о данных, свойства данных. Примером метаданных служат ключевые (характерные) слова документов, предназначенные для поисковых или отчетных систем. Системы управления контентом, рассматриваемые в данном обзоре, не поддерживают метаданные, хотя можно специально ввести дополнительные типы контента, представляющие собой метаданные.

После того, как все типы контента созданы, авторы информационного наполнения начинают создавать, изменять и удалять элементы контента указанного типа. CMS уже содержит некоторый набор визуальных компонентов, например, для редактирования текста, выбора изображений, выбора шаблона представления.

Кроме непосредственно редактирования элементов контента, необходимо предусмотреть разбиение контента по категориям или рубрикам.

В качестве решения проблемы представления в системах управления контентом используется технология *шаблонов*, определяющих внешний вид страницы. Разработчику шаблонов не нужно знать никаких технических тонкостей. На ранних этапах существования WWW шаблоны представляли "заготовки" HTML-кода, из которого путем манипуляций в HTML-редакторе получались готовые страницы. Сегодня такими заготовками манипулируют уже не дизайнеры в своих редакторах, а серверные web-приложения. Таким образом, современный шаблон Web-страницы

представляет собой блок HTML, который благодаря специальным тегам или внедренным сценариям, облегчает включение динамически сгенерированного содержания на этапе выполнения. При использовании подобных шаблонов программистам необходим некоторый стандартизированный интерфейс для работы с ними – шаблонный движок (в английском языке существует устоявшийся термин – template engine), который может иметь разнообразные дополнительные функции, например, поддерживать кэширование шаблонов, их динамическое обновление и т.д.

Механизм публикации информации в системе управления контентом отвечает за процесс создания, редактирования и удаления шаблонов страниц, а также за сопоставление типов контента и шаблонов страниц. В состав дополнительных возможностей системы публикации может входить предварительная генерация статической версии сайта. Эта опция очень полезна в случае размещения информационной системы на оборудовании с ограниченными возможностями.

Типичный процесс публикации информации в World Wide Web реализован в Microsoft Content Management Server. Обычным приемом обеспечения оформления информационного наполнения являются шаблоны представления информации. Поэтому первым этапом процесса является создание наборов шаблонов. Типичный шаблон содержит разметку HTML и места, куда в дальнейшем будут вставлены данные (placeholder'ы в терминологии Microsoft). Далее на основе этих шаблонов авторы информационного наполнения создают страницы и представляют их редакторам для одобрения. Редакторы, в свою очередь, могут либо отклонить страницу и вернуть ее автору на доработку, либо одобрить ее и передать модератору сайта. В первом случае процесс повторяется снова, во втором же модератор сайта проверяет расположение страницы на сайте, дату и срок ее публикации. Если все в порядке, страница становится видна пользователям. Несмотря на то, что рабочий процесс в Microsoft Content Management Server фиксирован и не может быть изменен в дальнейшем, подобное решение подходит большинству пользователей, которым необходимо публиковать информацию в World Wide Web.

Управление пользователями включает создание, изменение и удаление учетных записей отдельных пользователей и их групп, а также назначение прав для работы с элементами контента. Важной частью требований является наличие пользовательских профилей (profiles), с помощью которых можно сгенерировать персональное представление информации для каждого пользователя. Полезной является и возможность пользователя делегировать свои права. Это позволяет пользователям переназначать исполнителя конкретной работы и избегать простоев из-за отсутствия отдельного лица.

Системы управления контентом управляют учетными записями пользователей на основе собственных групп, не используя существующие идентификационные системы, например, Windows. Аутентификация сред-

ствами Windows позволила бы значительно упростить администрирование. При этом система управления контентом могла бы использовать операционную систему локального компьютера или контролера домена для проверки и сопровождения учетной записи пользователя.

Представление информации создается на основе данных, а также предпочтений конкретного пользователя. Персонификация достигается путем использования профилей – специальных записей, в которых хранится информация, специфичная для конкретных пользователей.

В самом общем виде архитектуру систем управления Web-контентом можно представить следующим образом (рис. 16).

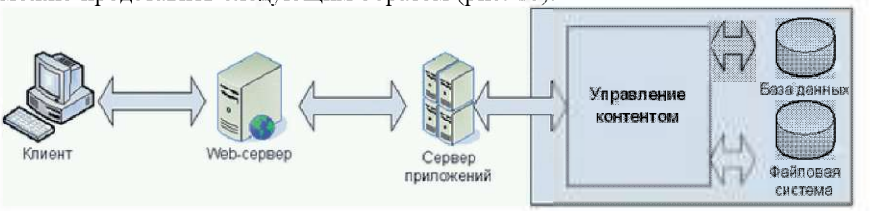


Рисунок 16 – Архитектура CMS

В основе данной технологии лежит трехзвенная архитектура клиент/сервер. Такая архитектура разбивает процесс обработки данных между клиентом, сервером приложений и хранилищем данных. В отличие от традиционной двухзвенной архитектуры здесь присутствует сервер приложений как промежуточное звено между клиентом и хранилищем данных.

В системе присутствует два хранилища. В первом (обычно реляционная СУБД) хранятся все данные, которые публикуются на сайте. Во втором (обычно файловая система) хранятся элементы представления – шаблоны, графические изображения и т.д.

