**1. Системы обработки данных: основные определения, классификация, характеристики, предъявляемые требования.**

Системы обработки данных - СОД предназначены для учета и оперативного регулирования хозяйственных операций, подготовки стандартных документов для внешней среды (счетов, расчета заработной платы и т.п.). Такие системы наряду с функциями ввода, выборки, коррекции информации выполняют математические расчеты без применения методов оптимизации.

## Типы СОД

**1. Одномашинные СОД**

Одномашинные СОД построены на базе единственной ЭВМ с однопроцессорной структурой. Создание их, включая разработку ПО не вызывает трудностей, однако имеет ограничения по производительности и допускает простой системы в течение нескольких часов, из-за отказа оборудования.

**2. Вычислительные комплексы (ВК).**

Для увеличения производительности СОД несколько ЭВМ (CPU) связываются между собой, образуя многомашинный вычислительный комплекс (ММВК). ММВК различают с косвенной и прямой связью между ЭВМ. Связь между ЭВМ осуществляется через общее запоминающее устройство или через адаптер, обеспечивающий обмен данными между каналами ввода/вывода двух ЭВМ.

**3. Системы телеобработки (СТ)**

Системы, предназначенные для обработки данных, передаваемых по каналам связи, называют системами телеобработки. Пользователи взаимодействуют с системой посредством терминалов, подключаемых через каналы связи к средствам обработки данных ЭВМ. Данные передаются по каналам связи в форме сообщений, несущих в себе, кроме собственно данных, служебную информацию, необходимую для управления процессами передачи и защиту данных от искажений.

**4. Вычислительные сети**

Вычислительная сеть объединяет территориально разрозненный СОД в единый комплекс с помощью каналов передачи данных с единым ПО.

Узлы связи (УС) принимают данные и передают. Ядром сети является базовая сеть передачи данных (СПД), которая включает УС и каналы связи. К ЭВМ подключены терминалы, через которые пользователи взаимодействуют с сетью. Совокупность терминалов и средств связи для подключения их к ЭВМ образуют терминальную сеть. Таким образом, вычислительная сеть представляет собой композицию СПД, сеть ЭВМ и терминальную сеть.

**5. Локальная вычислительная сеть (ЛВС).**

ЛВС — это совокупность близко расположенных ЭВМ. ЭВМ связываются с помощью моноканала, единого для всех ЭВМ сети. В моноканале наиболее часто используется витая пара, коаксиальный кабель или волоконно-оптический кабель. Сетевые адаптеры (СА) — контроллеры, реализующие операции ввода-вывода через моноканал.

## Классификация СОД

СОД, построенные на основе отдельных ЭВМ, вычислительных комплексов и систем, образуют класс сосредоточенных (**централизованных**) систем, в которых вся обработка реализуется ЭВМ, вычислительным комплексом или специальный системой. Системы телеобработки и вычислительные сети относятся к классу **распределенных** систем, в которых процессы обработки данных рассредоточены по многим компонентам.

**2. Вычислительные сети: виды, свойства, характеристика.**

**Компьютерная (вычислительная) сеть** — это совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

**Распределенная обработка данных** — это обработка данных, выполняемая на независимых, но связанных между собой компьютерах.

В зависимости **от территориального расположения** абонентов **компьютерные сети делятся на**:

1. глобальные — вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах.

2. региональные — вычислительная сеть связывает абонентов, расположенных в большом городе, регионе, отдельной стране;

3. локальные — вычислительная сеть объединяет абонентов, расположенных в пределах небольшой территории. К классу локальных сетей относятся сети отдельных предприятий, фирм, офисов и т. д.

**По скорости** передачи информации компьютерные сети делятся на низко-, средне- и высокоскоростные.

**По типу среды передачи** разделяются на сети коаксиальные, на витой паре, оптоволоконные, с передачей информации по радиоканалам.

В общем случае **компьютерная сеть представляется** совокупностью сети рабочих станций, сети серверов и базовой сети передачи данных.

**Рабочая станция** (клиентская машина, терминал) — это компьютер, за которым непосредственно работает абонент компьютерной сети. Сеть рабочих станций представлена совокупностью рабочих станций и средств связи, обеспечивающих взаимодействие рабочих станций с сервером и между собой.

**Сервер** — это компьютер, выполняющий общие задачи компьютерной сети и предоставляющий услуги рабочим станциям. Сеть серверов — это совокупность серверов и средств связи, обеспечивающих подключение серверов к базовой сети передачи данных.

**Базовая сеть передачи данных** — это совокупность средств передачи данных между серверами. Она состоит из каналов связи и узлов связи. Узел связи — это совокупность средств коммутации и передачи данных в одном пункте. Узел связи принимает данные, поступающие по каналам связи, и передает данные в каналы, ведущие к абонентам.

**Характеристики компьютерной сети:**

- операционные возможности — перечень основных действий по обработке данных.

- производительность — представляет собой суммарную производительность компьютеров, участвующих в решении задачи пользователя;

- время доставки сообщений — определяется как статистическое среднее время от момента передачи сообщения в сеть до момента получения сообщения адресатом;

- стоимость предоставляемых услуг.

[к 5-му вопросу](#вопрос5)

**3. Сети с коммутацией каналов; с коммутацией сообщений; с коммутацией пакетов.**

**Коммутация каналов** (используется при передаче аудиоинформации по обычным телефонным линиям связи) – процесс, который по запросу осуществляет соединение двух или более станций данных и обеспечивает монопольное использование канала передачи данных до тех пор, пока не произойдет разъединение. Коммутация каналов подразумевает образование непрерывного составного физического канала из последовательно соединенных отдельных канальных участков для прямой передачи данных между узлами. Отдельные каналы соединяются между собой специальной аппаратурой – коммутаторами, которые могут устанавливать связи между любыми конечными узлами сети.

Достоинством сетей коммутации каналов является простота реализации (образование непрерывного составного физического канала), а недостатком - низкий коэффициент использования каналов, высокая стоимость передачи данных, повышенное время ожидания других пользователей.

**Коммутация сообщений** (применяется в основном для передачи электронной почты, в телеконференциях, электронных новостях) – процесс пересылки данных, включающий прием, хранение, выбор исходного направления и дальнейшую передачу сообщений без нарушения их целостности. Используются в тех случаях, когда не ожидается немедленной реакции на сообщение. Для передачи звука традиционно используется телефон, изображений – телевидение, текста – телеграф, данных – вычислительные сети.

К достоинствам можно отнести - уменьшение стоимости передачи данных. Недостатком данного способа является низкая скорость передачи информации, невозможность ведения диалога между пользователями.

**Коммутация пакетов** (для передачи данных, в настоящее время используется также для передачи аудио - и видеоинформации) – это коммутация сообщений, представляемых в виде адресуемых пакетов, когда канал передачи данных занят только во время передачи пакета и по ее завершению освобождается для передачи других пакетов. Коммутаторы сети, в роли которых выступают шлюзы и маршрутизаторы, принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге станции назначения.

Пакетная коммутация подразумевает обмен небольшими пакетами (часть сообщения) фиксированной структуры, которые не дают возможности образования очередей в узлах коммутации. Достоинства: быстрое соединение, надежность, эффективность использования сети.

**4. Модель взаимодействия открытых систем OSI.**

Необходимость стандартизации протоколов важна для понимания сетями друг друга при их взаимодействии. **Протокол**— это набор правил и методов взаимодействия объектов вычислитель­ной сети.  Была разработана система стандартных протоколов, получившая название **модели взаимодействия открытых систем**(**OSI),**часто называемая также эталонной семиуровневой логической моделью открытых систем.   
Открытая система — система, доступная для взаимодействия с другими система­ми в соответствии с принятыми стандартами.   
В общем случае сеть должна иметь 7 функциональных уровней.

|  |  |  |
| --- | --- | --- |
| **Уровень OSI** | **Назначение** | **Примеры протоколов** |
| 7 Прикладной | Обеспечивает прикладным процессам пользователя средства доступа к сетевым ресурсам; является интерфейсом между программами пользователя и сетью. Управляет запуском программ пользователя, их выполнением, вводом-выводом данных. | HTTP, SMTP, FTP, DNS, Telnet и т. д. |
| 6 Представления | Интерпретация и преобразование пере­даваемых в сети данных к виду, удобному для прикладных процессов. Имеет интерфейс с прикладными программами. | X.226 |
| 5 Сеансовый | Организация и проведение сеансов связи между прикладными процессами (инициализация и поддержание сеанса между абонен­тами сети, управление очередностью и режимами передачи данных). | X.225, RPC, NetBEUI и т. д. |
| 4 Транспортный | Обеспечивает надежную и экономичную передачу данных между взаимодействующими объектами сеансового уровня. Большое внимание на этом уровне уделено контролю достоверности передаваемой информации. | TCP, UDP и т. д. |
| 3 Сетевой | Обеспечивает маршрутизацию передачи данных в сети, устанавливает логический канал между объектами для реализации протоколов транспортного уровня. На этом уровне выполняется структуризация данных — разбивка их на пакеты и присвое­ние пакетам сетевых адресов. | IP, IPX и т. д. |
| 2 Канальный | Формирование и управление физическим ка­налом передачи данных между объектами сетевого уровня (установление, поддер­жание и разъединение логических каналов), обеспечение контроля и исправления ошибок передачи. | LAP-B, HDLC, SNAP, SDLC, IEEE 802.2 и т.д. |
| 1 Физический | Формирует физическую среду передачи данных, устанавливает соединения объектов сети с этой средой. Управление выполняется на уров­не битов цифровых (импульсы, их амплитуда, форма) и аналоговых (амплитуда, частота, фаза непрерывного сигнала). | Ethernet, Token Ring |

**Информация, передаваемая между уровнями**, имеет стандартный формат: заголовок (header), служебная информация, данные, концевик. Каждый уровень при передаче блока информации нижестоящему уровню снабжает его своим заго­ловком. Заголовок вышестоящего уровня воспринимается нижестоящим как пе­редаваемые данные.   
  
Указанные уровни управления можно по разным признакам объединять в **группы**:   
-  уровни 1, 2 и частично 3 реализуются в большей части за счет аппаратных средств; верхние уровни с 4 по 7 и частично 3 обеспечиваются программными средствами;

- уровни 1 и 2 ответственны за физические соединения; уровни 3-6 заняты орга­низацией передачи, передачей и преобразованием информации в понятную для абонентской аппаратуры форму; уровень 7 обеспечивает выполнение приклад­ных программ пользователя.

**5. Структура и основные элементы вычислительной сети. Основные устройства базовой сети передачи данных.**

**(еще можно смотреть этот вопрос:** [**2. Вычислительные сети: виды, свойства, характеристика**](#вопрос2)**)**

С технической точки зрения **сеть** – это система для передачи информации, состоящая из множества терминалов и коммуникационной среды.

**Терминалы** (рабочие станции) – это конечные устройства, которые являются источником или получателем информации (от другого терминала).

**Коммуникационная среда** служит для передачи информации между терминалами. Состоит из каналов связи и узлов.

**Канал связи** – совокупность устройств, осуществляющих передачу информации на большие расстояния.

**Узлы** – промежуточные устройства, в которых сходятся каналы (как минимум три) сети передачи информации. Они играют роль диспетчеров, позволяя более полно и оптимально использовать каналы. Узлы коммутируют каналы, сообщения, временно хранят сообщения.

Узел, который предоставляет какой-либо ресурс другим узлам, называют **сервером**  
данного ресурса (файл-сервер, принт-сервер и т.д.). Компьютер, использующий ресурсы сервера, называется **клиентом**.

Наличие серверов позволяет экономить дисковое пространство для хранения информации и время его обработки. Например, достаточно установить один экземпляр программного пакета на файл-сервер вместо того, чтобы ставить его на нескольких рабочих станциях.

Все **сети делятся на две группы**: с выделенным сервером или без него. В сети без выделенного файл-сервера каждый компьютер может выступать в роли и клиента и сервера, а в сети с выделенным файл-сервером один компьютер полностью выделяется для хранения и пересылки данных и не может быть использован как рабочая станция.

В разных сетях проблема согласованного взаимодействия решалась в рамках так называемых протоколов. **Протокол** – это система соглашений, касающихся различных аспектов информационного обмена взаимодействующих объектов. Понятие протокола охватывает правила и процедуры, алгоритмы и требования, определяющие порядок взаимодействия на всех уровнях компьютерных сетей.

Связь сетей с различными протоколами и системами протоколов осуществляется узловыми станциями, так называемыми **шлюзам**и. Их задача – осуществление всех необходимых преобразований для обеспечения межсетевого взаимодействия. Обычно он реализуется на базе некоторого узлового компьютера.

**6. Топологии сетей. Краткая сравнительная характеристика.**

Термин «топология сети» характеризует физическое расположение компьютеров, кабелей и других компонентов сети. Каждая топология может диктовать не только тип кабеля, но и способ его прокладки. Топология может также определять способ взаимодействия компьютеров в сети.

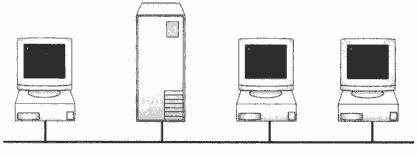
Все сети строятся на основе трех базовых топологий:

* шина (bus);
* звезда (star);
* кольцо (ring).

Если компьютеры подключены вдоль одного кабеля [сегмента], топология называется шиной. В том случае, когда компьютеры подключены к сегментам кабеля, исходящим из одной точки, или концентратора, топология называется звездой. Если кабель, к которому подключены компьютеры, замкнут в кольцо, такая топология носит название кольца.

**Шина**

В топологии «шина» используется один кабель, именуемый магистралью или сегментом, вдоль которого подключены все компьютеры сети.

****

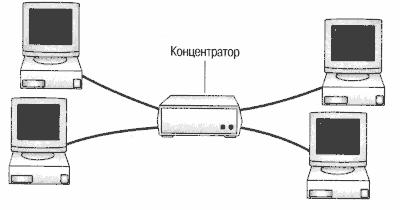
В сети с топологией «шина» компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов.

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, тем медленнее сеть.

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.

**Звезда**

При топологии «звезда» все компьютеры с помощью сегментов кабеля подключаются к центральному компоненту, именуемому концентратором (hub). Сигналы от передающего компьютера поступают через концентратор ко всем остальным.

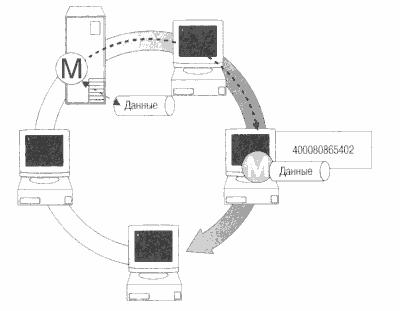


В сетях с топологией «звезда» подключение кабеля и управление конфигурацией сети централизованны. Но есть и недостаток: так как все компьютеры подключены к центральной точке, для больших сетей значительно увеличивается расход кабеля. К тому же, если центральный компонент выйдет из строя, нарушится работа всей сети. А если выйдет из строя только один компьютер (или кабель, соединяющий его с концентратором), то лишь этот компьютер не сможет передавать или принимать данные по сети. На остальные компьютеры в сети это не повлияет.

**Кольцо**

При топологии «кольцо» компьютеры подключаются к кабелю, замкнутому в кольцо. 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, усиливая сигналы и передавая их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть.

Один из принципов передачи данных в кольцевой сети носит название передачи маркера. Маркер последовательно, от одного компьютера к другому, передается до тех пор, пока его не получит тот, который «хочет» передать данные. Передающий компьютер изменяет маркер, помещает электронный адрес в данные и посылает их по кольцу.



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.

**7. Локальные и глобальные вычислительные сети.**

**Компьютерная сеть** – это совокупность компьютеров и различных устройств, обеспечивающих информационный обмен между компьютерами в сети без использования каких-либо промежуточных носителей информации.

По территориальной распространенности сети могут быть локальными, глобальными, и региональными.

**Локальные компьютерные сети.**

Локальная сеть объединяет компьютеры, установленные в одном помещении или в одном здании. В небольших локальных сетях все компьютеры обычно равноправны, т. е. пользователи самостоятельно решают, какие ресурсы своего компьютера (диски, каталоги, файлы) сделать общедоступными по сети. Такие сети называются **одноранговыми**.

Если к локальной сети подключено более десяти компьютеров, то одноранговая сеть может оказаться недостаточно производительной. Для увеличения производительности, а также в целях обеспечения большей надежности при хранении информации в сети некоторые компьютеры специально выделяются для хранения файлов или программ-приложений. Такие компьютеры называются серверами, а локальная сеть — **сетью на основе серверов.**Каждый компьютер, подключенный к локальной сети, должен иметь специальную плату (сетевой адаптер). Между собой компьютеры (сетевые адаптеры) соединяются с помощью кабелей.

Общая схема соединения компьютеров в локальные сети называется топологией сети. Сети могут иметь топологию «шина» и «звезда». В первом случае все компьютеры подключены к одному общему кабелю (шине), во втором - имеется специальное центральное устройство (хаб), от которого идут «лучи» к каждому компьютеру, т.е. каждый компьютер подключен к своему кабелю.

Структура типа «шина» проще и экономичнее, так как для нее не требуется дополнительное устройство и расходуется меньше кабеля. Но она очень чувствительна к неисправностям кабельной системы. Если кабель поврежден хотя бы в одном месте, то возникают проблемы для всей сети. Место неисправности трудно обнаружить.  
В этом смысле «звезда» более устойчива. Поврежденный кабель – проблема для одного конкретного компьютера, на работе сети в целом это не сказывается.

В сети, имеющей структуру типа «кольцо» информация передается между станциями по кольцу с переприемом в каждом сетевом контроллере. Переприем производится через буферные накопители, выполненные на базе оперативных запоминающих устройств, поэтому при выходе их строя одного сетевого контроллера может нарушиться работа всего кольца. Достоинство кольцевой структуры – простота реализации устройств, а недостаток – низкая надежность.