Получая запрос, сервер приложений обрабатывает его, связываясь с хранилищем данных, в каком бы месте необходимые данные не находились. Клиент лишь получает результат в виде HTML-файла. Таким образом, сервер приложений является стандартизованной платформой для динамической доставки контента и построения основных приложений. Серверов приложений может быть много, а связь с ними происходит через Web-сервер.

На сегодняшний день существует множество систем управления контентом. Все CMS можно разделить на **платные** и **бесплатные**. Основным и самым главным преимуществом платных CMS является полноценная техническая поддержка. Это значит, что администратор в любой момент может позвонить или написать специалистам, которые занимались разработкой CMS, и задать все интересующие вопросы, на которые впоследствии должен получить грамотные и развернутые ответы. Кроме того, что Вы покупаете полностью настроенный и готовый к работе продукт, кото-

рый не требует никаких посторонних вмешательств в настройки, чего нельзя сказать о бесплатных системах управления контентом. Другими словами, после установки данной CMS на сервер, можно сразу же приступать к созданию сайта, и этот сайт будет корректно выполнять свои функции. Например, если Вы создаёте интернет-магазин, то все функции покупки, оплаты и доставки будут однозначно правильно работать, и Вам ничего не нужно будет менять в настройках, а уж тем более, в исходном коде.

Главным преимуществом бесплатных систем управления контентом является то, что они бесплатные. Кроме того, в интернете большое число бесплатных плагинов, модулей и шаблонов для этих CMS, с помощью которых можно создать сайт абсолютно любой сложности. Однако со всем этим разнообразием дополнений Вам придется разбираться самостоятельно, и если Вы собираетесь сделать более-менее серьёзный сайт на бесплатной CMS, то Вам не обойтись без базовых знаний HTML, CSS, а может быть и PHP.

Если взять в качестве примера интернет-магазин, то его разработка на платной CMS займёт у начинающего пользователя около недели. Такой же магазин на бесплатной системе управления контентом придется разрабатывать больше времени. Таким образом, если у Вас много свободного времени, есть навыки работы с компьютером, желание и интерес много работать, то есть все шансы создать сайт, который не будет уступать по красоте и функционалу проектам, созданным на платных CMS.

Рассмотрим основные виды платных и бесплатных CMS, самые популярные и часто используемые. Среди бесплатных выделяются следующие:

1. Joomla! Пожалуй, самая популярная на сегодняшний день бесплатная система управления контентом. Имеет множество модулей, плагинов и дополнений. Вместе с этим Joomla! имеет много ненужных вещей, которые снижают производительность системы, а также создаёт множество дублей страниц, что не очень хорошо для продвижения.
2. WordPress. Еще одна не менее популярная система. Изначально, разрабатывалась для создания блогов. Но, как показала практика, с помощью WordPress можно создавать корпоративные сайты, в том числе интернет-магазины. Для WordPress также создано множество бесплатных плагинов, имеется документацию на русском языке.
3. Drupal. Еще одна бесплатная система управления контентом. В большей степени это новостная система. Подходит для создания интернет-сообществ, форумов или блогов, однако на ней можно создать абсолютно любой сайт.
4. Ucoz. Здесь всё совсем просто. Чтобы создать сайт, необходимо просто зарегистрироваться на официальном сайте Ucoz. Вам даже не обязательно знать HTML. Ucoz – это конструктор сайтов в самом прямом смысле этого слова. Кроме того, Вам даже не придётся тра-

титься на хостинг, ucoz его предоставит. Впрочем, сервис является условно бесплатным, и если Вы хотите, чтобы Ваш сайт был в безрекламном пространстве, и он не относится к категории социальных, то лучше и в этом сервисе пользоваться платными услугами.

5. PHPShop.CMS Free. Полностью бесплатная система с открытым исходным кодом. На сайте разработчика можно найти шаблоны и модули, которые встраиваются в CMS. Подходит для создания коммерческих корпоративных сайтов, а самое интересное – можно создавать интернет-магазины. Техподдержка организована в виде форума.
6. Wolf CMS. Активно развивающаяся CMS, распространяется на условиях Free Software и опубликованная под GNU General Public License v.3. Написана, как и большинство современных систем, на PHP. Главным преимуществом этой системы является низкая требовательность к ресурсам и простота использования. Исходный код сайта виден прямо в панели администратора. Однако для работы с ней потребуются некоторые знания HTML и PHP.
7. OpenCMS. Созданный группой европейских разработчиков бесплатная система, которая подойдйт для профессиональной разработки сайтов. Как и в Wolf CMS, здесь не так много готовых модулей и шаблонов, поэтому, по большей части, придется всё делать самостоятельно. Довольно быстрая и безопасная система.

К популярным платным система относятся:

1. 1С-Битрикс. Самая популярная на сегодняшний день платная система управления контентом, на которой построен в том числе и официальный сайт Юго-Западного государственного университета. Система отличается большой разницей стоимости версий, которая заключается в наличии тех или иных модулей. Так, на базовой лицензии можно сделать, максимум, сайт-визитку. А для того, чтобы создать интернет-магазин, потребуется существенно более дорогая лицензия. Но для организации это не очень большие деньги, поэтому профессиональные разработчики сайтов активно используют эту CMS.
2. NetCat. Для её использования не требуется специальных знаний языков программирования или разметки. В системе реализованы различные функциональные возможности, например, интерактивное общение с посетителями сайта. Стоит эта CMS гораздо дешевле Битрикс.
3. UMI.CMS. «Коробочная» CMS, которая позволяет управлять сайтом и контентом без входа в административный интерфейс. Таким образом, по заверению разработчиков, сайт может администрировать даже ребёнок. Но на деле всё оказывается не так просто. Проблемы начинаются уже на стадии установки на хостинг, интерфейс также не отличается удобством.

4. DataLife Engine подходит для новостных сайтов и имеет среднюю стоимость. Можно найти достаточно модулей расширения, чтобы создать многофункциональный сайт. Очень хорошо оптимизирован для продвижения SEO-специалистами. Можно использовать бесплатную demo-версию, только с некоторыми ограничениями – закрыт исходный код, ограничение на количество новостей и комментариев.
5. HostCMS. Коммерческая система управления контентом, которая имеет несколько вариантов лицензирования, в том числе бесплатную версию. В бесплатной версии отсутствуют следующие модули: поиск по сайту, формы, резервное копирование, пользователи сайта, файловый менеджер, форумы, реклама и несколько других.
6. Amiro.CMS. Разработчики позиционируют эту систему, как самую быструю CMS рунета. Система позволяет создавать и поддерживать сайты любого уровня сложности. Одной из особенностей данной CMS можно считать быструю настройку. Так, с готовым шаблоном можно соорудить сайт за несколько часов.

В заключение можно сказать, что каждая существующая система имеет свои достоинства и недостатки, поэтому каждый системный администратор по-своему определяет, какой продукт будет наиболее подходящим для организации и для управления контентом.

Контрольные вопросы

1. Что такое веб-служба?
2. Что такое URL?
3. Что такое URI?
4. Что такое IRC?
5. Что такое FTP?
6. Что такое WWW?
7. Назовите функции криптографических систем.
8. Что означает понятие симметричного шифрования?
9. Что означает понятие асимметричного шифрования?
10. Что такое электронная подпись?
11. Что такое хэш-функция?
12. Что такое электронный сертификат?
13. Какие требования защиты операций обеспечивает SET?

Задание

На виртуальной машине с установленной операционной системой Windows Server добавить роль веб-сервера и сервера приложений с поддержкой веб-сервера (IIS), общим доступом к TCP-портам и активацией через HTTP. Для веб-сервера должна быть установлена служба FTP-сервера. С помощью диспетчера служб IIS создать тестовый сайт, размещаемый на данном сервере.

Зарегистрироваться на сервисе ucoz.ru, создать тестовый сайт, изучить средства создания и управления сайтом на данной платформе, после чего удалить сайт.

8. ВИРТУАЛИЗАЦИЯ

В широком смысле, понятие виртуализации представляет собой сокрытие настоящей реализации какого-либо процесса или объекта от истинного его представления для того, кто им пользуется [19]. Продуктом виртуализации является нечто удобное для использования, на самом деле, имеющее более сложную или совсем иную структуру, отличную от той, которая воспринимается при работе с объектом. Иными словами, происходит отделение представления от реализации чего-либо. В компьютерных технологиях под термином «виртуализация» обычно понимается абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию. Проще говоря, пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как объект устроен в действительности.

Сам термин «виртуализация» в компьютерных технологиях появился в шестидесятых годах прошлого века вместе с термином «виртуальная машина», означающим продукт виртуализации программно-аппаратной платформы. В то время виртуализация была, скорее, интересной технической находкой, чем перспективной технологией. Разработки в сфере виртуализации в шестидесятых-семидесятых годах проводились только компанией IBM. С появлением в компьютере IBM M44/44X экспериментальной системы пэйджинга, впервые был употреблен термин «виртуальная машина» (virtual machine), который заменил более ранний термин «псевдомашина» (pseudo machine). Затем в мэйнфреймах IBM серии System 360/370, можно было использовать виртуальные машины для сохранения предыдущих версий операционных систем. До конца девяностых годов никто кроме IBM так и не решался использовать эту оригинальную технологию всерьез. Однако в девяностых годах стали очевидны перспективы подхода виртуализации: с ростом аппаратных мощностей, как персональных компьютеров, так и серверных решений, вскоре представится возможность использовать несколько виртуальных машин на одной физической платформе.

В 1997 году компания Connectix выпустила первую версию Virtual PC для платформы Macintosh, а в 1998 году VMware запатентовала свои техники виртуализации. Компания Connectix впоследствии была куплена корпорацией Microsoft, а VMware корпорацией EMC, и на данный момент обе эти компании являются двумя основными потенциальными конкурентами на рынке технологий виртуализации в будущем. Потенциальными – потому что сейчас VMware безоговорочный лидер на этом рынке, однако у Microsoft, как всегда, есть козырь в рукаве.

Со времени своего появления термины «виртуализация» и «виртуальная машина» приобрели множество различных значений и употреблялись в

разных контекстах. Давайте попробуем разобраться с тем, что такое виртуализация на самом деле.

Понятие виртуализации условно можно разделить на две фундаментально различающиеся категории:

- **виртуализация платформ** – продуктом этого вида виртуализации являются виртуальные машины – некие программные абстракции, запускаемые на платформе реальных аппаратно-программных систем;
- **виртуализация ресурсов** – данный вид виртуализации преследует своей целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей и т. п.