**Региональные компьютерные сети.**

Локальные сети не позволяют обеспечить совместный доступ к информации пользователям, находящимся, например, в различных частях города. На помощь приходят региональные сети, объединяющие компьютеры в пределах одного региона (города, страны, континента).

**Глобальная компьютерная сеть Интернет.**

Интернет — это глобальная компьютерная сеть, объединяющая многие локальные, региональные и корпоративные сети и включающая в себя десятки миллионов компьютеров.

В каждой локальной или корпоративной сети обычно имеется, по крайней мере, один компьютер, который имеет постоянное подключение к Интернету. Основу Интернета составляют более ста миллионов серверов, постоянно подключенных к сети. К серверам Интернета могут подключаться с помощью локальных сетей или коммутируемых телефонных линий сотни миллионов пользователей сети.

Для того, чтобы связаться с некоторым компьютером в сети Интернет, Вам надо знать его уникальный Интернет - адрес. Существуют два равноценных формата адресов, которые различаются лишь по своей форме: IP - адрес и DNS - адрес.

**IP - адрес** состоит из четырех блоков цифр, разделенных точками. Он может иметь такой вид: 84.42.63.1. Каждый блок может содержать число от 0 до 255. С понятием IP - адреса тесно связано понятие "хост". Под хостом понимается любое устройство, использующее протокол TCP/IP для общения с другим оборудованием. Это может быть не только компьютер, но и маршрутизатор, концентратор и т.п.

IP - адрес имеет числовой вид, так как его используют в своей работе компьютеры. Но он весьма сложен для запоминания, поэтому была разработана доменная система имен: **DNS**. DNS - адрес включает более удобные для пользователя буквенные сокращения, которые также разделяются точками на отдельные информационные блоки (домены). Например:  
[www.klyaksa.net](http://www.klyaksa.net/)

Если Вы вводите DNS - адрес, то он сначала направляется в так называемый сервер имен, который преобразует его в 32 - битный IP - адрес для машинного считывания.

Доменная система имен имеет иерархическую структуру: домены верхнего уровня - домены второго уровня и так далее. Домены верхнего уровня бывают двух типов: географические (двухбуквенные - каждой стране свой код) и административные (трехбуквенные).

Портал Клякс@.net зарегистрировал домен второго уровня klyaksa в административном домене верхнего уровня net.

**URL** (Uniform Resource Locator, унифицированный определитель ресурсов) - это адрес некоторой информации в Интернет. Он имеет следующий формат: тип ресурса://адрес узла/прочая информация.

Наиболее распространенными считаются следующие типы ресурсов: ftp:// ftp – сервер,   
http:// адрес в WWW, mailto:// адрес электронной почты   
  
Ресурсная часть URL всегда заканчивается двоеточием и двумя или тремя наклонными чертами. Далее следует конкретный адрес узла, который Вы хотите посетить. За ним в качестве ограничителя моет стоять наклонная черта.

Итак, в Интернет возможны следующие виды адресов:

|  |  |
| --- | --- |
| **Адрес** | формат |
| **IP** | 12.105.58.9 |
| **DNS** | компьютер.сеть.домен |
| **URL** | тип ресурса://DNS - адрес |

**8. Физические среды передачи данных. Сравнительная характеристика.**

**Линия связи** состоит в общем случае из физической среды, по которой передаются информационные сигналы, аппаратуры передачи данных и промежуточной аппаратуры. Синонимом термина линия связи является термин канал связи.

**Физическая среда** передачи данных может представлять собой кабель, то есть набор проводов, изоляционных и защитных оболочек, соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются информационные сигналы.

В зависимости от среды передачи данных линии **связи разделяются на**:

□ проводные (воздушные);

□ кабельные (медные и волоконно-оптические);

□ радиоканалы наземной и спутниковой связи.

**Проводные (воздушные) линии** связи представляют собой провода, проложенные между столбами и висящие в воздухе. По таким линиям связи традиционно передаются телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

**Кабельные линии**. Кабель состоит из проводников, заключенных в несколько слоев изоляции. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных сетях применяются **три основных типа кабеля**: кабели на основе скрученных пар медных проводов, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели (первые два типа кабелей называют также медными кабелями).

**Скрученная пара проводов называется витой парой**. Скручивание проводов снижает влияние внешних и взаимных помех на полезные сигналы, передаваемые по кабелю.

**Коаксиальный кабель** состоит из несимметричных пар проводников. Каждая пара представляет собой внутреннюю медную жилу и соосную с ней внешнюю жилу, отделенной от внутренней жилы диэлектрической изоляцией. Внешняя жила играет двоякую роль — по ней передаются информационные сигналы, также она является экраном, защищающим внутреннюю жилу от внешних электромагнитных полей.

**Волоконно-оптический кабель** состоит из тонких гибких стеклянных волокон, по которым распространяются световые сигналы. Это наиболее качественный тип кабеля — он обеспечивает передачу данных с очень высокой скоростью и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех.

У волоконно-оптических кабелей есть один серьезный недостаток — сложность соединения волокон с разъемами и между собой при необходимости увеличения длины кабеля. Сама стоимость волоконно-оптических кабелей ненамного превышает стоимость кабелей на витой паре, однако проведение монтажных работ с оптоволокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования.

**Радиоканалы наземной и спутниковой связи** образуются с помощью передатчика и приемника радиоволн. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. Диапазоны коротких, средних и длинных волн (КБ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (Amplitude Modulation, AM) по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для которых характерна частотная модуляция (Frequency Modulation, FM), а также диапазонах сверхвысоких частот (СВЧ, или microwaves).

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных, но наиболее перспективными являются волоконно-оптические кабели. Популярной средой является также витая пара, которая характеризуется отличным отношением качества к стоимости, а также простотой монтажа. Спутниковые каналы и радиосвязь используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя — например, при прохождении канала через малонаселенную местность.

**9. Понятие информации. Энтропия и её свойства. Установление количественной меры информации в общем случае.**

Термин ***информация*** происходит от латинского слова ***informatio*** – разъяснение, изложение. Первоначальное значение этого термина – «сведения, передаваемые людьми устным, письменным или иным способом».

В информатике под информацией понимается сообщение, снижающее степень неопределенности знаний о состоянии предметов или явлений и помогающее решить поставленную задачу. Изменение некоторой физической величины во времени, обеспечивающее передачу сообщений, называется ***сигналом***.

Для того чтобы в материальном мире происходил обмен информацией, ее преобразование и передача, должны существовать носитель информации, передатчик, канал связи, приемник и получатель информации. Канал связи представляет собой среду, в которой происходит передача информации. Канал связи объединяет источник и получателя информации в единую ***информационную систему.***

Зарегистрированные сигналы называются ***данными***. Для их регистрации с целью хранения и передачи необходим некоторый язык. Этот язык должен быть понятен как отправителю информации, так и ее получателю. Для получения информации нужен ***метод обработки данных***. Информация – это продукт взаимодействия данных и адекватных им методов их обработки.

Информация существует только в момент взаимодействия данных и методов. В остальное время она содержится в виде данных. Из одних и тех же данных разные получатели могут извлечь разную информацию в зависимости от адекватности методов их обработки.

**Энтропия** — это количество информации, приходящейся на одно элементарное сообщение источника, вырабатывающего статистически независимые сообщения.

**Информацио́нная энтропи́я** — мера неопределённости или непредсказуемости [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), неопределённость появления какого-либо символа [первичного алфавита](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B2%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82). При отсутствии информационных потерь численно равна количеству информации на символ передаваемого сообщения.

Например, в последовательности букв, составляющих какое-либо предложение на русском языке, разные буквы появляются с разной частотой, поэтому неопределённость появления для некоторых букв меньше, чем для других. Если же учесть, что некоторые сочетания букв встречаются очень редко, то неопределённость уменьшается еще сильнее.

## Свойства

Энтропия является количеством, определённым в контексте вероятностной модели для [источника данных](https://ru.wikipedia.org/w/index.php?title=%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1).

Например, кидание монеты имеет энтропию: 1 [бит](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) на одно кидание (при условии его независимости), а количество **возможных состояний** равно:2^1=2 **возможных состояния** (значения) ("орёл" и "[решка](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%BA%D0%B0)").

У источника, который генерирует строку, состоящую только из букв «А», энтропия равна нулю, а количество **возможных состояний** равно:2^0=1 **возможное состояние** (значение) («А») и от основания логарифма не зависит.

Энтропия не может быть отрицательной

**Количественная мера информации**

Для того чтобы сообщение могло быть передано от источника к потребителю, оно должно быть каким-либо образом отображено, зафиксировано. Такое отображение осуществляется с помощью некоторых символов (знаков). Конечный упорядоченный набор знаков, используемых для передачи сообщений, называют *алфавитом.* Последовательность символов алфавита — *словом.* Сообщение фиксируется как последовательность слов. В канале связи способ представления сообщения может изменяться: один алфавит заменяется другим. Правило отображения одного алфавита в другой называется *кодом,* а сама процедура—*перекодировкой* сообщения.

Поэтому имеет смысл говорить о количестве информации, связанной с появлением каждого символа, и о количестве информации, соответствующей сообщению в целом.

В технических устройствах хранения, передачи и обработки информации для ее кодирования часто используются алфавиты, содержащие лишь два различных знака. Алфавит из двух символов называют двоичным и говорят о двоичном представлении информации. При таком представлении буквы, цифры и любые другие символы изображаются двоичными словами — последовательностями из нулей и единиц. Если считать, что с нулем и единицей в двоичном алфавите связаны одинаковые вероятности их появления р1 =Р2==1/2, то количество информации на один символ при двоичном кодировании будет равно H=log2 2=1 бит.

Таким образом, количество информации (в битах), заключенной в двоичном слове, равно длине слова (числу двоичных знаков в нем). Полезно отметить, что само слово "бит" происходит от сокращения английских слов binary digit*—* двоичный знак. Для удобства использования введены более крупные, чем бит, единицы количества информации. Восьмиразрядное двоичное слово (и количество информации, содержащейся в нем) называют *байт.* Объем информации в 1024 байта носит название *килобайт* и обозначается Кбайт. Используются и еще более крупные единицы измерения: *мегабайт* (1024 Кбайта) обозначается Мбайт и *гигабайт* —Гбайт (1024 Мбайта).

Для каждого действующего канала связи существенной характеристикой является его *пропускная способность,* т. е. максимальное количество информации, которое может быть передано по этому каналу в единицу времени.

**10. Избыточность и поток информации источника сообщений.**

Как известно, энтропия характеризует среднее количество информации, несомое одним символом источника. Она максимальна, когда символы вырабатываются источником с равной вероятностью. Если же некоторые символы появляются чаще других, энтропия уменьшается. Чем меньше энтропия источника отличается от максимальной, тем рациональнее он работает, тем большее количество информации несут его символы.

Для сравнения источников по их информативности введём параметр, называемый избыточностью, равный



Источник, избыточность которого R = 0, называется оптимальным. Если R = 1, то Н(Х) = 0, и, следовательно, информация, вырабатываемая источником, равна нулю.

Следует, однако, иметь в виду, что не всегда нужно стремиться к тому, чтобы R = 0. Некоторая избыточность бывает полезной для обеспечения надежности передачи сообщений. Простейшим видом введения избыточности для борьбы с шумами является многократная передача одного и того же символа.

**Поток информации источника сообщений**

При работе источника сообщений на его выходе отдельные символы появляются через некоторые интервалы времени; в этом смысле мы можем говорить о длительности отдельных символов, и, следовательно, может быть поставлен вопрос о количестве информации, вырабатываемой источником в единицу времени.

Длительность выдачи знаков источником в каждом состоянии в общем случае может быть различной. Тогда средняя длительность выдачи источником одного знака:



где P(Sk) – вероятность того, что источник сообщений находится в состоянии

Sk; P(xi) – вероятность появления символа xi в состоянии Sk; txi – длительность

выдачи знака xi источником в состоянии Sk.

Энтропия источника, приходящаяся на единицу времени, может быть названа скоростью создания сообщений, или потоком информации, т.е.

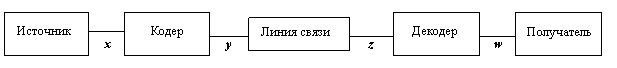


Для увеличения потока информации необходимо по возможности уменьшить среднюю длительность символов t. С этой целью, например, необходимо, чтобы длительность тех символов, вероятность появления которых больше, была меньше, чем для символов, вероятность появления которых относительно велика. Таким образом, для получения большого потока информации на выходе источника необходимо не только обеспечить по возможности большую энтропию на символ, но и правильно выбрать длительность разных символов.

**11. Модель информационной системы передачи дискретных сообщений в отсутствии шумов.**

Если в дискретном канале отсутствуют шумы, то, как бы ни кодировались сигналы, потерь информации не будет. Однако, в большинстве случаев, информацию нужно представлять не только безошибочно, но и экономно.

Упрощенная блок-схема дискретного канала связи без шумов приведена на рисунке



На вход такого канала подаются дискретные сообщения (х). Для передачи на расстояние воздействия любого источника сообщения, как правило, преобразуются в первичные электрические сигналы. Последние с помощью кодирующего устройства преобразуются в кодированные сигналы.

Назначение кодирующего устройства (кодера) состоит в том, чтобы представить выходное сообщение источника в некоторой стандартной форме, например в виде последовательности двоичных сигналов. Основная задача кодирования заключается в том, чтобы стандартное представление было наиболее экономным, т.е. требовало в среднем наименьшего возможного числа двоичных сигналов.

Кодированные сигналы передаются по каналу связи и на его приемном конце восстанавливаются в точно такой же или в иной форме (z). Декодирующее устройство преобразует кодированные сигналы (z) в сообщения (w) в форме, наиболее удобной для данного получателя.

При отсутствии шумов (помех) можно принять z =у. Способ кодирования должен быть таким, чтобы, в случае отсутствия шумов на приемной стороне, по полученным кодированным сигналам можно было однозначно восстановить вид переданных сообщений, т. е. чтобы w =x. Это накладывает некоторые ограничения на допустимые комбинации символов кода. Так, например, если мы закодируем символы х1 → 1,х2 → 2 и х3 → 01, то при получении сигнала 01 на приемной стороне мы не будем знать, что было передано: сообщение х3 или два сообщениях 2х1. Следовательно, необходимо либо отказаться от такого способа кодирования, либо заранее условиться, что комбинацию х2х1 передавать нельзя (она запрещена). На возможную последовательность символов кода могут накладываться и другие ограничения.

Совокупность запретов, обусловленных принятым способом кодирования и построением аппаратуры, будем относить к фиксированным ограничениям, накладываемым на информационный канал.

**12. Пропускная способность канала связи**

Пропускная способность (количество бит информации, передаваемое в единицу времени) и достоверность передачи данных (вероятность доставки неискаженного бита или же вероятность искажения бита) зависят, с одной стороны, от характеристик физической среды, а с другой — определяются характеристиками способа передачи данных. Следовательно, нельзя говорить о пропускной способности линии связи, до того как для нее определен протокол физического уровня. Например, поскольку для цифровых линий всегда определен протокол физического уровня, задающий битовую скорость передачи данных, то для них всегда известна и пропускная способность.

Пропускная способность (throughput) линии характеризует максимально возможную скорость передачи данных по линии связи. Пропускная способность измеряется в битах в секунду (бит/с), а также в производных единицах, таких как килобит в секунду (кбит/с), мегабит в секунду (Мбит/с), гигабит в секунду (Гбит/с) и т. д.

Пропускная способность линий связи и коммуникационного сетевого оборудования традиционно измеряется в битах в секунду, а не в байтах в секунду. Это связано с тем, что данные в сетях передаются последовательно, то есть побитно, а не параллельно, байтами, как это происходит между устройствами внутри компьютера. Такие единицы измерения, как килобит, мегабит или гигабит, в сетевых технологиях строго соответствуют степеням 10 (то есть килобит — это 1000 бит, а мегабит — это 1000000 бит), как это принято по всех отраслях науки и техники, а не близким к этим числам степеням 2, как это принято в программировании, где приставка «кило»- равна = 1024.

Выбор способа представления дискретной информации в виде сигналов, подаваемых на линию связи, называется **физическим кодированием**. От выбранного способа кодирования зависит спектр сигналов и, соответственно, пропускная способность линии.

Большинство способов кодирования используют изменение какого-либо параметра периодического сигнала — частоты, амплитуды и фазы синусоиды. Периодический сигнал, параметры которого изменяются, называют несущим сигналом или несущей частотой, если в качестве такого сигнала используется синусоида.

Если сигнал изменяется так, что можно различить только два его состояния, то любое его изменение будет соответствовать наименьшей единице информации — биту. Если же сигнал может иметь более двух различимых состояний, то любое его изменение будет нести несколько битов информации.

На пропускную способность линии оказывает влияние не только физическое, но и **логическое кодирование**. Логическое кодирование выполняется до физического кодирования и подразумевает замену битов исходной информации новой последовательностью битов, несущей ту же информацию, но обладающей, кроме этого, дополнительными свойствами, например возможностью для приемной стороны обнаруживать ошибки в принятых данных. Сопровождение каждого байта исходной информации одним битом четности — это пример очень часто применяемого способа логического кодирования при передаче данных с помощью модемов. Другим примером логического кодирования может служить шифрование данных, обеспечивающее их конфиденциальность при передаче через общественные каналы связи. При логическом кодировании чаще всего исходная последовательность бит заменяется более длинной последовательностью, поэтому пропускная способность канала по отношению к полезной информации при этом уменьшается.

**13. Экономное кодирование сообщений.**

В связи с непрерывным увеличением объема информации, накапливаемой во всех областях человеческой деятельности, все большее значение приобретают технологии ее компактного представления. Под компактным представлением информации следует понимать некоторую форму хранения информации, экономную с точки зрения занимаемого объема информационного носителя.

Способ получения компактного представления информации носит название экономное кодирование. Первый алгоритм экономного кодирования был предложен С. Морзе (имеется в виду знаменитая азбука Морзе).