Рисунок 17 – Виды виртуализации

Под виртуализацией платформ понимают создание программных систем на основе существующих аппаратно-программных комплексов, зависящих или независящих от них. Система, предоставляющая аппаратные ресурсы и программное обеспечение, называется хостовой (host), а симулируемые ей системы — гостевыми (guest). Чтобы гостевые системы могли стабильно функционировать на платформе хостовой системы, необходимо, чтобы программное и аппаратное обеспечение хоста было достаточно надежным и предоставляло необходимый набор интерфейсов для доступа к его ресурсам. Есть несколько видов виртуализации платформ, в каждом из которых осуществляется свой подход к понятию «виртуализация». Виды виртуализации платформ зависят от того, насколько полно осуществляется симуляция аппаратного обеспечения. До сих пор нет единого соглашения о терминах в сфере виртуализации, поэтому некоторые из приведенных далее видов виртуализации могут отличаться от тех, что предоставят другие источники.

Различают следующие виды виртуализации платформ:

1. *Полная эмуляция (симуляция).*

   При таком виде виртуализации виртуальная машина полностью виртуализует все аппаратное обеспечение при сохранении гостевой операционной системы в неизменном виде. Такой подход позволяет эмулировать различные аппаратные архитектуры. Например, можно

запускать виртуальные машины с гостевыми системами для x86-процессоров на платформах с другой архитектурой (например, на RISC-серверах компании Sun). Долгое время такой вид виртуализации использовался, чтобы разрабатывать программное обеспечение для новых процессоров еще до того, как они были физически доступными. Такие эмуляторы также применяют для низкоуровневой отладки операционных систем. Основной минус данного подхода заключается в том, что эмулируемое аппаратное обеспечение весьма и весьма существенно замедляет быстродействие гостевой системы, что делает работу с ней очень неудобной, поэтому, кроме как для разработки системного программного обеспечения, а также образовательных целей, такой подход мало где используется. Примеры продуктов для создания эмуляторов: Bochs, PearPC, QEMU (без ускорения), Hercules Emulator.

2. *Частичная эмуляция (нативная виртуализация).*
   В этом случае виртуальная машина виртуализует лишь необходимое количество аппаратного обеспечения, чтобы она могла быть запущена изолированно. Такой подход позволяет запускать гостевые операционные системы, разработанные только для той же архитектуры, что и у хоста. Таким образом, несколько экземпляров гостевых систем могут быть запущены одновременно. Этот вид виртуализации позволяет существенно увеличить быстродействие гостевых систем по сравнению с полной эмуляцией и широко используется в настоящее время. Кроме того, в целях повышения быстродействия в платформах виртуализации, использующих данный подход, применяется специальная «прослойка» между гостевой операционной системой и оборудованием (гипервизор), позволяющая гостевой системе напрямую обращаться к ресурсам аппаратного обеспечения. Гипервизор, называемый также «Монитор виртуальных машин» (Virtual Machine Monitor) – одно из ключевых понятий в мире виртуализации. Применение гипервизора, являющегося связующим звеном между гостевыми системами и аппаратурой, существенно увеличивает быстродействие платформы, приближая его к быстродействию физической платформы. К минусам данного вида виртуализации можно отнести зависимость виртуальных машин от архитектуры аппаратной платформы. Примеры продуктов для нативной виртуализации: VMware Workstation, VMware Server, VMware ESX Server, Virtual Iron, Virtual PC, VirtualBox, Parallels Desktop и другие.
3. *Частичная виртуализация*, а также «виртуализация адресного пространства» («address space virtualization»).
   При таком подходе, виртуальная машина симулирует несколько экземпляров аппаратного окружения (но не всего), в частности, пространства адресов. Такой вид виртуализации позволяет совместно использовать ресурсы и изолировать процессы, но не позволяет раз-

делять экземпляры гостевых операционных систем. Строго говоря, при таком виде виртуализации пользователем не создаются виртуальные машины, а происходит изоляция каких-либо процессов на уровне операционной системы. В данный момент многие из известных операционных систем используют такой подход. Примером может послужить использование UML (User-mode Linux), в котором «гостевое» ядро запускается в пользовательском пространстве базового ядра (в его контексте).

4. *Паравиртуализация.*
   При применении паравиртуализации нет необходимости симулировать аппаратное обеспечение, однако, вместо этого (или в дополнение к этому), используется специальный программный интерфейс (API) для взаимодействия с гостевой операционной системой. Такой подход требует модификации кода гостевой системы, что, с точки зрения сообщества, Open Source не так и критично. Системы для паравиртуализации также имеют свой гипервизор, а API-вызовы к гостевой системе, называются «hypercalls» (гипервызовы). Многие сомневаются в перспективах этого подхода виртуализации, поскольку в данный момент все решения производителей аппаратного обеспечения в отношении виртуализации направлены на системы с нативной виртуализацией, а поддержку паравиртуализации приходится искать у производителей операционных систем, которые слабо верят в возможности предлагаемого им средства. В настоящее время провайдерами паравиртуализации являются компании XenSource и Virtual Iron, утверждающие, что быстродействие паравиртуализации выше.
5. *Виртуализация уровня операционной системы.*
   Сутью данного вида виртуализации является виртуализация физического сервера на уровне операционной системы в целях создания нескольких защищённых виртуализованных серверов на одном физическом. Гостевая система, в данном случае, разделяет использование одного ядра хостовой операционной системы с другими гостевыми системами. Виртуальная машина представляет собой окружение для приложений, запускаемых изолированно. Данный тип виртуализации применяется при организации систем хостинга, когда в рамках одного экземпляра ядра требуется поддерживать несколько виртуальных серверов клиентов.
   Примеры виртуализации уровня ОС: Linux-VServer, Virtuozzo, OpenVZ, Solaris Containers и FreeBSD Jails.
6. *Виртуализация уровня приложений.*
   Этот вид виртуализации не похож на все остальные: если в предыдущих случаях создаются виртуальные среды или виртуальные машины, использующиеся для изоляции приложений, то в данном случае само приложение помещается в контейнер с необходимыми эле-

ментами для своей работы: файлами реестра, конфигурационными файлами, пользовательскими и системными объектами. В результате получается приложение, не требующее установки на аналогичной платформе. При переносе такого приложения на другую машину и его запуске, виртуальное окружение, созданное для программы, разрешает конфликты между ней и операционной системой, а также другими приложениями. Такой способ виртуализации похож на поведение интерпретаторов различных языков программирования (недаром интерпретатор, Виртуальная Машина Java (JVM), тоже попадает в эту категорию). Примером такого подхода служат: Thinstall, Altiris, Trigence, Softricity.

При описании виртуализации платформ мы рассматривали понятие виртуализации в узком смысле, преимущественно применяя его к процессу создания виртуальных машин. Однако если рассматривать виртуализацию в широком смысле, можно прийти к понятию виртуализации ресурсов, обобщающим в себе подходы к созданию виртуальных систем. Виртуализация ресурсов позволяет концентрировать, абстрагировать и упрощать управление группами ресурсов, таких как сети, хранилища данных и пространства имен. В сфере виртуализации ресурсов выделяют следующие виды:

1. *Объединение, агрегация и концентрация компонентов.*

   Под таким видом виртуализации ресурсов понимается организация нескольких физических или логических объектов в пулы ресурсов (группы), представляющих удобные интерфейсы пользователю. Примеры такого вида виртуализации:

   1. многопроцессорные системы, представляющиеся нам как одна мощная система,
   2. RAID-массивы и средства управления томами, комбинирующие несколько физических дисков в один логический,
   3. виртуализация систем хранения, используемая при построении сетей хранения данных SAN (Storage Area Network),
   4. виртуальные частные сети (VPN) и трансляция сетевых адресов (NAT), позволяющие создавать виртуальные пространства сетевых адресов и имен.

Кластеризация компьютеров и распределенные вычисления (grid computing).

Этот вид виртуализации включает в себя техники, применяемые при объединении множества отдельных компьютеров в глобальные системы (метакомпьютеры), совместно решающие общую задачу.

Разделение ресурсов (partitioning).

При разделении ресурсов в процессе виртуализации происходит разделение какого-либо одного большого ресурса на несколько однотипных объектов, удобных для использования. В сетях хранения данных это называется зонированием ресурсов («zoning»).

Инкапсуляция.

Многим это слово известно как сокрытие объектом внутри себя своей реализации. Применительно к виртуализации, можно сказать, что это процесс создания системы, предоставляющей пользователю удобный интерфейс для работы с ней и скрывающей подробности сложности своей реализации. Например, использование центральным процессором кэша для ускорения вычислений не отражается на его внешних интерфейсах.

Виртуализация ресурсов, в отличие от виртуализации платформ, имеет более широкий и расплывчатый смысл и представляет собой массу различных подходов, направленных на повышение удобства обращения пользователей с системами в целом. Поэтому, далее мы будем опираться в основном на понятие виртуализации платформ, поскольку технологии, связанные именно с этим понятием, являются в данный момент наиболее динамично развивающимися и эффективными.

Виртуализация операционных систем за последние три-четыре года очень хорошо продвинулась вперед, как в технологическом, так и в маркетинговом смысле. С одной стороны, пользоваться продуктами виртуализации стало намного проще, они стали более надежными и функциональными, а с другой – нашлось немало новых интересных применений виртуальным машинам. Сферу применения виртуализации можно определить, как «место, где есть компьютеры», однако на данный момент можно обозначить следующие варианты использования продуктов виртуализации:

1. *Консолидация серверов.*

   В данный момент приложения, работающие на серверах в IT-инфраструктуре компаний, создают небольшую нагрузку на аппаратные ресурсы серверов (в среднем 5 – 15 процентов). Виртуализация позволяет мигрировать с этих физических серверов на виртуальные и разместить их все на одном физическом сервере, увеличив его загрузку до 60-80 процентов и, повысив тем самым коэффициент использования аппаратуры, что позволяет существенно сэкономить на аппаратуре, обслуживании и электроэнергии.
2. *Разработка и тестирование приложений.*

   Множество продуктов виртуализации позволяют запускать несколько различных операционных систем одновременно, позволяя тем самым разработчикам и тестерам программного обеспечения тестировать их приложения на различных платформах и конфигурациях. Также удобные средства по созданию «снимков» текущего состояния системы одним кликом мыши и такого же простого восстановления из этого состояния, позволяют создавать тестовые окружения для различных конфигураций, что существенно повышает скорость и качество разработки.
3. *Использование в бизнесе.*

Этот вариант использования виртуальных машин является наиболее обширным и творческим. К нему относится все, что может понадобиться при повседневном обращении с IT-ресурсами в бизнесе. Например, на основе виртуальных машин можно легко создавать резервные копии рабочих станций и серверов (просто скопировав папку), строить системы, обеспечивающие минимальное время восстановления после сбоев и т. п. К данной группе вариантов использования относятся все те бизнес-решения, которые используют основные преимущества виртуальных машин.