**Методы** экономного кодирования можно разделять на методы кодирования информации **без потерь (lossless)** и **с потерями (lossy).**

Методы последней группы применяются к информации, содержащей отдельные несущественные составляющие (при определенных условиях эти составляющие могут быть частично или полностью удалены из информации). Одним из примеров такой информации является пренебрежение неразличимыми человеческим зрением особенностями пространственных и временных изменений изображений. В основе методов экономного кодирования информации с потерями лежит учет специфических особенностей восприятия информации.

Если методы экономного кодирования с потерями могут быть ориентированы на произвольный тип информации, методы экономного кодирования без потерь используются исключительно для обработки дискретной информации. Под дискретностью информации следует понимать возможность ее представления в виде совокупности конечного или счетного числа натуральных чисел.

Информация, обладающая последовательной структурой (тестовая информация), является одним из наиболее распространенных видов информации. Таким образом, проблема экономного кодирования дискретной информации с последовательной структурой в рамках теории информации фактически сводится к проблеме экономного кодирования натуральных последовательностей.

**14. Представление дискретной информации в каналах передачи данных**

**(цифровое кодирование).**

При цифровом кодировании дискретной информации применяют **потенциальные и импульсные коды.** В потенциальных кодах для представления логических единиц и пулей используется только значение потенциала сигнала.

Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо частью импульса — перепадом потенциала определенного направления.

**Требования к методам цифрового кодирования**

При использовании прямоугольных импульсов для передачи дискретной информации необходимо выбрать такой способ кодирования, который одновременно достигал бы нескольких целей:

□ имел при одной и той же битовой скорости наименьшую ширину спектра результирующего сигнала;

□ обеспечивал синхронизацию между передатчиком и приемником;

□ обладал способностью распознавать ошибки;

□ обладал низкой стоимостью реализации.

**Потенциальный код без возвращения к нулю**

Метод потенциального кодирования, называемый также кодированием без возвращения к нулю (Non Return to Zero, NRZ) имеет такое название потому, что при передаче последовательности единиц сигнал не возвращается к нулю в течение такта. Метод NRZ прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, поэтому приемник лишен возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные.

В результате в чистом виде код NRZ в сетях не используется. Тем не менее используются его различные модификации, в которых устраняют плохую самосинхронизацию кода NRZ.

**Биполярный импульсный код**

В биполярном импульсном коде единица представлена импульсом одной полярности, а ноль — другой. Каждый импульс длится половину такта. Такой код обладает отличными самосинхронизирующими свойствами. Кроме того, спектр у него ниже, чем у потенциальных кодов. Из-за слишком широкого спектра биполярный импульсный код используется редко.

**15. Локальные сети. Технологии Ethernet, Token Ring, FDDI.**

**(локальные сети смотреть тут:** [**7. Локальные и глобальные вычислительные сети**](#вопрос7)**)**

**Технология Ethernet**

**Ethernet**— семейство технологий [пакетной](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BA%D0%B5%D1%82_%28%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8%29) передачи данных для [компьютерных сетей](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C). Ethernet — это самый распространенный на сегодняшний день стандарт локальных сетей.

Стандарты Ethernet определяют проводные соединения и электрические сигналы на [физическом уровне](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%BB%D0%BE%D0%B9), формат [кадров](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80_%28%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8%29) и протоколы управления доступом к среде — на [канальном уровне](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C) [модели OSI](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).

Название «Ethernet» (буквально «эфирная сеть») отражает первоначальный принцип работы этой технологии: всё, передаваемое одним узлом, одновременно принимается всеми остальными (то есть имеется некое сходство с [радиовещанием](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE%D0%B2%D0%B5%D1%89%D0%B0%D0%BD%D0%B8%D0%B5)). В настоящее время практически всегда подключение происходит через [коммутаторы (switch)](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D1%82%D0%B0%D1%82%D0%BE%D1%80), так что кадры, отправляемые одним узлом, доходят лишь до адресата (исключение составляют передачи на [широковещательный адрес](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%80%D0%BE%D0%BA%D0%BE%D0%B2%D0%B5%D1%89%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81)) — это повышает скорость работы и безопасность сети.

Все виды стандартов Ethernet (в том числе Fast Ethernet и Gigabit Ethernet)

используют один и тот же метод разделения среды передачи данных — метод

CSMA/CD.

**Метод доступа CSMA/CD**

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (carrier-sense-multiply-access with collision detection, CSMA/ CD). Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей среде, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Простота схемы подключения — один из факторов, определивших успех стандарта Ethernet.

**МАС-адреса**

На МАС-уровне для идентификации сетевых интерфейсов узлов сети используются уникальные 6-байтовые адреса, называемые МАС-адресами. Обычно МАС-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например 11-A0-17-3D-BC-01. Каждый сетевой адаптер имеет, по крайней мере, один МАС-адрес.

Первый бит (младший) старшего байта адреса назначения является признаком того, является адрес индивидуальным или групповым. Если он ранен 0, то адрес является индивидуальным (unicast), а если 1, то это групповой адрес (multicast).

Второй бит старшего байта адреса определяет способ назначения адреса —

централизованный или локальный. Если этот бит равен 0 (что бывает почти

всегда в стандартной аппаратуре Ethernet), то адрес назначен централизованно.

Двадцать четыре бита, отводимые производителю для адресации интерфейсов

его продукции, позволяют выпустить 16 миллионов интерфейсов под одним

идентификатором организации. Уникальность централизованно распределяемых адресов распространяется на все основные технологии локальных сетей — Ethernet, Token Ring, FDDI и т. д.

**Этапы доступа к среде**

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier-sense, CS). Признаком незанятости среды является отсутствие на ней несущей частоты. Если среда свободна, то узел имеет право начать передачу кадра первым. Кадр данных всегда сопровождается преамбулой (preamble), которая состоит из 7 байтов и 8-го байта. Последний байт носит название ограничителя начала кадра — Start of Frame. Преамбула нужна для вхождения приемника в побитовый и побайтовый синхронизм с передатчиком.

Сети **Token Ring**, так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передастся с помощью кадра специального формата, называемого маркером, или токеном, (token).

Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры — посланный кадр всегда возвращается в станцию-отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций выполняет роль так называемого активного монитора. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением МАС-адреса. Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор.

В сети Token Ring кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станциями и может непосредственно обмениваться данными только с ними. Для обеспечения доступа станций к физической среде по кольцу циркулирует кадр специального формата и назначения — маркер. В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции — той, которая является предыдущей в кольце. Такая станция называется ближайшим активным соседом, расположенным выше по потоку (данных) — Nearest Active Upstream Neighbor, NAUN. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

Получив маркер, станция анализирует его и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право доступа к физической среде для передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника. Все станции кольца ретранслируют кадр побитно, как повторители. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер, давая другим станциям сети возможность передавать данные.

Для различных видов сообщений, передаваемым кадрам, могут назначаться различные приоритеты: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция. Маркер также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше

(или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

За наличие в сети маркера, причем единственной его копии, отвечает активный

монитор. Если активный монитор не получает маркер в течение длительного

времени, то он порождает новый маркер.

**Технология FDDI** (Fiber Distributed Data Interface — оптоволоконный интерфейс распределенных данных) — это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

**Основные характеристики технологии**

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец — это основной способ повышения отказоустойчивости в сети FDDI, и узлы должны быть подключены к обоим кольцам.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца. Вторичное кольцо (Secondary) в этом режиме не используется.

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным, вновь образуя единое кольцо. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному — в обратном

(изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимую реконфигурацию. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей.

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных, поэтому для нее определен специальный метод доступа. Этот метод очень близок к методу доступа сетей Token Ring и также называется методом маркерного (или токенного) кольца — token ring. Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время

зависит от загрузки кольца — при небольшой загрузке оно увеличивается, а при перегрузках может уменьшаться до нуля. Эти изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса — асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

Особенности метода доступа FDDI

Для передачи синхронных кадров станция всегда имеет право захватить маркер

при его поступлении. При этом время удержания маркера имеет заранее заданную фиксированную величину.

Если же станции кольца FDDI нужно передать асинхронный кадр, то для выяснения возможности захвата маркера при его очередном появлении станция должна измерить интервал времени, который прошел с момента предыдущего прихода маркера. Этот

интервал называется временем оборота маркера (Token Rotation Time, TRT).

Интервал TRT сравнивается с другой величиной — максимально допустимым

временем оборота маркера по кольцу Т\_0рr. Каждая станция может предложить свое значение Т\_0рr, в результате для кольца устанавливается минимальное из предложенных станциями времен.

**16. Управление доступом к каналу передачи данных в локальных сетях.**

В локальных сетях, использующих разделяемую среду передачи данных (например, локальные сети с топологией шина и физическая звезда), актуальным является доступ рабочих станций к этой среде, так как если два ПК начинают одновременно передавать данные, то в сети происходит столкновение.

Для того чтобы избежать этих столкновений необходим специальный механизм, способный решить эту проблему. **Шинный арбитраж** - это механизм призванный решить проблему столкновений. Он устанавливает правила, по которым рабочие станции определяют, когда среда свободна, и можно передавать данные.

Существуют два метода шинного арбитража в локальных сетях:  
• обнаружение столкновений  
• передача маркера

**Обнаружение столкновений**.  
Когда в локальных сетях работает метод обнаружения столкновений, компьютер сначала слушает, а потом передает. Если компьютер слышит, что передачу ведет кто-то другой, он должен подождать окончания передачи данных и затем предпринять повторную попытку.

Прослушивание канала до передачи называется “прослушивание несущей” (carrier sense), а прослушивание во время передачи — обнаружение столкновений (collision detection). Компьютер, поступающий таким образом, использует метод, называющийся “обнаружение столкновений с прослушиванием несущей”, сокращенно CSCD.

**Передача маркера в локальных сетях**

Для того чтобы передать данные, компьютер сначала должен получить разрешение. Это значит, он должен “поймать” циркулирующий в сети пакет данных специального вида, называемый маркером. Маркер перемещается по замкнутому кругу, минуя поочередно каждый сетевой компьютер.

Каждый раз, когда компьютер должен послать сообщение, он ловит и держит маркер у себя. Как только передача закончилась, он посылает новый маркер в путешествие дальше по сети. Такой подход дает гарантию, что любой компьютер рано или поздно получит право поймать и удерживать маркер до тех пор, пока его собственная передача не закончится.

**Методы обмена данными в локальных сетях**

Существует несколько групп методов доступа, основанных на временном разделении канала:  
• централизованные и децентрализованные  
• детерминированные и случайные

**Централизованный доступ** управляется из центра управления сетью, например от сервера. Децентрализованный метод доступа функционирует на основе протоколов без управляющих воздействий со стороны центра.

**Детерминированный доступ** обеспечивает каждой рабочей станции гарантированное время доступа (например, время доступа по расписанию) к среде передачи данных. Случайный доступ основан на равноправности всех станций сети и их возможности в любой момент обратиться к среде с целью передачи данных.  
Централизованный доступ к моноканалу

**Обмен данными в сети с топологией шина**. В этой топологии, возможно, такое же централизованное управление, как и в “звезде”. Один из узлов (центральный) посылает всем остальным запросы, выясняя, кто хочет передавать, и затем разрешает передачу тому из них, кто после окончания передачи сообщает об этом.  
  
**Обмен данными в сети с топологией кольцо** (децентрализованный детерминированный метод доступа)  
1. В данной сети применяется метод доступа “передача маркера”. Алгоритм передачи следующий:  
а) узел, желающий передать, ждет свободный маркер, получив который помечает его как занятый (изменяет соответствующие биты), добавляет к нему свой пакет и результат отправляет дальше в кольцо;  
б) каждый узел, получивший такой маркер, принимает его, проверяет, ему ли адресован пакет;  
в) если пакет адресован этому узлу, то узел устанавливает в маркере специально выделенный бит подтверждения и отправляет измененный маркер с пакетом дальше;  
г) передававший узел получает обратно свою посылку, прошедшую через все кольцо, освобождает маркер (помечает его как свободный) и снова посылает маркер в сеть. При этом передававший узел знает, была ли получена его посылка или нет.

Рассмотрим наиболее часто применяющийся метод множественного доступа с контролем несущей и обнаружением коллизий (столкновений) (**CSMA/CD**). Суть алгоритма в следующем:  
1) узел, желающий передавать информацию, следит за состоянием сети, и как только она освободится, то начинает передачу;  
2) узел передает данные и одновременно контролирует состояние сети (контролем несущей и обнаружением коллизий). Если столкновений не обнаружилось, передача доводится до конца;  
3) если столкновение обнаружено, то узел усиливает его (передает еще некоторое время) для гарантии обнаружения всеми передающими узлами, а затем прекращает передачу. Также поступают и другие передававшие узлы;  
4) после прекращения неудачной попытки узел выдерживает случайно выбираемый промежуток времени tзад, а затем повторяет свою попытку передать, при этом контролируя столкновения.

При повторном столкновении tзад увеличивается. В конечном счете, один из узлов опережает другие узлы и успешно передает данные. Метод CSMA/CD часто называют методом состязаний. Этот метод для сетей с шиной топологией реализуется протоколом Ethernet.

**17. Идентификация абонентов вычислительной сети: физические и логические адреса, имена, порты.**

**Физический адрес**

**MAC-адрес** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Media Access Control* — управление доступом к среде, также **Hardware Address**) — это уникальный идентификатор, присваиваемый каждой единице активного оборудования [компьютерных сетей](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C).

При проектировании стандарта [Ethernet](https://ru.wikipedia.org/wiki/Ethernet) было предусмотрено, что каждая сетевая карта (равно как и встроенный сетевой интерфейс) должна иметь уникальный шестибайтный номер (MAC-адрес), прошитый в ней при изготовлении. Этот номер используется для идентификации отправителя и получателя [фрейма](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80_%28%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8%29), и предполагается, что при появлении в сети нового компьютера (или другого устройства, способного работать в сети) сетевому администратору не придётся настраивать MAC-адрес.

Уникальность MAC-адресов достигается тем, что каждый производитель получает диапазон из шестнадцати миллионов (224) адресов, и по мере исчерпания выделенных адресов может запросить новый диапазон. Поэтому по трём старшим байтам MAC-адреса можно определить производителя. Существуют таблицы, позволяющие определить производителя по MAC-адресу.

В широковещательных сетях (таких, как сети на основе [Ethernet](https://ru.wikipedia.org/wiki/Ethernet)) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу. Таким образом, MAC-адреса формируют основу сетей на [канальном уровне](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C), которую используют протоколы более высокого ([сетевого](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, [ARP](https://ru.wikipedia.org/wiki/ARP) и [RARP](https://ru.wikipedia.org/wiki/RARP) в сетях [IPv4](https://ru.wikipedia.org/wiki/IPv4)).

**Структура MAC-адреса**

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Structura_MAC-%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%B0.gif)

48-разрядный MAC-адрес разделен на четыре части.

Первые 3 части содержат 24-битный код производителя. При этом используются только [младшие 22 разряда](https://ru.wikipedia.org/wiki/OUI#.D0.A0.D0.B0.D0.B7.D1.80.D1.8F.D0.B4.D0.BD.D0.BE.D1.81.D1.82.D1.8C_OUI) ([бита](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82)), 2 старшие имеют специальное назначение:

* первый бит (младший бит первого байта) указывает, для одиночного (0) или [группового](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%80%D0%BE%D0%BA%D0%BE%D0%B2%D0%B5%D1%89%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81) (1) адресата предназначен [кадр](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80)
* следующий бит указывает, является ли MAC-адрес глобально (0) или локально (1) администрируемым.

Последние 24 бита выбираются изготовителем для каждого экземпляра устройства.

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

[Администратор сети](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D1%81%D0%B5%D1%82%D0%B8) имеет возможность, вместо использования «зашитого», назначить устройству MAC-адрес по своему усмотрению. Такой **локально администрируемый MAC-адрес** выбирается произвольно.

**Логический адрес**

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100, IP-адрес назначается администратором при конфигурировании компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно. Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей, поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Та-

ким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

**Классы сетей**

Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса:

* Если адрес начинается с 0, то сеть относят к классу А, и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) В сетях класса А количество узлов должно быть больше 216, но не превышать 224.
* Если первые два бита адреса равны 10, то сеть относится к классу В и является сетью средних размеров с числом узлов 28 - 216. В сетях класса В под адрес сети и под адрес узла отводится по 16 битов, то есть по 2 байта.
* Если адрес начинается с последовательности 110, то это сеть класса С с числом узлов не больше 28. Под адрес сети отводится 24 бита, а под адрес узла - 8 битов.
* Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.
* Если адрес начинается с последовательности 11110, то это адрес класса Е, он зарезервирован для будущих применений.

**18. Прямое и обратное разрешение адресов. Протоколы разрешения**

**адресов (ARP, RARP).**

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии.

Взаимодействие технологии TCP/IP с частными технологиями подсетей происходит многократно при перемещении пакета IP по составной сети. На каждом маршрутизаторе протокол IP определяет, в какую следующую подсеть и какому пограничному узлу в этой подсети надо направить пакет. Таким пограничным узлом является маршрутизатор, и протоколу IP известен его IP-адрес. Очевидно, что для того, чтобы частная технология подсети смогла доставить пакет на следующий маршрутизатор, необходимо:

□ во-первых, упаковать пакет в кадр соответствующего для данной подсети формата (например, Ethernet);

□ во-вторых, снабдить кадр адресом, формат которого был бы понятен локальной технологии подсети (преобразовать, например, IP-адрес в МАС-адрес).

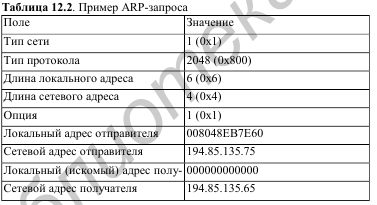
Для определения локального адреса по IP-адресу используется протокол разрешения, адресов (Address Resolution Protocol, **ARP**).