4. *Использование виртуальных рабочих станций*.
   С приходом эры виртуальных машин будет бессмысленно делать себе рабочую станцию с ее привязкой к аппаратуре. Теперь создав однажды виртуальную машину со своей рабочей или домашней средой, можно будет использовать её на любом другом компьютере. Также можно использовать готовые шаблоны виртуальных машин (Virtual Appliances), которые решают определенную задачу (например, сервер приложений). Концепция такого использования виртуальных рабочих станций может быть реализована на основе хост-серверов для запуска на них перемещаемых десктопов пользователей (нечто подобное мэйнфреймам). В дальнейшем эти десктопы пользователь может забрать с собой, не синхронизируя данные с ноутбуком. Этот вариант использования также предоставляет возможность создания защищенных пользовательских рабочих станций, которые могут быть использованы, например, для демонстрации возможностей программы заказчику. Можно ограничить время использования виртуальной машины – и по прошествии этого времени виртуальная машина перестанет запускаться. В этом варианте использования заложены большие возможности.

Все перечисленные варианты использования виртуальных машин фактически являются лишь сферами их применения в данный момент, со временем, несомненно, появятся новые способы заставить виртуальные машины работать в различных отраслях IT. Но давайте посмотрим, как сейчас обстоят дела с виртуализацией.

На сегодняшний день проекты по виртуализации IT-инфраструктуры активно внедряются многими ведущими компаниями, занимающимися системной интеграцией и являющимися авторизованными партнерами провайдеров систем виртуализации. В процессе виртуализации IT-инфраструктуры создается виртуальная инфраструктура – комплекс систем на основе виртуальных машин, обеспечивающих функционирование всей IT-инфраструктуры, обладающий многими новыми возможностями при сохранении существующей схемы деятельности IT-ресурсов. Вендоры различных платформ виртуализации готовы предоставить информацию об успешных проектах по внедрению виртуальной инфраструктуры в крупных банках, промышленных компаниях, больницах, образовательных учрежде-

ниях. Множество достоинств виртуализации операционных систем позволяют компаниям экономить на обслуживании, персонале, аппаратном обеспечении, обеспечении бесперебойной работы, репликации данных и восстановлении после сбоев. Также рынок виртуализации начинает наполняться мощными средствами управления, миграции и поддержки виртуальных инфраструктур, позволяющими использовать преимущества виртуализации наиболее полно. Давайте посмотрим, как именно виртуализация позволяет компаниям, внедряющим у себя виртуальную инфраструктуру, экономить деньги.

Сегодня можно привести 10 причин использовать виртуальные машины:

1. Экономия на аппаратном обеспечении при консолидации серверов. Существенная экономия на приобретении аппаратного обеспечения происходит при размещении нескольких виртуальных серверов на одном физическом сервере. В зависимости, от вендора платформы виртуализации, доступны возможности по балансировке рабочей нагрузки, контролю выделяемых ресурсов, миграции между физическими хостами и восстановлению данных. Все это влечет за собой реальную экономию денежных средств на обслуживании, управлении и администрировании инфраструктуры серверов.
2. Возможность поддержания старых операционных систем в целях обеспечения совместимости. При выходе новой версии операционной системы, старую версию можно поддерживать на виртуальной машине, пока не будет полностью обкатана новая ОС. И наоборот, можно «поднять» новую ОС на виртуальной машине и опробовать ее без ущерба для основной системы.
3. Возможность изолировать потенциально опасные окружения. Если какое-то приложение или компонент вызывает сомнения в его надежности и защищенности, можно использовать его на виртуальной машине без опасности повредить жизненно важные компоненты системы. Такую изолированную среду называют также «песочницей» (sandbox). Помимо этого, можно создавать виртуальные машины, ограниченные политиками безопасности (например, машина перестанет запускаться через две недели).
4. Возможность создания требуемых аппаратных конфигураций. Иногда требуется использовать заданную аппаратную конфигурацию (процессорное время, количество выделяемой оперативной и дисковой памяти) при проверке работоспособности приложений в определенных условиях. Довольно сложно без виртуальной машины «загнать» физическую машину в такие условия. В виртуальных машинах — это пара кликов мыши.
5. Виртуальные машины могут создавать представления устройств, которых у вас нет. Например, многие системы виртуализации позволяют создавать виртуальные SCSI диски, виртуальные многоядерные

процессоры и т. п. Это может пригодиться для создания различного рода симуляций.