Рассмотрим работу протокола ARP в сетях с широковещанием. В результате

конфигурирования сети каждый интерфейс знает свои IP-адрес и МАС-адрес. Кроме того, на каждом интерфейсе (сетевом адаптере или порту маршрутизатора) поддерживается отдельная ARP-таблица, определяющая соответствие между IP-адресами и МАС-адресами других узлов данной подсети. Первоначально, при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

Пусть в какой-то момент модуль IP передаст пакет на уровень сетевых интерфейсов, например драйверу Ethernet, и ему требуется найти на основе известного IP-адреса МАС-адрес узла назначения. Для этого протокол IP обращается к протоколу ARP. Работа ARP начинается с просмотра ARP-таблицы соответствующего интерфейса. В случае, если оказалось невозможным определить локальный адрес исходящего IP-пакета из ARP-таблицы, то этот пакет запоминается в буфере, а протокол ARP формирует запрос (ARP-запрос), вкладывает его в кадр протокола канального уровня и рассылает широковещательно.

Все интерфейсы подсети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. Интерфейс, который констатировал совпадение, формирует ARP-ответ, указывая в нем свой IP-адрес и свой локальный адрес, а затем отправляет его уже направленно, так как в ARP-запросе имеется локальный адрес отправителя. ARP-запросы и ответы используют один и тот же формат пакета.



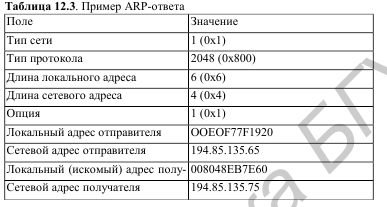
В поле «Тип сети» для сетей Ethernet указывается значение 1.

Поле «Тип протокола» позволяет использовать протокол ARP не только для протокола IP, но и для других сетевых протоколов. Для IP значение этого поля равно 0x0800.

Длина локального адреса для протокола Ethernet равна 6 байт, а длина IP-адреса — 4 байт. В поле операции для ARP-запросов указывается значение 1, если это запрос, и 2, если это ответ.

Из этого запроса видно, что в сети Ethernet узел с IP-адресом 194.85.135.75 пытается определить, какой МАС-адрес имеет другой узел той же сети, сетевой адрес которого 194.85.135.65. Поле искомого локального адреса заполнено нулями.

Ответ присылает узел, опознавший свой IP-адрес. Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу.



Этот ответ получает машина, отправившая ARP-запрос. В результате обмена этими двумя ARP-сообщениями модуль IP, имеющий адрес 194.85.135.75, определяет, что IP-адресу 194.85.135.65 соответствует МАС-адрес OOEOF77F1920. Этот адрес будет затем помещен в заголовок кадра Ethernet, ожидавшего отправления IP-пакета.

Чтобы уменьшить число ARP-обращений в сети, найденное соответствие между IP-адресом и МАС-адресом записывается в ARP-таблицу соответствующего интерфейса.

Теперь, если вдруг вновь возникнет необходимость послать пакет по адресу 194.85.135.65, то протокол, прежде чем посылать широковещательный запрос, проверит, нет ли уже такого адреса в ARP-таблице. Ниже показана ARP-таблица.

Поле «Тип записи» может содержать одно из двух значений — «динамический» или «статический». Статические записи создаются вручную с помощью утилиты агр и существуют до тех пор, пока компьютер или маршрутизатор не будут выключены.



Динамические же записи создаются модулем протокола ARP, использующим

широковещательные возможности локальных сетевых технологий.

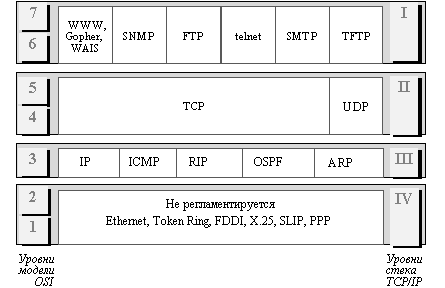
Динамические записи должны периодически обновляться. Если запись не обновлялась в течение определенного времени, то она исключается из таблицы. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют ARP-кэшем.

В некоторых случаях возникает обратная задача — нахождение IP-адреса по известному локальному адресу. Тогда в действие вступает реверсивный протокол ARP (Reverse Address Resolution Protocol, **RARP**). Этот протокол используется, например, при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих МАС-адрес своего сетевого адаптера.

**19. Иерархия протоколов в IP-сетях (стек протоколов IP).**

Так как стек TCP/IP был разработан до появления модели OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Структура протоколов TCP/IP приведена на рисунке. Протоколы TCP/IP делятся на 4 уровня.



Самый нижний (***уровень IV***) соответствует **физическому и канальному уровням модели OSI**. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня: для локальных сетей это Ethernet, Token Ring, FDDI, Fast Ethernet, для глобальных сетей - протоколы соединений "точка-точка" SLIP и PPP, frame relay. Обычно при появлении новой технологии локальных или глобальных сетей она быстро включается в стек TCP/IP за счет разработки соответствующего RFC, определяющего метод инкапсуляции пакетов IP в ее кадры.

Следующий уровень (***уровень III***) - это уровень **межсетевого взаимодействия**, который занимается передачей пакетов с использованием различных транспортных технологий локальных сетей, территориальных сетей, линий специальной связи и т. п.

В качестве основного протокола сетевого уровня в стеке используется протокол **IP**, который изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенных как локальными, так и глобальными связями. Поэтому протокол IP хорошо работает в сетях со сложной топологией, рационально используя наличие в них подсистем и экономно расходуя пропускную способность низкоскоростных линий связи. Протокол IP является дейтаграммным протоколом, то есть он не гарантирует доставку пакетов до узла назначения, но старается это сделать.

К уровню межсетевого взаимодействия относятся и все протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации **RIP**(Routing Internet Protocol) и **OSPF**(Open Shortest Path First), а также протокол межсетевых управляющих сообщений **ICMP** (Internet Control Message Protocol). Последний протокол предназначен для обмена информацией об ошибках между маршрутизаторами сети и узлом - источником пакета.

Следующий уровень (***уровень II***) называется **основным**. На этом уровне функционируют протокол управления передачей**TCP** (Transmission Control Protocol) и протокол дейтаграмм пользователя **UDP**(User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования виртуальных соединений. Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным способом, как и IP, и выполняет только функции связующего звена между сетевым протоколом и многочисленными прикладными процессами.

Верхний уровень (***уровень I***) называется **прикладным**. Он включает в себя такие протоколы как, как протокол копирования файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовые сервисы доступа к удаленной информации, такие как WWW и многие другие.

Протокол пересылки файлов **FTP** (File Transfer Protocol) реализует удаленный доступ к файлу. Для того, чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений - TCP. Кроме пересылки файлов протокол FTP предлагает и другие услуги. Так, пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов. Наконец, FTP выполняет аутентификацию пользователей. Прежде, чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое имя и пароль. Для доступа к публичным каталогам FTP-архивов Internet парольная аутентификация не требуется, и ее обходят за счет использования для такого доступа предопределенного имени пользователя Anonymous.

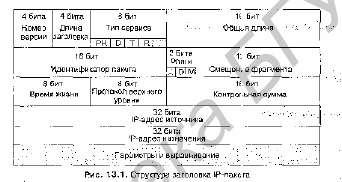
Протокол **telnet** обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленного компьютера. При использовании сервиса telnet пользователь фактически управляет удаленным компьютером так же, как и локальный пользователь, поэтому такой вид доступа требует хорошей защиты. Поэтому серверы telnet всегда используют как аутентификацию по паролю.

**20. Протокол IP. Назначение, особенности, характеристики. IP-сети.**

Протокол IP протокол сетевого уровня в стеке TCP/IP. Название протокола IP — протокол межсетевого взаимодействия (Internet Protocol, IP) — отражает его суть: он должен передавать пакеты между сетями. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно к узлу-получателю.

Протокол IP относится к протоколам без установления соединения. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных: отсутствует квитирование — обмен подтверждениями между отправителем и получателем, нет процедуры упорядочивания, повторных передач или других подобных функций. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Все вопросы обеспечения надежности при доставке данных по составной сети в стеке TCP/IP решает протокол TCP, работающий непосредственно над протоколом IP. Именно TCP организует повторную передачу пакетов, когда в этом возникает необходимость.

**Структура IP-пакета**

IP-пакет состоит из заголовка и поля данных. Заголовок имеет следующую структуру 

Поле **Номер версии** (Version), указывает версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4) и готовится переход на версию 6 (IPv6).

Поле **Длина заголовка** (IHL) указывает значение длины заголовка.

Поле **Тип сервиса** (Type of Service, ToS) занимает один байт. Оно используется пока очень редко и обычно в нем устанавливается 0. Это поле позволяет задавать приоритетность пакета и критерий выбора маршрута. Приоритет может иметь значения от самого низкого — 0 (нормальный пакет) до самого высокого — 7 (пакет управляющей информации).

Поле **Общая длина** (Total Length) означает общую длину пакета с учетом заголовка и поля данных.

Поле **Идентификатор пакета** (Identification) используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета.

Поле **Смещение фрагмента** (Fragment Offset) задает смещение в байтах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации.

Поле **Время жизни** (Time to Live) означает предельный срок, в течение которого пакет может перемешаться по сети.

Идентификатор **Протокол верхнего уровня** (Protocol) указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета.

**Контрольная сумма** (Header Checksum) рассчитывается только по заголовку. При вычислении контрольной суммы значение самого поля «контрольная сумма» устанавливается в нуль. Если контрольная сумма неверна, то пакет будет отброшен, как только ошибка обнаружится.

Поля **IP-адрес источника** (Source IP Address) и **IP-адрес назначения** (Destination IP Address) имеют одинаковую структуру.

**Понятие IP-сети**

Под IP-сетью подразумевается сеть, построенная на базе стека протоколов TCP/IP, который позволяет создавать как небольшие локальные, так и глобальные сети. Частным случаем IP-сети является глобальная сеть Internet. Корпоративная IP-сеть, построенная с применением интернет-сервисов для внутреннего пользования, называется Intranet.

**21. Адресация IP.**

IP-адрес представляет собой 32-разрядный номер, который уникально идентифицирует узел (компьютер или устройство, например, принтер или маршрутизатор) в сети TCP/IP.  
  
IP-адреса обычно представлены в виде 4-х разрядов, разделенных точками, например 192.168.123.132. В двоичном обозначении этот адрес представляет собой 32-разрядный номер 110000000101000111101110000100. Такой номер сложно интерпретировать, поэтому разбейте его на четыре части по восемь двоичных знаков.  
  
Эти 8-разрядные секции называются «октеты». Тогда данный IP-адрес будет иметь вид: 11000000.10101000.01111011.10000100. Десятичные числа, разделенные точками, и есть октеты, преобразованные из двоичного в десятичное обозначение.  
  
Чтобы глобальная сеть TCP/IP работала эффективно как совокупность сетей, маршрутизаторы, обеспечивающие обмен пакетами данных между сетями, не знают точного расположения узла, для которого предназначен пакет. Маршрутизаторы знают только, к какой сети принадлежит узел, и используют сведения, хранящиеся в таблицах маршрутизации, чтобы доставить пакет в сеть узла назначения. Как только пакет доставлен в необходимую сеть, он доставляется в соответствующий узел.  
  
Для осуществления этого процесса IP-адрес состоит из двух частей. Первая часть IP-адреса обозначает адрес сети, последняя часть – адрес узла. Если рассмотреть IP-адрес 192.168.123.132 и разбить его на эти две части, то получится следующее:

192.168.123.0 – адрес сети.

0.0.0.132 – адрес узла.

**Маска подсети**

Протокол TCP/IP использует маску подсети, чтобы определить, в какой сети находится узел: в локальной подсети или удаленной сети.  
  
В протоколе TCP/IP части IP-адреса, используемые в качестве адреса сети и узла, не зафиксированы, следовательно, указанные выше адреса сети и узла невозможно определить без наличия дополнительных сведений. Данные сведения можно получить из другого 32-разрядного номера под названием «маска подсети». В этом примере маской подсети является 255.255.255.0. Значение этого номера понятно, если знать, что число 255 в двоичном обозначении соответствует числу 11111111; таким образом, маской подсети является номер:

11111111.11111111.11111111.0000000

Расположив следующим образом IP-адрес и маску подсети, можно выделить составляющие сети и узла:

11000000.10101000.01111011.10000100 – IP-адрес (192.168.123.132)

11111111.11111111.11111111.00000000 – маска подсети (255.255.255.0)

Первые 24 разряда (число единиц в маске подсети) распознаются как адрес сети, а последние 8 разрядов (число оставшихся нолей в маске подсети) – адрес узла. Таким образом, получаем следующее:

11000000.10101000.01111011.00000000 – адрес сети (192.168.123.0)

00000000.00000000.00000000.10000100 – адрес узла (000.000.000.132)

Из данного примера с использованием маски подсети 255.255.255.0 видно, что код сети 192.168.123.0, а адрес узла 0.0.0.132. Когда пакет с конечным адресом 192.168.123.132 доставляется в сеть 192.168.123.0 (из локальной подсети или удаленной сети), компьютер получит его из сети и обработает.  
  
**Классы сетей**

Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса:

* Если адрес начинается с 0, то сеть относят к классу А, и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) В сетях класса А количество узлов должно быть больше 216, но не превышать 224.
* Если первые два бита адреса равны 10, то сеть относится к классу В и является сетью средних размеров с числом узлов 28 - 216. В сетях класса В под адрес сети и под адрес узла отводится по 16 битов, то есть по 2 байта.
* Если адрес начинается с последовательности 110, то это сеть класса С с числом узлов не больше 28. Под адрес сети отводится 24 бита, а под адрес узла - 8 битов.
* Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.
* Если адрес начинается с последовательности 11110, то это адрес класса Е, он зарезервирован для будущих применений.

В таблице приведены диапазоны номеров сетей, соответствующих каждому классу сетей.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Наименьший адрес** | **Наибольший адрес** |
| **A** | 01.0.0 | 126.0.0.0 |
| **B** | 128.0.0.0 | 191.255.0.0 |
| **C** | 192.0.1.0 | 223.255.255.0 |
| **D** | 224.0.0.0 | 239.255.255.255 |
| **E** | 240.0.0.0 | 247.255.255.255 |

**Основные шлюзы**

Связь между TCP/IP-компьютером и узлом из другой сети обычно осуществляется через устройство, называемое маршрутизатором. С точки зрения TCP/IP маршрутизатор, указанный на узле, связывающем подсеть узла с другими сетями, называется основным шлюзом.   
При попытке установления связи между узлом и другим устройством с помощью протокола TCP/IP узел сопоставляет определенную маску подсети и IP-адрес назначения с маской подсети и своим собственным IP-адресом. В результате этого сопоставления компьютер узнает, для какого из узлов предназначен данный пакет – локального или удаленного.  
  
Если в результате этого процесса назначением является локальный узел, то компьютер просто отправляет пакет в локальную подсеть. Если в результате сопоставления выясняется, что назначением является удаленный узел, компьютер направляет пакет на основной шлюз, определенный в свойствах TCP/IP. Таким образом, именно маршрутизатор отвечает за отправку пакета в правильную подсеть.

**Соглашения о специальных адресах**

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

* если IР-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет;
* если в поле номера сети стоят 0,

|  |
| --- |
| 0 0 0 0 .......0 Номер узла |

то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет;

* если все двоичные разряды IP-адреса равны 1,

|  |
| --- |
| 1 1 1 1 .........................................1 1 |

то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета.

* если в поле адреса назначения стоят сплошные 1,

|  |
| --- |
| Номер сети 1111................11 |

то пакет, имеющий такой адрес рассылается всем узлам сети с заданным номером.

**22. Параметры настройки IP. Средства конфигурирования.**

***Тестирование стека TCP/IP с помощью утилиты Ping***

*Ping (Packet Internet Gropper)* - это диагностическое средство, используемое для тестирования конфигурации сети TCP/IP и диагностики ошибок соединения. Ping использует сообщения echo request и echo reply протокола ICMP, для того чтобы определить, доступен ли хост с определенным адресом и работоспособен ли он. Если Ping завершается успешно, то он выдает следующее сообщение:

Для проверки корректности конфигурации вашего компьютера и тестирования соединения с маршрутизатором необходимо выполнить следующие действия.

1. Выполните команду Ping с адресом обратной связи для того, чтобы проверить, что стек TCP/IP установлен правильно:
2. ping 127.0.0.1
3. Выполните команду Ping с IP-адресом вашего компьютера для проверки возможного дублирования IP-адресов:
4. ping IP-адрес вашего компьютера
5. Выполните команду Ping с IP-адресом маршрутизатора по умолчанию:
6. ping IP-адрес маршрутизатора по умолчанию
7. Выполните команду Ping с IP-адресом удаленного хоста, чтобы проверить, что вы можете осуществлять взаимодействие через маршрутизатор:
8. ping IP-адрес вашего хоста

#### Использование службы DHCP для автоматической конфигурации сети

Протокол Dynamic Host Configuration Protocol (DHCP) обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие конфликтов адресов за счет централизованного управления их распределением. DHCP динамически распределяет IP-адреса для компьютеров. Администратор управляет процессом назначения адресов с помощью параметра "продолжительности аренды" (lease duration), которая определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от сервера DHCP в аренду.

Примером работы протокола DHCP может служить ситуация, когда компьютер, являющийся клиентом DHCP, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

Протокол DHCP использует модель клиент-сервер (рисунок 9.2). Во время старта системы (состояние "инициализация") компьютер-клиент DHCP посылает сообщение discover (исследовать), которое широковещательно распространяется по локальной сети и передается всем DHCP-серверам частной интерсети. Каждый DHCP-сервер, получивший это сообщение, отвечает на него сообщением offer (предложение), содержащее IP-адрес и конфигурационную информацию.

Компьютер-клиент DHCP переходит в состояние "выбор" и собирает конфигурационные предложения от DHCP-серверов. Затем он выбирает одно из этих предложений, переходит в состояние "запрос" и отправляет сообщение request (запрос) тому DHCP-серверу, чье предложение было выбрано.

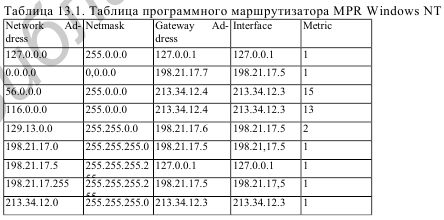
Выбранный DHCP-сервер посылает сообщение acknowledgment (подтверждение), которое содержит IP-адрес, который уже был послан ранее на стадии исследования, а также параметр аренды для этого адреса. После того, как клиент получит это подтверждение, он переходит в состояние "связь", находясь в котором он может принимать участие в работе сети TCP/IP. Компьютеры-клиенты, которые имеют локальные диски, сохраняют полученный адрес для использования при последующих стартах системы. При приближении момента истечения срока аренды адреса компьютер пытается обновить параметры аренды у DHCP-сервера, а если этот IP-адрес не может быть выделен снова, то ему возвращается другой IP-адрес.

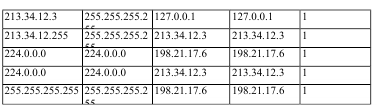
Для того, чтобы клиентский компьютер получал параметры стека от сервера DHCP, необходимо в параметрах стека TCP/IP отметить опцию "Obtain an IP address from DHCP server".

**23. IP-маршрутизация.**

Программные модули протокола IP устанавливаются на всех конечных станциях и маршрутизаторах сети. Для продвижения пакетов они используют таблицы маршрутизации.

Если представить, что в качестве маршрутизатора в работает штатный программный маршрутизатор MPR операционной системы Microsoft Windows NT, то его таблица маршрутизации могла бы выглядеть так же, как





**Назначение полей таблицы маршрутизации**

В таблице есть ключевые параметры, необходимые для работы.

К таким параметрам, во-первых, относится адрес сети назначения (столбец «Network Address» в маршрутизаторе MPR).

Вторым обязательным полем таблицы маршрутизации является адрес следующего маршрутизатора (столбец «Gateway Address» в маршрутизаторе MPR).

Третий ключевой параметр — адрес порта, на который нужно направить пакет (поле «Interface» в таблице Windows NT).

Остальные параметры для принятия решения о пути следования пакета являются необязательными.

Наличие или отсутствие поля маски в таблице говорит о том, насколько современен данный маршрутизатор. Стандартным решением сегодня является использование поля маски в каждой записи таблицы, как это сделано в таблице маршрутизатора MPR Windows NT (поле «Netmask»).

**Источники и типы записей в таблице маршрутизации**

Одним из источников записей в таблицу маршрутизации является программное обеспечение стека TCP/IP, которое при инициализации маршрутизатора автоматически заносит в таблицу несколько записей, в результате чего создается так называемая минимальная таблица маршрутизации.

Это, во-первых, записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, информация о которых появляется в стеке при ручном конфигурировании интерфейсов компьютера или маршрутизатора. К таким записям в приведенных примерах относятся записи о сетях 213.34.12.0 и 198.21.17.0, а также запись 0.0.0.0 в маршрутизаторе MPR Windows NT.

Во-вторых, программное обеспечение автоматически заносит в таблицу маршрутизации записи об адресах особого назначения. В приведенных примерах таблица маршрутизатора MPR Windows NT содержит наиболее полный набор записей такого рода. Несколько записей в этой таблице связаны с особым адресом 127.0.0.0 (loopback), который используется для локального тестирования стека TCP/IP. Пакеты, направленные в сеть с номером 127.0.0.0, не передаются протоколом IP на канальный уровень для последующей передачи в сеть, а возвращаются в источник — локальный модуль IP. Записи с адресом 224.0.0.0 требуются для обработки групповых адресов (multicast address).

И, наконец, третьим источником записей могут быть протоколы маршрутизации, такие как RIP или OSPF. Подобные записи всегда являются динамическими, то есть имеют ограниченный срок жизни. Программный маршрутизатор Windows NT не показывает источник появления той или иной записи в таблице.

**24. Протокол TCP: назначение, характеристика, особенности применения.**

В то время как задачей уровня межсетевого взаимодействия, к которому относится протокол IP, является передача данных между любой парой сетевых интерфейсов в составной сети, задача транспортного уровня, которую решают протоколы TCP и UDP, заключается в передаче данных между любой парой прикладных процессов, выполняющихся в сети.

**Надежный протокол доставки сообщений TCP**

Если роль связующего звена между сетью и приложениями берут на себя оба протокола транспортного уровня (TCP и UDP), то задачу обеспечения надежного канала обмена данными между прикладными процессами в составной сети решает только протокол TCP.

Протокол TCP работает непосредственно над протоколом IP и использует для транспортировки своих блоков данных этот дейтаграммный и поэтому потенциально ненадежный протокол. Надежность передачи данных протоколом TCP достигается за счет того, что он основан на установлении логических соединений между взаимодействующими процессами. До тех пор пока модули протокола TCP на обоих оконечных хостах продолжают функционировать корректно, а составная сеть не распалась на несвязные части, ошибки в передаче данных на уровне протокола IP не будут влиять на правильность обмена хостов сегментами.

Информация, поступающая к протоколу TCP в рамках логического соединения от протоколов более высокого уровня, рассматривается протоколом TCP как неструктурированный поток байтов. Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая называется сегментом.

Сегменты могут быть разного размера, однако об их максимальных размерах участники соединения должны договориться. Заметим, что сегментом называют как единицу передаваемых данных в целом (поле данных и заголовок TCP), так и отдельно поле данных.

Протокол IP используется протоколом TCP в качестве транспортного средства. Перед отправкой сегментов протокол TCP помещает их в оболочку IP-пакета. В заголовок сегмента протокол TCP помещает номер порта назначения и номер порта отправителя.

**Соединения**

Для надежной передачи данных между двумя прикладными процессами предусматривается установление логического соединения. Номер порта в совокупности с номером сети и номером конечного узла однозначно определяют прикладной процесс в сети. Этот набор идентифицирующих параметров (IP-адрес, номер порта) имеет название сокет (socket).

Каждый взаимодействующий процесс идентифицируется сокетом, а каждое соединение — парой сокетов взаимодействующих процессов.

В протоколе TCP каждая сторона соединения посылает противоположной стороне следующие параметры:

□ максимальный размер сегмента, который она готова принимать;

□ максимальный объем данных (возможно несколько сегментов), которые она разрешает другой стороне передавать в свою сторону, даже если та еще не получила подтверждения на предыдущую порцию данных;

□ начальный порядковый номер байта, с которого она начинает отсчет потока данных в рамках данного соединения.

После того как в результате переговорного процесса модулей TCP с двух сторон соединения параметры процедуры обмена определены, одни из них остаются постоянными в течение всего сеанса связи, а некоторые адаптивно изменяются.

Соединение в TCP позволяет вести передачу данных одновременно в обе стороны, то есть полнодуплексную передачу.

[к 26 вопросу](#вопрос26)

**Последовательный номер и номер подтверждения**

В рамках установленного соединения правильность передачи каждого сегмента должна подтверждаться квитанцией получателя. Квитирование — это один из традиционных методов обеспечения надежной связи.

Когда отправитель посылает сегмент, он помещает в одно из полей заголовка TCP в качестве идентификатора сегмента номер первого байта данного сегмента, который называют последовательным номером (sequence number). На основании этого номера модуль TCP-получатель не только отличает данный сегмент от других, но и позиционирует полученный фрагмент относительно общего потока байтов, он может сделать также вывод, не является ли этот сегмент дубликатом и не было ли пропусков между полученным сегментом и предыдущим и т. д.

В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в которое помещает число, на единицу превышающее максимальный номер байта в полученном сегменте. Это число называют номером подтверждения (acknowledgement number). Номер подтверждения также является номером для следующего ожидаемого байта данных, передаваемого в обратном направлении. Квитанция (подтверждение) в протоколе TCP посылается только в случае правильного приема данных, отрицательные квитанции не посылаются.

В протоколе TCP в одном и том же сегменте могут быть помещены и данные, которые приложение посылает другой стороне, и квитанция, которой модуль TCP подтверждает получение данных.

**Время ожидания квитанции**

Когда протокол TCP передает в сеть сегмент, он «на всякий случай» помещает его копию в очередь повторной передачи и запускает таймер. Когда приходит подтверждение на этот сегмент, соответствующая копия удаляется из очереди. Если же подтверждение не приходит до истечения срока, то сегмент посылается повторно. Может случиться, что повторный сегмент придет тогда, когда исходный сегмент уже окажется на месте, тогда дубликат будет попросту отброшен.

Выбор времени ожидания (тайм-аута) очередной квитанции является важной задачей, результат решения которой влияет на производительность протокола TCP. В протоколе TCP тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент.

**25. Протокол UDP: назначение, характеристика, особенности применения.**

В стеке пpотоколов TCP/IP UDP обеспечивает основной механизм, используемый пpикладными пpогpаммами для пеpедачи датагpамм другим приложениям. Помимо посылаемых данных каждое UDP-сообщение содеpжит номеp поpта-пpиемника и номеp поpта-отпpавителя, делая возможным для программ UDP на машине-получателе доставлять сообщение соответствующему реципиенту, а для получателя посылать ответ соответствующему отправителю.

UDP использует Internet Protocol для пеpедачи сообщения от одной мащины к дpугой и обеспечивает ту же самую ненадежную доставку сообщений, что и IP. UDP не использует подтвеpждения пpихода сообщений, не упоpядочивает пpиходящие сообщения и не обеспечивает обpатной связи для управления скоростью передачи инфоpмации между машинами. Поэтому, UDP-сообщения могут быть потеpяны, pазмножены или пpиходить не по поpядку. Кpоме того, пакеты могут пpиходить pаньше, чем получатель сможет обpаботать их.

Пpикладные пpогpаммы, использующие UDP, несут полную ответственность за пpоблемы надежности, включая потеpю сообщений, дублирование, задеpжку, неупоpядоченность или потеpю связи.

#### Фоpмат UDP-сообщений

Каждое UDP-сообщение называется пользовательской датагpаммой. Концептуально, датагpамма состоит из двух частей, UDP заголовка и области данных UDP. Заголовок состоит из четыpех полей, котоpые опpеделяют поpт, из котоpого было послано сообщение, поpт, в котоpый сообщение пpиходит, длину сообщения и контpольную сумму UDP.

**26. Программный интерфейс сетевых функций. Соединители. WinSocket.**

Соединитель и сокет это одно и тоже. Про сокеты читать [тут](#сокет)

И еще инфа про сокеты:

Сокеты поддерживают целый ряд протоколов, самыми популярными из которых сегодня являются User Datagram Protocol (UDP) и Transmission Control Protocol (TCP).

UDP-сокеты не требуют установления логических соединений и обычно применяются для широковещательной и многоадресной (multicast) связи. В UDP нет средств надежной доставки сообщений и контроля правильного порядка пакетов, поэтому за обнаружение потери пакетов, устранение таких проблем и упорядочение пакетов отвечает приложение-получатель.

TCP-сокеты ориентированы на логические соединения, предоставляя надежный коммуникационный путь двумя конечными точками. Важное преимущество TCP в том, что он гарантирует доставку сообщений и правильный порядок пакетов. В этой статье основное внимание уделяется TCP-сокетам.

TCP-сокеты могут быть либо клиентскими, либо серверными. Серверный сокет ожидает запросы на установление соединений, а клиентский — инициирует соединение. Как только соединение между сокетами установлено, клиент и сервер могут передавать и принимать данные или закрыть это соединение.

**WinSock** или **Windows socket** - это интерфейс программного программирования (**API**) созданный для реализации приложений в сети на основе протокола **TCP/IP**. То есть это просто группа функций.

Давайте расмотрим как производится подключение к серверу. Сначала программа подключается к адресу **IP** с созданием сокета. Программа будет ждать подключения. Для подключения программа клиент тоже создает сокет и пытается подключиться к сокету сервера. Но сервер не спит !!! Только он увидит попытку подключения он создаст новый сокет. И этот новый сокет будет использоваться для взаимодействия с клиентом. А тот, к которому была попытка подключения будет ждать следующего. На этой основе производится взаимодействие сервера и многими программами.

**27. Программирование TCP и UDP.**

**Взаимодействие без установления соединения (UDP)**

Этот вид взаимодействия предусматривает обмен датаграммами, характеризуется минимальным уровнем сервиса со стороны системы и минимальными служебными затратами. Целостность передаваемых данных системой не гарантируются, при необходимости о ней должны заботиться сами взаимодействующие программы.

Полезной особенностью большинства протоколов датаграммной передачи, в том числе и UDP, является широковещание – доставка соответствующим образом адресованного сообщения всем доступным узлам сети (подсети).

Действия клиента и сервера в общем схожи, и с точки зрения программирования сокетов отличия между ними сугубо номинальные. Каждый из них использует всего один сокет, служащий и для приема, и для передачи сообщений.

Характерным примером сетевой службы с подобным порядком взаимодействия служит echo: «эхо»-сервер принимает все запросы от всех клиентов и немедленно возвращает их отправителям.

Важно отметить, что некоторые действия в программах являются блокирующими, то есть приостанавливают выполнение на неопределенный срок. В первую очередь, это ожидание поступления сообщения. Подобная ситуация очень характерна для сетевых приложений, причем она усугубляется неопределенностью состояния партнера и каналов связи. В результате, например, при отсутствии сервера может произойти как распознавание

ошибки приема, так и переход клиента в состояние бесконечного ожидания.

**Взаимодействие с установлением соединения (TCP)**

Наличие постоянного соединения (виртуального канала) усложняет порядок взаимодействия, но зато обеспечивает более развитый сервис, в первую очередь контроль целостности сеанса и передаваемых данных.

В начале работы сервер создает и связывает с локальным адресом единственный сокет (назовем его базовый сокет), на котором он будет производить прием запросов на соединение. Именно порт, с которым он связан, указывается в качестве порта, занятого данным сервером. Далее базовый сокет переводится в режим «прослушивания», и, начиная с этого момента, в системе организуется очередь, принимающая запросы на соединение. Основной цикл сервера состоит в приеме этих запросов вызовом accept(). При поступлении такого запроса системой создается новое соединение, и вызов accept() на стороне сервера успешно завершается, возвращая новый сокет, связанный с этим соединением – сокет данных, уже готовый к использованию. Адрес для сокета данных выбирается системой автоматически, и он может совпадать с адресом базового сокета.

После этого сервер начинает вложенный цикл обмена данными (сообщениями) через новое соединение, чаще всего это ожидание входящего запроса, его обработка и отсылка ответа клиенту. В отличие от UDP при передаче данных адрес назначения указывать уже не нужно, поэтому используется более простой вызов send(). Признаком завершения в простейшем случае может служить прием одной или нескольких порций данных нулевой длины. В протоколе прикладного уровня может быть также предусмотрены извещения от клиента о прекращении обмена или возможность разрыва соединения по инициативе сервера. После окончания вложенного цикла сервер может вернуться к приему следующего соединения.

Со своей стороны клиент создает сокет и передает ее в вызов connect(), указывая также и адрес сервера, с которым нужно установить соединение – IP-адрес хоста, где находится сервер, и закрепленный за ним порт. После успешного завершения вызова connect() сокет будет связан с созданным соединением и готов к работе, клиент может начинать цикл обмена данными с сервером. Блокирующими вызовами здесь помимо recv() являются accept() и connect(). Вызов accept() фактически принимает запрос на соединение из очереди, и если очередь пуста, то он остается в состоянии ожидания. Вызов connect()

блокируется, если сгенерированный им запрос на соединение не обслужен сервером, но поставлен в очередь. Если в очереди сервера свободных мест нет, то запрос отвергается немедленно и соответствующий connect() клиента завершается как неуспешный. Кроме того, во время обслуживания клиентского соединения сервер с описанным алгоритмом приостанавливает прием новых соединений.

**28. Сетевые приложения: принципы построения, программирование.**

Компьютер, подключенный к сети, может выполнять следующие типы приложений:

* Локальное приложение целиком выполняется на данном компьютере и использует только локальные ресурсы. Для такого приложения не требуется никаких сетевых средств, оно может быть выполнено на автономно работающем компьютере.
* Централизованное сетевое приложение целиком выполняется на данном компьютере, но обращается в процессе своего выполнения к ресурсам других компьютеров сети. Например приложение, которое выполняется на клиентском компьютере, обрабатывает данные из файла, хранящегося на файл-сервере, а затем распечатывает результаты на принтере, подключенном к серверу печати. Очевидно, что работа такого типа приложений невозможна без участия сетевых служб и средств транспортировки сообщений.
* Распределенное (сетевое) приложение состоит из нескольких взаимодействующих частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи, причем каждая часть может выполняться и, как правило, выполняется на отдельном компьютере сети. Части распределенного приложения взаимодействуют друг с другом, используя сетевые службы и транспортные средства ОС. Распределенное приложение в общем случае имеет доступ ко всем ресурсам компьютерной сети.

Очевидным преимуществом распределенных приложений является возможность распараллеливания вычислений, а также специализация компьютеров. Например, в приложении, предназначенном, скажем, для анализа климатических изменений, можно выделить три достаточно самостоятельные части, допускающие распараллеливание.

Первая часть приложения, выполняющаяся на сравнительно маломощном персональном компьютере, могла бы поддерживать специализированный графический пользовательский интерфейс, вторая — заниматься статистической обработкой данных на высокопроизводительном мэйнфрейме, а третья — генерировать отчеты на сервере с установленной стандартной СУБД. В общем случае каждая из частей распределенного приложения может быть представлена несколькими копиями, работающими на разных компьютерах.