6. На одном хосте может быть запущено одновременно несколько виртуальных машин, объединенных в виртуальную сеть. Такая особенность предоставляет безграничные возможности по созданию моделей виртуальной сети между несколькими системами на одном физическом компьютере. Особенно это необходимо, когда требуется смоделировать некую распределенную систему, состоящую из нескольких машин. Также можно создать несколько изолированных пользовательских окружений (для работы, развлечений, работы в Интернет), запустить их и переключаться между ними по мере необходимости выполнения тех или иных задач.
7. Виртуальные машины предоставляют великолепные возможности по обучению работе с операционными системами. Можно создать репозиторий готовых к использованию виртуальных машин с различными гостевыми операционными системами и запускать их по мере необходимости в целях обучения. Их можно безнаказанно подвергать всяческим экспериментам, поскольку в случае порчи системы, её восстановление из сохраненного состояния займет пару минут.
8. Виртуальные машины повышают мобильность. Папка с виртуальной машиной может быть перемещена на другой компьютер, и там виртуальная машина может быть сразу запущена. Не требуется создавать никаких образов для миграции, и, к тому же, виртуальная машина отвязана от конкретной аппаратуры.
9. Виртуальные машины могут быть организованы в «пакеты приложений». Вы можете создавать виртуальной окружение для конкретного варианта использования (например, дизайнерскую машину, машину менеджера и т. п.), установив в ней все требуемое программное обеспечение, и разворачивать десктопы по мере необходимости.
10. Виртуальные машины более управляемы. При использовании виртуальных машин существенно повышается управляемость в отношении создания резервных копий, создания снимков состояний виртуальных машин и восстановлений после сбоев.

На этом, конечно, достоинства виртуальных машин не исчерпываются, это лишь пища для размышления и исследования их возможностей. Безусловно, как и у всякого нового и перспективного решения, у виртуальных машин есть и свои недостатки:

1. Невозможность эмуляции всех устройств. В данный момент все основные устройства аппаратных платформ поддерживаются вендорами систем виртуализации, однако если вы используете, например, какие-либо контроллеры или устройства, не поддерживаемые ими, придется отказаться от виртуализации такого окружения.
2. Виртуализация требует дополнительных аппаратных ресурсов. В настоящее время использование различных техник виртуализации по-

зволило приблизить показатели быстродействия виртуальных машин к реальным, однако, чтобы физический хост смог запускать хотя бы пару виртуальных машин, требуется достаточное для них количество аппаратных ресурсов.

3. Некоторые платформы виртуализации требовательны к конкретному аппаратному обеспечению. В частности, замечательная платформа компании VMware, ESX Server, была бы и вовсе замечательной, если бы не предъявляла жестких требований к аппаратному обеспечению.
4. Хорошие платформы виртуализации стоят хороших денег. Порой, стоимость развертывания одного виртуального сервера равна стоимости еще одного физического, в определенных условиях это может оказаться нецелесообразным. К счастью, есть множество бесплатных решений, но они, в основном, ориентированы на домашнего пользователя и малый бизнес.

Несмотря на перечисленные и вполне устранимые недостатки, виртуализация продолжает набирать обороты.

Контрольные вопросы

1. Что такое виртуализация?
2. Что понимается под виртуализацией платформ?
3. Что понимается под виртуализацией ресурсов?
4. В чём состоит полная эмуляция?
5. Что происходит при частичной эмуляции?
6. Что происходит при частичной виртуализации?
7. Что такое паравиртуализация?
8. В чём суть виртуализации уровня операционной системы?
9. Что означает виртуализация уровня приложений?
10. Какие существуют виды виртуализации ресурсов?
11. Для чего может применяться виртуализация?
12. Какие недостатки имеют виртуальные машины?

Задание

Создайте на бесплатной платформе Oracle VirtualBox виртуальные машины и установите на них дистрибутивы операционной системы Linux: Ubuntu, Alt и Mint.

ЗАКЛЮЧЕНИЕ

В данном учебном пособии были рассмотрены главные теоретические вопросы, которыми должен владеть любой системный администратор. Одним из факторов, определяющих его уровень, является владение терминологией построения и управления компьютерными сетями, теоретической базой сетевых технологий, а также администрирования баз данных, которые также являются частью информационных систем и технологий, требующих грамотного системного администратора.

Вместе с этим стоит отметить, что владения теоретической базой не хватит для того, чтобы полноценно работать системным администратором. Многие вещи, рассмотренные и не рассмотренные в данном учебном пособии желательно опробовать на практике, то есть поработать, что называется, руками. В особенности это касается работы с командами интерфейса командной строки, которыми нужно владеть не хуже языка программирования, так как неправильно набранная команда не будет распознана операционной системой подобно тому, как компилятор не скомпилирует программу, написанную с нарушениями синтаксиса языка программирования. Поэтому начинающим системным администраторам рекомендуется начинать с графических средств администрирования типа оболочки Диспетчер серверов, автоматически запускаемой при старте операционной системы Windows Server. В процессе работы можно осваивать командную строку, и только убедившись, что овладение командами достигло требуемого уровня, можно переходить на использование Windows PowerShell и других средств применения DOS-команд, в том числе на работу с консольной версией операционной системы Windows Server.

Кроме того, осваивать серверные и другие незнакомые операционные системы лучше всего с помощью программных средств виртуализации, которые значительно упрощают управление новой операционной системой и не приведут к неполадкам в компьютере хоста, к которым может привести настоящая установка системы на физический жёсткий диск. Чем большим количеством операционных систем владеет администратор, тем выше его профессиональный уровень. Особенно приветствуется многими компаниями владение UNIX-подобными операционными системами, которые требуют отдельного рассмотрения, поэтому не вошли в данное учебное пособие.