Однако, чтобы добиться всех тех преимуществ, которые сулят распределенные приложения, разработчикам этих приложений приходится решать множество проблем, например: на сколько частей следует разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу и т. д. Заметим, что все сетевые службы, включая файловую службу, службу печати, службу электронной почты, службу удаленного доступа, интернет-телефонию и т. д., по определению относятся к классу распределенных приложений.

Действительно, любая сетевая служба включает в себя клиентскую и серверную части, которые могут и обычно выполняются на разных компьютерах.

**29. Обеспечение функционирования IP. Средства управления, настройки, диагностики.**

В операционных системах Windows за настройку, поддержку и анализ проблем в TCP/IP сетях отвечает оболочка сетевых сервисов **Netsh**. Netsh – это системная утилита-сценарий командной строки, позволяющая управлять конфигурацией различных сетевых служб, как на локальном, так и на удаленном компьютерах. Утилита Netsh для корректной работы тесно взаимодействует с другими компонентами операционной системы, используя динамические библиотеки.

## Добавление IPv4 адресов

Иногда вам может понадобиться, чтобы один компьютер выглядел в сети как несколько компьютеров или, чтобы ваша сеть была разделена на разные подсети и компьютеру необходим доступ к обеим подсетям только с одним сетевым адаптером. При помощи утилиты командной строки Netsh и команды **Add Address** вы можете выполнить такие действия, то есть добавить новые IPv4 адреса и основные шлюзы на определенный интерфейс со статическим адресом. Стоит обратить внимание на то, что если для сетевого интерфейса IPv4-адреса будут назначаться DHCP-сервером, при добавлении адреса она будет отключена. Синтаксис команды следующий:

Netsh Interface IPv4 Add Address name=[имя\_подключения] address=[ IPv4-адрес] mask=[маска] type=[тип] gateway=[шлюз] gwmetric=[метрика\_шлюза] остальные\_параметры

## Настройка конфигурации IPv4

Как администраторам, так и штатным пользователям операционных систем Windows как минимум один раз в жизни приходилось менять IP-конфигурацию на своем компьютере. При обнаружении сетевой платы во время установки операционной системы по умолчанию включаются две версии протокола IP – IPv4 и IPv6. Также по умолчанию всем подключениям IPv4-адреса назначаются автоматически для того чтобы компьютер мог подключаться к DHCP-серверам. Компьютеры с таким типом подключения называются DHCP-клиентами. Из командной строки вы можете настраивать IP-адресацию вручную или автоматически. При настройке вручную вы назначаете компьютеру статический IP-адрес. Как вы уже знаете, такой адрес является фиксированным и не изменится до тех пор, пока вы вручную его не поменяете. При динамической настройке вы конфигурируете компьютер так, чтобы он получал свой IP-адрес от DHCP-сервера, расположенного в той же сети. Такой адрес всегда назначается при загрузке компьютера, и он может меняться.

Для настройки IP-конфигурации сетевых подключений вы также можете использовать утилиту командной строки Netsh, где статические и динамические IP-адреса назначаются в контексте Interface IPv4при помощи команды Set Address. Для работы с данной командой используется следующий синтаксис:

Netsh Interface IPv4 Add Address name=[имя\_подключения] source=[dhcp | static] address=[ IPv4-адрес] mask=[маска] gateway=[шлюз] gwmetric=[метрика\_шлюза] остальные\_параметры

## Отображение сведений конфигурации IPv4

Во время настройки сетевых подключений вам может понадобиться просмотреть конфигурацию существующих подключений. Функционал утилиты контекста **Interface IPv4** утилиты **Netsh** позволяет вам выполнять такие действия. Команда **Show addresses** предназначена для проверки доступных интерфейсов и их текущей конфигурации. В выводе команды отображаются данные по следующим строкам:

* **DHCP включен**. Эта строка отображает, является ли адрес статическим или использует DHCP-конфигурацию;
* **IP-адрес**. В данной строке вы можете увидеть IP-адрес, настроенный для данного интерфейса;
* **Маска подсети**. Здесь указана маска подсети, которая связана с указанным выше IP-адресом;
* **Основной шлюз**. В этой строке вы можете увидеть IP-адрес основного шлюза для текущего интерфейса;
* **Метрика шлюза**. В этой строке отображается метрика основного шлюза, который указан выше и отображается она только в том случае, если существует несколько основных шлюзов;
* **Метрика интерфейса**. Данная строка показывает метрику текущего интерфейса и также, как и предыдущая строка, отображается только при наличии нескольких сетевых интерфейсов.

## Удаление IPv4 адресов

Контекст Interface IPv4 утилиты Netsh командной строки также позволяет вам удалять некоторые конфигурационные параметры TCP/ IP. Синтаксис у данной команды следующий:

Delete address name=[имя\_сетевого\_подключения] address=[ IPv4-адрес] gateway=[имя\_шлюза]

**ipconfig** — утилита командной строки для управления сетевыми интерфейсами, для вывода деталей текущего соединения и управления клиентскими сервисами DHCP и DNS.

**Примеры вывода для Windows**

C:\>ipconfig /all

Windows 2000 IP Configuration

Host Name . . . . . . . . . . . . : wikipedia

Primary DNS Suffix . . . . . . . :

Node Type . . . . . . . . . . . . : Hybrid

IP Routing Enabled. . . . . . . . : No

WINS Proxy Enabled. . . . . . . . : No

DNS Suffix Search List. . . . . . : wikipedia.org

Ethernet adapter Local Area Connection 2:

Connection-specific DNS Suffix . : wikipedia.org

Description . . . . . . . . . . . : Intel(R) PRO/100 VE Netwon #3

Physical Address. . . . . . . . . : 00-D0-B7-A6-F1-11

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

IP Address. . . . . . . . . . . . : 192.168.0.100

Subnet Mask . . . . . . . . . . . : 255.255.0.0

Default Gateway . . . . . . . . . : 192.168.0.3

DHCP Server . . . . . . . . . . . : 192.168.0.1

DNS Servers . . . . . . . . . . . : 192.168.0.1

Primary WINS Server . . . . . . . : 192.168.0.75

Lease Obtained. . . . . . . . . . : 27 May 2004 09:04:06

Lease Expires . . . . . . . . . . : 30 May 2004 09:04:06

**Ping** — утилита для проверки соединений в сетях на основе TCP/IP.

Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, от англ. Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, то есть косвенно определять загруженность на каналах передачи данных и промежуточных устройствах.

**Пример**

ping ru.wikipedia.org

Обмен пакетами с rr.esams.wikimedia.org [91.198.174.225] по 32 байт:

Ответ от 91.198.174.225: число байт=32 время=54мс TTL=32

Ответ от 91.198.174.225: число байт=32 время=54мс TTL=32

Ответ от 91.198.174.225: число байт=32 время=54мс TTL=32

Ответ от 91.198.174.225: число байт=32 время=54мс TTL=32

Статистика Ping для 91.198.174.225:

Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),

Приблизительное время приема-передачи в мс:

Минимальное = 54 мсек, Максимальное = 54 мсек, Среднее = 54 мсек

**30. Конфигурирование рабочих мест. Протокол DHCP.**

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора, на которые будут приходить и с которых будут отправляться IP-пакеты, должен быть назначен IP-адрес. Назначение IP-адресов может происходить. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие — еще свободны. Протокол Dynamic Host Configuration Protocol (DHCP) освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов. DHCP может поддерживать автоматическое динамическое распределение адресов, а также более простые способы ручного и автоматического статического назначения адресов. Протокол DHCP работает в соответствии с моделью клиент-сервер. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DIICP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры. Предполагается, что DHCP-клиент и DHCP-сервер находятся в одной IP-сети.

При ручной процедуре назначения статических адресов активное участие принимает администратор, который сообщает DHCP-серверу информацию о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному клиенту один и тот же назначенный ему администратором адрес.

При автоматическом статическом способе DHCP-сервер самостоятельно, без вмешательства администратора, выбирает клиенту произвольный IP-адрес из пула наличных IP-адресов. Границы пула задает администратор при конфигурировании DHCP-сервера. Адрес дается клиенту из пула в постоянное пользование, то есть с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдаст адрес клиенту на ограниченное время, называемое временем аренды (lease duration), что дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества DHCP — автоматизации рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере, динамическое разделение адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

Рассмотрим работу протокола DHCP в ситуации, когда компьютер, являющийся DHCP-клиентом, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

DHCP-сервер может назначить клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например, маску, IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п.

**31. Групповое вещание в компьютерных сетях. Адресация при групповом вещании. Протоколы IGMP, DVMRP, MOSPF, PIM-SM. Понятие реверсивной маршрутизации.**  Групповое вещание, то есть доставка данных из одного источника сразу нескольким получателям, давно доказала свою полезность и необходимость в мире коммуникаций. Ведущие производители сетевого оборудования и программных средств стремятся встроить поддержку группового вещания в свои продукты — маршрутизаторы, коммутаторы, операционные системы.

# Стандартная модель группового вещания IP

Основной целью группового вещания является создание эффективного механизма передачи данных от одного источника нескольким получателям. Для решения этой задачи могут использоваться несколько подходов, например индивидуальная рассылка, широковещательная рассылка, привлечение сервисов прикладного уровня.

То есть передача по принципу **«один ко многим»** сводится к нескольким передачам «один к одному». Очевидно, что передача нескольких идентичных копий на участках, где маршруты к разным членам группы перекрываются (это особенно характерно для начальных участков), приводит к избыточному трафику.

При **широковещательной рассылке** (broadcast) станция направляет пакеты, используя широковещательные адреса. В этой схеме, для того чтобы доставить данные группе узлов-получателей, источник генерирует один экземпляр данных, но снабжает этот экземпляр широковещательным адресом, который диктует маршрутизаторам сети копировать эти данные и рассылать их всем конечным узлам независимо от того, «заинтересованы» узлы в получении этих данных или нет. В этом случае, как и в предыдущем, существенная доля трафика является избыточной.

В случае привлечения **сервисов прикладного уровня** функции по обеспечению групповой доставки перекладываются на самих членов группы. Источник генерирует один экземпляр данных и, используя индивидуальный адрес, передает данные одному из членов группы, который генерирует копию и направляет ее другому члену группы и т. д. Перемещение решения задачи с нижних транспортных уровней на прикладной уровень повышает суммарные накладные расходы сети на реализацию групповой доставки и делает этот механизм менее гибким.

Таким образом, традиционные Механизмы доставки пакетов стека TCP/IP мало пригодны для поддержки группового вещания. В такой ситуации наиболее эффективным решением является использование специально разработанного механизма группового вещания, ориентированного на сокращение избыточного трафика и накладных расходов сети.

Главная идея группового вещания состоит в следующем: источник генерирует только один экземпляр сообщения с групповым адресом, которое затем, по мере перемещения по сети, копируется на каждой из «развилок», ведущих к тому или иному члену группы, указанной в адресе данного сообщения. При таком подходе данные рассылаются только тем узлам, которые заинтересованы в их получении.

Есть несколько принципиальных положений, регламентирующих поведение конечных узлов сети, которые являются источниками и получателями группового трафика.

* Дейтаграммный подход. Источник может посылать пакеты UDP/IP в любое время без необходимости регистрировать или планировать передачи, реализуя сервис «по возможности».
* Открытые группы. Источники должны знать только групповой адрес. Они не должны знать членов группы и не обязательно должны быть членами той группы, которой они посылают данные. Группа может быть образована узлами, принадлежащими к разным IP-сетям и подсетям. Группа может иметь любое число источников данных.
* Динамические группы. Хосты могут присоединяться к группам или покидать группы без необходимости регистрации, синхронизации или переговоров с каким-либо централизованным элементом группового управления. Членство в группе является динамическим, поскольку хосты могут присоединиться к группе или выйти из группы в любой момент времени, к тому же они могут быть членами нескольких групп.

Из концепции открытых групп следует, что групповое вещание может быть организовано как по схеме «один ко многим», так и по схеме «многие ко многим».

В более поздней модели, называемой групповым вещанием из конкретного источника (Source Specific Multicast, SSM), хосты могут регистрировать свою заинтересованность не только относительно определенной группы, указывая соответствующий групповой адрес, но и в отношении совершенно определенных источников группового трафика, указывая соответствующие индивидуальные адреса. Возможность запроса конкретных источников является ключевой в модели SSM. Модель сервиса группового вещания SSM строится по схеме «один ко многим» и предусматривает возможность работы хостов в двух дополнительных режимах:

* в режиме исключения хост может требовать, чтобы ему направлялись пакеты для его группы, но только те, которые поступают от источников, не входящих в его список исключенных источников;
* в режиме включения хост может требовать получение группового трафика только от тех источников, которые перечислены в списке включенных источников.

# Адреса группового вещания

Ранее в главе, изучая типы IP-адресов, мы отмечали, что адреса IPv4 из диапазона 224.0.0.0-239.255.255.255 относятся к классу D и они зарезервированы для группового вещания.

Адреса из этого диапазона используются:

* для идентификации групп;
* для идентификации адресов источников группового вещания (в рамках модели SSM);
* для административных нужд при реализации группового вещания.

В общем случае адреса используются динамически, то есть если после остановки вещания источник снова начинает передачу, то он в общем случае может задействовать новый адрес группового вещания. Так называемые хорошо известные источники обычно наделяются постоянным групповым адресом.

# Основные типы протоколов группового вещания

На основе описанной концепции для стека TCP/IP был разработан ряд протоколов, с помощью которых можно организовать групповое вещание с различной степенью эффективности. Эти протоколы делятся на две категории.

* В первую входит один протокол — протокол IGMP, с помощью которого, во-первых, хосты сообщают о своем «желании» присоединиться к некоторой группе, во-вторых, маршрутизатор узнает о принадлежности хостов в непосредственно подключенных к нему подсетях к той или иной группе. Протокол IGMP работает в тесном взаимодействии с протоколами второй категории — протоколами маршрутизации группового вещания.
* Протоколы маршрутизации группового вещания необходимы для продвижения пакетов, несущих в себе информацию для групповых получателей, через сеть произвольной конфигурации. Эти протоколы — DVMRP, MOSPF, PIM — опираются на разные подходы, но в конечном итоге все они сводятся к построению графа, связывающего все хосты в определенной группе, причем между двумя хостами существует только один путь. Такой граф называют покрывающим деревом. Протоколы маршрутизации осуществляют постоянный мониторинг покрывающего дерева и время от времени отсекают те ветви дерева, которые из-за изменения состояния сети уже не ведут к членам той или иной группы.

**Понятие реверсивной маршрутизации**

Концепция продвижения по реверсивному пути — это еще одна концепция, которую необходимо понять всем, кто реализует групповое вещание. Механизм, используемый для маршрутизации трафика группового вещания, в определенном аспекте является прямо противоположным (реверсивным) тому механизму, который применяется для продвижения обычного трафика на основе индивидуальных адресов.

Традиционная маршрутизация на основе индивидуальных адресов основывается на адресе назначения. То есть маршрутизаторы перемещают пакет с индивидуальным адресом по сети вперед, в направлении приемника.

Напротив, все пакеты с групповым адресом маршрутизаторы тиражируют и передают копии во все стороны — на все интерфейсы, кроме того, с которого этот пакет поступил. При этом в сложных сетях возможно образование петель — замкнутых маршрутов. Для правильной работы сети зациклившиеся пакеты необходимо распознавать и отбрасывать.

Петля не может возникнуть, если ли пакет прибыл от источника по ожидаемому пути, проложенному в соответствии с обычным алгоритмом маршрутизации, основанном на анализе таблиц маршрутизации. А именно, маршрутизатор проверяет, является ли входной интерфейс, получивший групповой пакет, интерфейсом, через который пролегает кратчайший путь к источнику. Он делает это с помощью обычной таблицы маршрутизации, которая, как известно, содержит указания о рациональных путях ко всем сетям составной интерсети.

Проверка факта выполнения данного условия называется продвижением по реверсивному пути (Reverse Path Forwarding, RPF). Только пакеты, которые прошли RPF-проверку, являются кандидатами для дальнейшего продвижения вдоль путей, ведущих к потенциальным получателям трафика группового вещания.

**32. Групповые IP-адреса, групповые MAC-адреса. Построение дерева**

**кратчайшего пути к источнику.**

Для того, чтобы обратиться к нужному хосту (интерфейсу) в сети, Вы можете использовать три различных типа адреса:

* **unicast (однонаправленный) адрес**. Этот тип адреса ссылается на уникальный хост (интерфейс) в подсети. Примером такого адреса может служить 192.168.100.9. Примером же unicast MAC адреса - 80:C0:F6:A0:4A:B1.
* **broadcast (широковещательный) адрес**. Этот адрес позволяет Вам обратиться ко всем хостам (интерфейсам) в подсети. Broadcast IP адрес - 192.168.100.255 и MAC адрес - FF:FF:FF:FF:FF:FF.
* **multicast (групповой) адрес**. Этот тип адреса позволяет обратиться к конкретной группе хостов (интерфейсов) в подсети.

Групповые адреса очень полезны, когда получателем информации является не единичный хост и мы не хотим генерировать широковещательный запрос. Этот сценарий типичен для тех ситуаций в которых требуется посылка мультимедийной информации (например пересылка аудио или видео информации в реальном времени) некоторым хостам. В случае посылки такого рода информации каждому клиенту по unicast адресу у нас просто может не хватить пропускной способности сети. Передача же по широковещательному адресу тоже не лишена недостатков - клиент может находиться за пределами подсети, которая получает информацию.

**Групповые адреса**

Как читателю вероятно известно, пространство IP-адресов поделено на три класса адресов. Классы - A,B и C. Есть еще четвертый класс (D), зарезервированный для групповых адресов. Адреса IPv4 между 224.0.0.0 и 239.255.255.255 принадлежат классу D.

4 старших бита IP-адеса обозначают значения между 224 и 239. Остальные 28 бит, зарезервированы для идентификатора группы.

На сетевом уровне, групповые адреса IPv4 должны отображаться на физические адреса того типа сети с которой мы работаем. Если мы работаем с сетевым unicast-адресом, то мы должны получить соответствующий физический адрес используя протокол ARP. В случае использования нами группового адреса, ARP использовать нельзя, и физический адрес может быть получен другим путём.

**Построение дерева кратчайшего пути к источнику.**

Дерево кратчайшего пути (Shortest Path Tree, SPT) будет доставлять пакеты между источником данной группы и точкой встречи.

**Построение SPT-дерева от источника к точке встречи**.

Когда источник S становится активным и начинает посылать пакеты с групповым адресом в свою локальную сеть, маршрутизатор F, к которому эта сеть непосредственно подключена, замечает, что источник 5 стал источником группового вещания. Маршрутизатор F посылает Р1М-сообщение о регистрации (register) на индивидуальный адрес точки встречи (маршрутизатора D). При этом сообщение о регистрации инкапсулируется в пакет группового вещания от источника 5.

Когда маршрутизатор D (точка встречи) получает сообщение о регистрации, он реагирует на это двумя действиями. Во-первых, он продвигает инкапсулированные данные группового вещания по разделяемому дереву (RPT) от точки встречи до получателя, во-вторых, посылает PIM-сообщение о присоединении назад по направлению к источнику с тем, чтобы создать дерево кратчайшего пути (SPT). Это сообщение передается от одного маршрутизатора к другому, при этом информация о присоединении к группе фиксируется на соответствующих интерфейсах.

Как только дерево кратчайшего пути от источника к точке встречи построено, маршрутизатор D начинает получать по две копии каждого пакета группового вещания. Одна копия приходит от источника 5 по вновь созданному кратчайшему пути, другая — от маршрутизатора F, который, продолжая реагировать на выявленную активность источника 5, снова посылает сообщение о регистрации, в котором в инкапсулированном виде содержится вторая копия группового пакета. Когда маршрутизатор точки встречи распознает эту ситуацию, он посылает маршрутизатору F сообщение с требованием прекратить регистрацию (register stop). Получив это сообщение для данной пары источник-группа, маршрутизатор F прекращает генерировать сообщения о регистрации и инкапсулировать в них групповые пакеты источника. Вместо этого он начинает посылать их в исходном виде с групповым адресом, так как к этому моменту источник уже присоединился к дереву группы, и это присоединение зафиксировано на нужных маршрутизаторах.

**33. Использование группового вещания при построении современных сетей цифрового телевидения. Zala by и IPTV телевидение. OTT телевидение.**

Групповое вещание, то есть доставка данных из одного источника сразу нескольким получателям, давно доказала свою полезность и необходимость в мире коммуникаций. Применяемая ранее только в радио и телевизионных сетях, в последние годы технология группового вещания все шире внедряется в компьютерные сети. В условиях, когда компьютерные сети постепенно становятся средством для передачи практически всех видов информации, без реализации направленного широковещания в них не обойтись.

Поэтому технологии группового вещания являются сегодня предметом интенсивного изучения в исследовательском сетевом сообществе. Ведущие производители сетевого оборудования и программных средств стремятся встроить поддержку группового вещания в свои продукты — маршрутизаторы, коммутаторы, операционные системы.

**IPTV** или **Телевидение по протоколу интернета** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet Protocol Television*) (*IP-TV*, *IP-телевидение*) — технология [цифрового телевидения](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B5_%D1%82%D0%B5%D0%BB%D0%B5%D0%B2%D0%B8%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5) в сетях [передачи данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) по протоколу [IP](https://ru.wikipedia.org/wiki/IP), новое поколение телевидения. В последнее время часто путается с технологией [ОТТ](https://ru.wikipedia.org/wiki/OTT), которая, в свою очередь является подклассом IPTV в области распространения видеоконтента.

В качестве клиентского оборудования могут выступать все компьютеры (соответствующие системным требованиям), специализированные ТВ приставки, [медиа-плееры](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0-%D0%BF%D0%BB%D0%B5%D0%B5%D1%80), телевизоры с технологией SMART TV, мобильные устройства. На программном уровне доступ к ресурсам IPTV может осуществляться как при помощи специальных приложений (программ), так и при помощи обычного интернет-браузера, встроенного в устройство.

Доставка контента до клиентского оборудования осуществляется либо по управляемой IP-сети оператора связи с использованием технологии [multicast](https://ru.wikipedia.org/wiki/Multicast) или [unicast](https://ru.wikipedia.org/wiki/Unicast) (в зависимости от топологии сети), либо без привязки в сетях операторов связи ([ОТТ](https://ru.wikipedia.org/wiki/OTT)).

Главным достоинством IPTV является интерактивность и возможность предоставления пользователям широкого набора дополнительных услуг, связанных с потреблением контента. Возможности протокола [IP](https://ru.wikipedia.org/wiki/IP) позволяют предоставлять не только видеоуслуги, но и гораздо [более широкий пакет услуг, в том числе интерактивных и интегрированных.](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D0%BC%D0%B0%D1%88%D0%BD%D0%B8%D0%B9_%D0%B8%D0%BD%D1%84%D0%BE%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80&action=edit&redlink=1)

Помимо базовых услуг, IPTV может включать ряд дополнительных сервисов ([Video Telephony](https://ru.wikipedia.org/wiki/Video_Telephony), [Voting](https://ru.wikipedia.org/w/index.php?title=Voting&action=edit&redlink=1), [Information Portals](https://ru.wikipedia.org/w/index.php?title=Information_Portal&action=edit&redlink=1), [Web](https://ru.wikipedia.org/wiki/Web), [Games](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0), [MOD KOD](https://ru.wikipedia.org/w/index.php?title=MOD_KOD&action=edit&redlink=1)). Это возможно на основе унификации и стандартизации различных оконечных устройств, интеграции звука, видео и данных на основе [IP-протокола](https://ru.wikipedia.org/wiki/IP-%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) и предоставления услуг на единой технологической платформе.

В IPTV есть возможность использовать для одного видеоряда двух и более каналов звукового сопровождения, например на русском и английском языках.

Преимущество IPTV перед аналоговым кабельным ТВ:

* Изображение и звук обычно качественнее, вплоть до [HD](https://ru.wikipedia.org/wiki/HD)-разрешения и 5.1-канального аудио
* [Интерактивность](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) (возможность просмотреть, например, справку по фильму, оставить отзыв)
* Сервисные возможности timeshift и video-on-demand

**Технология OTT (Over the Top)**  является частью технологии IPTV. Термин OTT означает доставку видеосигнала на приставку (компьютер, мобильный телефон) пользователя по сети Интернет без прямого контакта с оператором связи в отличие от услуг [IPTV](https://ru.wikipedia.org/wiki/IPTV), которые предоставляются через управляемую оператором сеть с гарантированным [QoS](https://ru.wikipedia.org/wiki/QoS) ([QoE](https://ru.wikipedia.org/w/index.php?title=QoE&action=edit&redlink=1)).

Особенности

* предоставление шифрованных каналов с возможностью легальной записи отдельных передач на абонентские устройства пользователей, в том числе по подписке;
* транскодирование контента (каналов и фильмов) в различные форматы для просмотра на компьютере, телевизоре и мобильном телефоне;
* предоставление контента по запросу с возможностью просмотра онлайн и загрузки на абонентское устройство для дальнейшего просмотра.

Особенностью внедрения услуг OTT является возможность предоставить сервис любому пользователю Интернет.

**ZALA — торговая марка Белтелеком для предоставления услуг интерактивного телевидения IP-TV.**

Используемая технология передачи ТВ-сигнала по Интернет-протоколу, помогает реализовать целый ряд ранее недоступных для пользователей функций. Новая услуга сочетает в себе преимущества кабельного ТВ с интерактивностью сети Интернет.

**34. Имена абонентов вычислительной сети. Служба доменных имен**

**Internet (DNS).**

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста, так и средствами централизованной службы. На раннем этапе развития Интернета па каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «IP-адрес — доменное имя», например 102.54.94.97 — rhino.acme.com.

По мере роста Интернета файлы hosts также росли, и создание масштабируемого решения для разрешения имен вышло на первый план. Таким решением стала специальная служба — система доменных имен (Domain Name System, DNS), DNS — это централизованная служба, основанная на распределенной базе отображений «доменное имя — IP-адрес». Служба DNS использует в своей работе протокол типа «клиент-сервер». В нем определены DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес. Служба DNS использует текстовые файлы почти такого формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый сервер службы DNS храпит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. Имеются две системы распределения имен па серверах. В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется другой подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmt.ru будет хранить отображения для всех имен, заканчивающихся на mmt.ru: www1.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д. Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер кроме таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых являются широко известными.

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения. Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяется кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена. Существенным же отличием является то, что файловая система расположена на одном компьютере, а служба DNS по своей природе является распределенной.

Существует две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент.

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.

2. DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени.

3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется нерекурсивной, или итеративной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Так как эта схема загружает клиента достаточно сложной работой, то она применяется редко.

Во втором варианте реализуется рекурсивная процедура.

1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, к которому принадлежит имя клиента.

2. Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше.

3. Если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делает клиент в первом варианте. Получив ответ, он передаст его клиенту, который все это время просто ожидает его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется косвенной, или рекурсивной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют процедуру кэширования проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на определенное время — обычно от нескольких часов до нескольких дней.

**35. Контроль и диагностика в IP-сетях. Протокол ICMP.**

Можно представить ряд ситуаций, когда протокол IP не может доставить пакет адресату, например истекает время жизни пакета, в таблице маршрутизации отсутствует маршрут к заданному в пакете адресу назначения и т. д., и т. п.

Как мы не раз отмечали, протокол IP доставляет данные, руководствуясь принципом «по возможности», то есть не предпринимает мер для гарантированной передачи данных адресату. Это свойство «необязательности» протокола IP компенсируется протоколами более высоких уровней стека TCP/IP, например TCP на транспортном уровне и в какой-то степени DNS на прикладном уровне. Они берут на себя обязанности по обеспечению надежности, применяя такие известные приемы, как нумерация сообщений, подтверждение доставки, повторная посылка данных.

Протокол ICMP также служит дополнением, компенсирующим ненадежность протокола IP, но несколько другого рода. Он не предназначен для исправления возникших при передаче пакета проблем: если пакет потерян, ICMP не может послать его заново. Задача ICMP другая — он является средством оповещения отправителя о «несчастных случаях», произошедших с его пакетами.

Для передачи по сети ICMP-сообщение инкапсулируется в поле данных IP-пакета. IP-адрес узла-источника определяется из заголовка пакета, вызвавшего инцидент.

Сообщение, прибывшее в узел-источник, может быть обработано там либо ядром операционной системы, либо протоколами транспортного и прикладного уровней, либо приложениями, либо просто проигнорированы. Важно, что обработка ICMP-сообщений не входит в обязанности протоколов IP и ICMP.

Заметим, что некоторые из пакетов могут исчезнуть в сети, не вызвав при этом никаких оповещений. В частности, протокол ICMP не предусматривает передачу сообщений о проблемах, возникающих при обработке IP-пакетов, несущих ICMP-сообщения об ошибках. Такое решение было принято разработчиками протокола, чтобы не порождать «штормы» в сетях, когда количество сообщений об ошибках лавинообразно возрастает.

Все типы сообщений имеют один и тот же формат, однако интерпретация полей существенно зависит от того, к какому типу относится сообщение.

Заголовок ICMP-сообщения состоит из:

* тип — числовой идентификатор типа сообщения;
* код — числовой идентификатор, более тонко дифференцирующий тип ошибки;
* контрольная сумма — подсчитывается для всего ICMP-сообщения.

Сообщения можно разделить на две группы:

* сообщения об ошибках;
* сообщения запрос-ответ.

Сообщения типа запрос-ответ связаны в пары: эхо-запрос — эхо-ответ, запрос маски -ответ маски, запрос времени — ответ времени. Отправитель сообщения-запроса всегда рассчитывает на получение соответствующего сообщения-ответа.

Сообщения, относящиеся к группе сообщений об ошибках, конкретизируются уточняющим кодом ошибки.

**36. Управление вычислительной сетью. Протокол и служба SNMP.**

SNMP (англ. Simple Network Management Protocol — простой протокол сетевого управления) — стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. К поддерживающим SNMP устройствам относятся маршрутизаторы, коммутаторы, серверы, рабочие станции, принтеры и другие. Протокол обычно используется в системах сетевого управления для контроля подключенных к сети устройств на предмет условий, которые требуют внимания администратора. SNMP состоит из набора стандартов для сетевого управления, включая протокол прикладного уровня, схему баз данных и набор объектов данных.

SNMP предоставляет данные для управления в виде переменных, описывающих конфигурацию управляемой системы. Эти переменные могут быть запрошены (а иногда и заданы) управляющими приложениями.

На каждой управляемой системе есть постоянно запущенная программа, называемая *агент*, которая через SNMP передаёт информацию *менеджеру*.

*Менеджеры* SNMP обрабатывают данные о конфигурации и функционировании управляемых систем и преобразуют их во внутренний формат, удобный для поддержания протокола SNMP. Протокол также разрешает активные задачи управления, например, изменение и применение новой конфигурации через удаленное изменение этих переменных. Доступные через SNMP переменные организованы в иерархии.

Управляемые протоколом SNMP сети состоят из трех ключевых компонентов:

* *Управляемое устройство*;
* *Агент* — программное обеспечение, запускаемое на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства;
* *Система сетевого управления* (Network Management System, NMS) — программное обеспечение, взаимодействующее с менеджерами для поддержки комплексной структуры данных, отражающей состояние сети[[1]](https://ru.wikipedia.org/wiki/SNMP#cite_note-NMS-1).

*Управляемое устройство* — элемент сети (оборудование или программное средство), реализующий интерфейс управления (не обязательно SNMP), который разрешает однонаправленный (только для чтения) или двунаправленный доступ к конкретной информации об элементе. Управляемые устройства обмениваются этой информацией с менеджером. Управляемые устройства могут относиться к любому виду устройств: маршрутизаторы, серверы доступа, коммутаторы, мосты, концентраторы, IP-телефоны, IP-видеокамеры, компьютеры-хосты, принтеры и т.п.

*Агентом* называется программный модуль сетевого управления, располагающийся на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства. Агент обладает локальным знанием управляющей информации и переводит эту информацию в специфичную для SNMP форму или из неё (медиация данных).

В состав *Системы сетевого управления* (*NMS*) входит приложение, отслеживающее и контролирующее управляемые устройства. NMS обеспечивают основную часть обработки данных, необходимых для сетевого управления. В любой управляемой сети может быть одна и более NMS.

## Детали протокола

SNMP работает на прикладном уровне TCP/IP (седьмой уровень модели OSI). Менеджер может посылать запросы с любого доступного порта источника на порт агента. Ответ агента будет отправлен назад на порт источника на менеджере. Агент может генерировать уведомления с любого доступного порта.

Ниже перечислены основные протокольные единицы обмена SNMP:

### GetRequest

Запрос от менеджера к объекту для получения значения переменной или списка переменных. Требуемые переменные указываются в поле variable bindings (раздел поля values при этом не используется). Получение значений указанной переменной должно быть выполнено агентом как [Атомарная операция](https://ru.wikipedia.org/wiki/%D0%90%D1%82%D0%BE%D0%BC%D0%B0%D1%80%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F). Менеджеру будет возвращен Response (ответ) с текущими значениями.

### SetRequest

Запрос от менеджера к объекту для изменения переменной или списка переменных. Связанные переменные указываются в теле запроса. Изменения всех указанных переменных должны быть выполнены агентом как атомарная операция. Менеджеру будет возвращен Response с (текущими) новыми значениями переменных.

### Response

Возвращает связанные переменные и значения от агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest и InformRequest. Уведомления об ошибках обеспечиваются полями статуса ошибки и индекса ошибки. Хотя эта единица использовалась как ответ и на get-, и на set-запросы, она была названа GetResponse в SNMPv1.

**Служба SNMP**

Служба SNMP (Simple Network Management Protocol) поддерживает компьютеры, использующие протоколы TCP/IP и IPX. Эта служба является необязательной и может быть установлена после успешной настройки протокола TCP/IP.

Служба SNMP обеспечивает работу агента SNMP, делающего возможным удаленное централизованное управление компьютерами

**37. Маршрутизация в вычислительных сетях: задачи, проблемы,**

**принципы, критерии эффективности.**

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут — это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до получателя. Так, пакет, отправленный из узла А в узел В, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17, 13, 7, 6 и 3.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная

способность маршрута для последовательности пакетов. Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).

Чтобы по адресу сети назначения можно было выбрать рациональный маршрут

дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется таблицей маршрутизации.

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
| N1 | IP12(R1) | IP41 | 1 |
| N2 | - | IP41 | 0(подсоединена) |
| Маршрут по умолчанию | IP51 (R5) | IP42 | - |

В первом столбце приведенной выше таблицы перечисляются номера сетей,

входящих и интерсеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес соответствующего порта следующего маршрутизатора, на который надо

направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлеченный из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на время ее просмотра, потребует много места для хранения и т. п. Поэтому на практике число записей в таблице маршрутизации стараются уменьшить за счет использования специальной записи — «маршрутизатор по умолчанию» {default}. Действительно, если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединенных к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах.

Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например Default.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы

маршрутизации. В этом случае при выборе маршрута принимается во внимание столбец «Расстояние до сети назначения». Расстояние может измеряться хопами, временем прохождения пакета по линиям связи или другой величиной, отражающей качество данного маршрута по отношению к заданному критерию.

В табл. 11.1 расстояние между сетями измерялось хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0, однако в некоторых реализациях отсчет расстояний начинается с 1.

Наличие нескольких маршрутов к одному узлу делают возможным передачу трафика к этому узлу параллельно по нескольким каналам связи, что повышает пропускную способность и надежность сети.

Задачу маршрутизации решают не только промежуточные узлы-маршрутизаторы, но и конечные узлы — компьютеры. Средства сетевого уровня, установленные на конечном узле, при обработке пакета должны, прежде всего, определить, направляется ли он и другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то пакету не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант — единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, задание маршрута по умолчанию часто используется в компьютерах для сокращения объема их таблицы маршрутизации.