Несмотря на то, что не все вопросы сферы системного администрирования были рассмотрены в пособии, данная книга может быть использована для закладки теоретической базы системного администратора широкого профиля.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Финанби – финансы и бизнес [Электронный ресурс] // Режим доступа: https://www.finanbi.ru/sistemnyi-administrator-doljnostnaya-instrukciya-obyazannosti-funkcii-i-trebovaniya-316. Дата обращения: 10.07.18.
2. CyberPedia. Информационный ресурс [Электронный ресурс] // Режим доступа: https://cyberpedia.su/10x8d7b.html. Дата обращения: 15.12.18.
3. Борисов, Д.Н. Корпоративные информационные системы [Текст]: учебно-методическое пособие / Д.И. Борисов. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2007. - 98 с.
4. Российский новый университет. Администрирование в информационных системах (лекция 1) [Электронный ресурс] // Режим доступа: http://rdv.rosnou.ru/IT433/it_ad_01.pdf. Дата обращения: 10.11.18.
5. Ikasteko.ru – IT-сфера, программное обеспечение и многое другое. Эталонная модель OSI [Электронный ресурс] // Режим доступа http://www.ikasteko.ru/page/etalonnaja-model-osi. Дата обращения: 8.09.2018.
6. Mircosoft Docs. Протокол IP версии 6 [Электронный ресурс] // Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/framework/network-programming/internet-protocol-version-6. Дата обращения: 2.12.2018.
7. Вычислительные сети. Сетевая маршрутизация [Электронный ресурс] // Режим доступа: http://just-networks.ru/seti-tcp-ip/marshrutizatsiya. Дата обращения: 4.12.2018.
8. Национальный открытый университет ИНТУИТ. Служба каталогов Active Directory [Электронный ресурс] // Режим доступа: http://www.intuit.ru/studies/courses/966/216/lecture/5567. Дата обращения: 25.11.2019.
9. Регистратор доменных имён РЕГ.РУ [Электронный ресурс] // Режим доступа: https://www.reg.ru/. Дата обращения: 15.11.2018.
10. База знаний osLogic.ru. Основные понятия DHCP [Электронный ресурс] // Режим доступа: https://www.oslogic.ru/ knowledge/ 761/ osnovnye-ponyatiya-dhcp/. Дата обращения: 30.11.2018.
11. Сурков, Л.В. Технология Network Address Translation [Электронный ресурс]: методические указания к лабораторной работе по курсу «Корпоративные сети» / Л.В. Сурков. - М.: МГТУ им. Баумана, 2011. - 10 с. – Режим доступа: e-learning.bmstu.ru. Дата обращения: 1.12.2018.
12. Mircosoft Docs. Документация по SQL Server [Электронный ресурс] // Режим доступа: https://docs.microsoft.com/ru-ru/sql/sql-

server/sql-server-technical-documentation?view=sql-server-2017. Дата обращения: 5.12.2018.
13. КриптоПро. Цифровой сертификат [Электронный ресурс] // Режим доступа: http://www.cryptopro.ru/support/articles/2005/07/dig-cert. Дата обращения: 9.12.2018.
14. Кафедра цифровой экономики Поволжского государственного университета телекоммуникаций и информатики. История электронной коммерции [Элект ронный ресурс] // Режим доступа: http://www.цифрономика.рф/ istoriya-elektronnoy-kommertsii.php. Дата обращения: 12.11.2018.
15. Зайцева, Е.В. Основы электронного бизнеса [Электронный ресурс]: учебное пособие для специальности 080503 / Е.В. Зайцева. – Томск: кафедра ТУ, ТУСУР, 2012. – 263 с. Режим доступа: tu.tusur.ru/upload/posobia/z1.doc. Дата обращения: 15.12.2018.
16. Платёжная система «МИР». Безопасность карт «МИР» [Электронный ресурс] // Режим доступа: https://mironline.ru/about-card/security/. Дата обращения: 18.12.2018.
17. Российский новый университет. Администрирование в информационных системах (лекция 9) [Электронный ресурс] // Режим доступа: rdv.rosnou.ru/IT433/it_ad_10.pdf. Дата обращения: 5.09.2018.
18. Национальный открытый университет ИНТУИТ. Системы управления контентом [Электронный ресурс] // Режим доступа: www.intuit.ru/studies/courses/1036/239/lecture/6178. Дата обращения: 20.12.18.
19. IXBT.COM. Виртуализация: новый подход к построению IT-инфраструктуры [Электронный ресурс] // Режим доступа: https://www.ixbt.com/cm/virtualization.shtml. Дата обращения: 25.12.2018.

Учебное издание

Бобынцев Денис Олегович
Марухленко Анатолий Леонидович
Марухленко Леонид Олегович
Кужелева Светлана Анатольевна
Лисицын Леонид Александрович

Основы администрирования информационных систем

Учебное пособие

Ответственный редактор *С. Краснова*
Ответственный верстальщик *Д. Ананьева*

Издательство «Директ-Медиа»
117342, Москва, ул. Обручева, 34/63, стр. 1
Тел/факс + 7 (495) 334–72–11
E-mail: manager@directmedia.ru
www.biblioclub.ru