Еще одним отличием работы маршрутизатора и конечного узла при выборе маршрута является способ построения таблицы маршрутизации. Если маршрутизаторы обычно автоматически создают таблицы маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в виде постоянных файлов на дисках.

**38. Виды маршрутизации в вычислительных сетях.**

**Маршрутизация** — процесс определения лучшего пути, по которому пакет может быть доставлен получателю.

Возможные пути передачи пакетов называются [маршрутами](http://xgu.ru/w/index.php?title=%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82&action=edit&redlink=1). Лучшие маршруты к известным получателям хранятся в [таблице маршрутизации](http://xgu.ru/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_%D0%BC%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8).

В зависимости от способа заполнения таблицы маршрутизации, различают два вида маршрутизации:

* [Статическая маршрутизация](http://xgu.ru/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)
* [Динамическая маршрутизация](http://xgu.ru/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)

Устройство, выполняющее маршрутизацию, именуется *маршрутизатором*. В качестве маршрутизатора могут использоваться специализированные устройства или это могут быть обычные компьютеры, оснащённые несколькими сетевыми картами.

**Статическая маршрутизация** — вид [маршрутизации](http://xgu.ru/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), при котором маршруты вручную указываются администратором при настройке маршрутизатора.

К преимуществам статической маршрутизации можно отнести:

* Простоту настройки (в небольших сетях),
* Отсутствие дополнительной нагрузки на сеть (в отличии от динамических протоколов маршрутизации).

К недостаткам относится:

* Сложность масштабирования,
* При возникновении каких-либо изменений в сети, как правило потребуется вмешательство администратора и настройка новых, актуальных статических маршрутов,

**Динамическая маршрутизация** — вид маршрутизации, при котором таблица маршрутизации редактируется программно. В случае UNIX-систем *демонами маршрутизации*; в других системах — служебными программами, которые называются иначе, но фактически играют ту же роль.

Демоны маршрутизации обмениваются между собой информацией, которая позволяет им заполнить таблицу маршрутизации наиболее оптимальными маршрутами. Протоколы, с помощью которых производится обмен информацией между демонами, называется

**Бесклассовая междудоменная маршрутизация(Classless Inter-Domain Routing)**

Бесклассовая междудоменная маршрутизация - метод маршрутизации, используемый для увеличения количества подсетей, соответствующих заданной длине адреса.

**Динамическая адаптивная маршрутизация(Dynamic adaptive routing)**

Динамическая адаптивная маршрутизация - метод выбора маршрута в сетях с коммутацией каналов, учитывающий динамическое состояние:

- выходных трактов узла (локальная адаптивная маршрутизация);

или

- сети (глобальная адаптивная маршрутизация).

**Динамическая маршрутизация(Dynamic routing)**

Динамическая маршрутизация использует специальные алгоритмы маршрутизации:

- алгоритм "вектор-длина";

- алгоритм "состояние канала" и др.

**Иерархическая маршрутизация**

Иерархическая маршрутизация - метод маршрутизации, базирующийся на схеме разбивки большой сети на иерархическую систему подсетей с собственной маршрутизацией внутри каждого уровня.

**Лавинная маршрутизация**

Лавинная маршрутизация - метод маршрутизации пакетов, при котором информация, полученная маршрутизатором, передается во все подключенные к нему порты, за исключением того, откуда она получена.

**Распределенная маршрутизация**

Распределенная маршрутизация - метод маршрутизации, в котором каждый узел автономно принимает решение о выборе маршрута.

**Статическая маршрутизация (Static routing)**

Статическая маршрутизация - метод маршрутизации в сетях с коммутацией пакетов, при котором данные передаются по предопределенному пути и не зависит от колебаний трафика. Статическая маршрутизация основывается на изменяемых вручную таблицах маршрутизации. При задании статической маршрутизации должны быть заданы все взаимосвязи между логическими сетями, которые остаются неизменными.

**Централизованная маршрутизация (Centralized routing)**

Централизованная маршрутизация - метод маршрутизации пакетов или сообщений в сетях передачи данных, при котором решение о выборе маршрута принимается центром управления и сообщается всем узлам, находящимся на данном маршруте.

**39. Алгоритмы и протоколы маршрутизации.**

Задача маршрутизации решается на основе анализа таблиц маршрутизации, размещенных во всех маршрутизаторах и конечных узлах сети. Основная работа по созданию таблиц маршрутизации выполняется автоматически, но и возможность вручную скорректировать или дополнить таблицу тоже, как правило, предусматривается.

Для автоматического построения таблиц маршрутизации маршрутизаторы обмениваются информацией о топологии составной сети в соответствии со специальным служебным протоколом. Протоколы этого типа называются протоколами маршрутизации (или маршрутизирующими протоколами). Протоколы маршрутизации (например, RIP, OSPF, NLSP) следует отличать от собственно сетевых протоколов (например, IP, IPX). Те и другие выполняют функции сетевого уровня модели OSI — участвуют в доставке пакетов адресату через разнородную составную сеть. Но в то время как первые собирают и передают по сети чисто служебную информацию, вторые предназначены для передачи пользовательских данных, как это делают протоколы канального уровня. Протоколы маршрутизации используют сетевые протоколы как транспортное средство. При обмене маршрутной информацией пакеты протокола маршрутизации помещаются в поле данных пакетов сетевого уровня или даже транспортного уровня.

В том, что маршрутизаторы для принятия решения о продвижении пакета обращаются к адресным таблицам, можно увидеть их некоторое сходство с мостами и коммутаторами. Однако природа используемых ими адресных таблиц значительно различается. Вместо МАС-адресов в таблицах маршрутизации указываются номера сетей, которые соединяются в интерсеть. Другим отличием таблиц маршрутизации от адресных таблиц мостов является способ их создания. В то время как мост строит таблицу, пассивно наблюдая за проходящими через него информационными кадрами, посылаемыми конечными узлами сети друг другу, маршрутизаторы по своей инициативе обмениваются специальными служебными пакетами, сообщая соседям об известных им сетях в интерсети, маршрутизаторах и о связях этих сетей с маршрутизаторами. Обычно учитывается не только топология связей, но и их пропускная способность и состояние. Это позволяет маршрутизаторам быстрее адаптироваться к изменениям конфигурации сети, а также правильно передавать пакеты в сетях с произвольной топологией, допускающей наличие замкнутых контуров.

С помощью протоколов маршрутизации маршрутизаторы составляют карту связей сети той или иной степени подробности. На основании этой информации для каждого номера сети принимается решение о том, какому следующему маршрутизатору надо передавать пакеты, направляемые в эту сеть, чтобы маршрут оказался рациональным. Результаты этих решений заносятся и таблицу маршрутизации. При изменении конфигурации сети некоторые записи в таблице становятся недействительными. В таких случаях пакеты, отправленные по ложным маршрутам, могут «зацикливаться» и теряться. От того, насколько быстро протокол маршрутизации приводит и соответствие содержимое таблицы реальному состоянию сети, зависит качество работы всей сети

## Общий алгоритм выбора оптимального маршрута

Общий алгоритм выбора оптимального маршрута на основе метрик сегментов пути сформулировал американский математик Дейкстра. Описание алгоритма выбора маршрута представлено ниже:

Извлечь [IP](http://wiki.mvtom.ru/index.php/IP)-адрес (ID) места назначения из дейтограммы.

Вычислить [IP](http://wiki.mvtom.ru/index.php/IP)-адрес сети назначения (IN)

IF IN соответствует какому-либо адресу локальной сети, послать дейтограмму по этому адресу;

else if IN присутствует в маршрутной таблице, то послать дейтограмму к серверу, указанному в таблице;

else if описан маршрут по умолчанию, то послать дейтограмму к этому серверу;

else выдать сообщение об ошибке маршрутизации

**40. Верхние уровни модели OSI. Сеансовый уровень. Уровень представления.**

**Сеансовый уровень** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Session layer*) — 5-й уровень [сетевой модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI), отвечает за поддержание [сеанса связи](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D0%B0%D0%BD%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)&action=edit&redlink=1), позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.

Сеансы передачи составляются из запросов и ответов, которые осуществляются между приложениями. Службы сеансового уровня обычно используются в средах приложений, в которых требуется использование [удалённого вызова процедур](https://ru.wikipedia.org/wiki/%D0%A3%D0%B4%D0%B0%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B2%D1%8B%D0%B7%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80).

В рамках семантических конструкций сеансового уровня сетевой архитектуры OSI этот уровень отвечает на служебные запросы с [представительского уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C) и осуществляет служебные запросы к [транспортному уровню](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C).

Примером применения является организация [видеоконференций в сети](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D1%81%D0%B5%D0%BC%D0%B8%D0%BD%D0%B0%D1%80), когда звуковой и видео потоки должны быть синхронизированы для избежания проблем с[синхронизацией движения губ с речью](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B8%D0%BD%D1%85%D1%80%D0%BE%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B3%D1%83%D0%B1_%D1%81_%D1%80%D0%B5%D1%87%D1%8C%D1%8E&action=edit&redlink=1). [Управление правами на участие в разговоре](https://en.wikipedia.org/wiki/Floor_control) гарантирует, что тот, кто показывается на экране, действительно является собеседником, который в данный момент говорит.

**Уровень представления** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Presentation layer*) — шестой уровень [сетевой модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).

Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

На представительском уровне передаваемая по сети информация не меняет содержания. С помощью средств, реализованных на данном уровне, протоколы прикладных программ преодолевают синтаксические различия в представляемых данных или же различия в кодах символов.

**41. Прикладной уровень. Протоколы SSH, TLS, IRC.**

**Прикладной уровень** (Application layer) — это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или Гипертекстовые web-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

**Протокол прикладного уровня** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Application layer*) — протокол верхнего (7-го) уровня [сетевой модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI), обеспечивает взаимодействие сети и пользователя. Уровень разрешает приложениям пользователя иметь доступ к сетевым службам, таким, как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к [уровню представления](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C).

Пример: [HTTP](https://ru.wikipedia.org/wiki/HTTP),[POP3](https://ru.wikipedia.org/wiki/Post_Office_Protocol), [SMTP](https://ru.wikipedia.org/wiki/SMTP).

**SSH** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***S****ecure****Sh****ell* — «безопасная оболочка») — [сетевой протокол прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F), позволяющий производить [удалённое управление](https://ru.wikipedia.org/wiki/%D0%A3%D0%B4%D0%B0%D0%BB%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) [операционной системой](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и [туннелирование](https://ru.wikipedia.org/wiki/%D0%A2%D1%83%D0%BD%D0%BD%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8)) [TCP](https://ru.wikipedia.org/wiki/TCP)(один из основных [протоколов передачи данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) Интернета, предназначенный для управления [передачей данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) в сетях и подсетях [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP). Выполняет функции протокола транспортного уровня в [стеке протоколов IP](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F).)-соединений (например, для передачи файлов). Схож по функциональности с протоколами [Telnet](https://ru.wikipedia.org/wiki/Telnet) и [rlogin](https://ru.wikipedia.org/wiki/Rlogin), но, в отличие от них, [шифрует](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) весь [трафик](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA), включая и передаваемые [пароли](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D0%BB%D1%8C).

SSH позволяет безопасно передавать в незащищённой среде практически любой другой [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB). Таким образом, можно не только удалённо работать на компьютере через [командную оболочку](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D1%87%D0%BA%D0%B0), но и передавать по шифрованному каналу звуковой поток или видео (например, с [веб-камеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BA%D0%B0%D0%BC%D0%B5%D1%80%D1%8B)). Также SSH может использовать [сжатие](https://ru.wikipedia.org/wiki/%D0%A1%D0%B6%D0%B0%D1%82%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) передаваемых данных для последующего их шифрования.

**TLS** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Transport Layer Security* — безопасность транспортного уровня)— [криптографический протокол](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), обеспечивающие защищённую передачу данных между узлами в сети [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). TLS использует [асимметричную криптографию](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) для аутентификации, [симметричное шифрование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) для конфиденциальности и [коды аутентичности сообщений](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%BE%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0) для сохранения целостности сообщений.

Данный протокол широко используется в приложениях, работающих с сетью [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), таких как [веб-браузеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), [работа с электронной почтой](https://ru.wikipedia.org/wiki/E-mail), [обмен мгновенными сообщениями](https://ru.wikipedia.org/wiki/IM) и [IP-телефония (VoIP)](https://ru.wikipedia.org/wiki/VoIP)

**IRC** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet Relay Chat*) — протокол прикладного уровня для обмена сообщениями в режиме реального времени. Разработан в основном для группового общения. IRC использует транспортный протокол TCP и криптографический TLS (опционально).

**42. Служба передачи файлов в вычислительной сети (FTP, TFTP).**

**FTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *File Transfer Protocol* — [протокол передачи](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) файлов) — стандартный [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), предназначенный для передачи файлов по TCP-сетям (например, Интернет). Использует 21й порт. FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3).

Протокол построен на архитектуре «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль [открытым текстом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол [SSH](https://ru.wikipedia.org/wiki/SSH) для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

**TFTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Trivial File Transfer Protocol* — простой [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) передачи [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB)) используется главным образом для первоначальной [загрузки](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%B0) [бездисковых рабочих станций](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%B4%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%80%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F). TFTP, в отличие от [FTP](https://ru.wikipedia.org/wiki/FTP), не содержит возможностей [аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F)(хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе [UDP](https://ru.wikipedia.org/wiki/UDP).

Основное назначение TFTP — обеспечение простоты реализации клиента. В связи с этим он используется для загрузки бездисковых рабочих станций, загрузки обновлений и конфигураций в «умные» сетевые устройства, записи статистики с [мини-АТС](https://ru.wikipedia.org/wiki/%D0%9E%D1%84%D0%B8%D1%81%D0%BD%D0%B0%D1%8F_%D0%90%D0%A2%D0%A1) ([CDR](https://ru.wikipedia.org/wiki/Call_Detail_Record)) и аппаратных [маршрутизаторов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80)/[файрволов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D1%80%D0%B2%D0%BE%D0%BB).

Поскольку протокол не поддерживает аутентификации, единственный метод идентификации клиента — это его сетевой адрес (который может быть подделан).

Windows NT поддерживает протокол передачи файлов (FTP) и упрощенный протокол (TFTP) в его реализации протокола TCP/IP. Оба эти протоколы могут использоваться для передачи файлов через Интернет. Ниже описаны различия между двумя протоколами:

* FTP может использоваться интерактивно. TFTP позволяет однонаправленной передачи файлов.
* FTP, зависит от TCP, ориентирован на установление соединения и обеспечивает надежное управление. TFTP зависит от UDP, требует меньше издержек и предоставляет практически элемент управления.
* Протокол FTP обеспечивает проверку подлинности пользователей. TFTP — нет.

**43. Электронная почта: принципы организации, протоколы (SMTP, POP, IMAP).**

Самым простым решением по организации обмена сообщениями напрашивается следующее. Допустим, что пользователь А имеет подключенный к некоторой сети передачи данных (СПД) Хост А. И пользователь Б с Хостом Б является абонентом той же сети. Тогда для передачи сообщений от А к Б необходимо организовать протокол обмена сообщениями (или просто, почтой) и этот протокол должен быть реализован на обеих хостах. Предположим, что СПД не предоставляет напрямую такого протокола, тогда на А и Б ложится вся нагрузка по приему/передаче почты. А что если пользователь является абонентом нескольких сетей? Тогда на его компьютере должно будет установлено все необходимое программное обеспечение, реализующее всевозможные протоколы. Нагрузка возрастет, если помимо программных интерфейсов необходимо будет устанавливать аппаратные.

Таким образом, для надежной доставки сообщений был разработан подход, в котором Электронная почта во многом похожа на обычную почтовую службу. Корреспонденция подготавливается пользователем на своем рабочем месте либо программой подготовки почты, либо просто обычным текстовым редактором. Затем пользователь должен вызвать программу отправки почты (как правило, программа подготовки почты вызывает программу отправки автоматически). Стандартной программой отправки на Unix системах является программа [**sendmail**](http://www.icmm.ru/%7Emasich/win/lexion/mail/sendmail.html). Sendmail работает как почтовый курьер, который доставляет обычную почту в отделение связи для дальнейшей рассылки. В Unix-системах sendmail сама является отделением связи. Она сортирует почту и рассылает ее адресатам.

Для передачи электронной почты в Internet разработан специальный протокол **Simple Mail Transfer Protocol (**[**SMTP**](http://www.icmm.ru/%7Emasich/win/lexion/mail/smtp.html)**)**, который является протоколом прикладного уровня и использует транспортный протокол **TCP**. Однако совместно с этим протоколом используется и **Unix-Unix-Copy (UUCP)** протокол. UUCP хорошо подходит для использования телефонных линий связи. Разница между SMTP и UUCP заключается в том, что при использовании первого протокола sendmail пытается найти машину-получателя почты и установить с ней взаимодействие в режиме on-line для того, чтобы передать почту в ее почтовый ящик. В случае использования SMTP почта достигает почтового ящика получателя за считанные минуты и время получения сообщения зависит только от того, как часто получатель просматривает свой почтовый ящик.

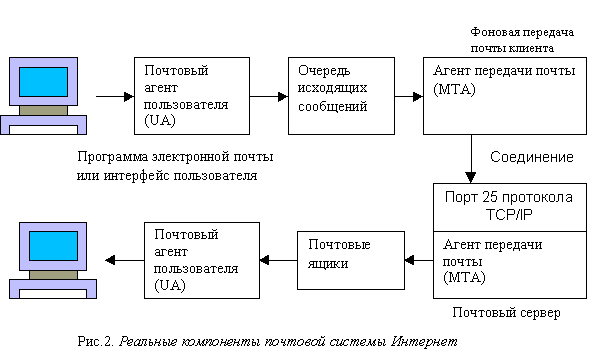
### Форматы почтовых адресов.

Основой любой почтовой службы является система адресов. Без точного адреса невозможно доставить почту адресату. В Internet принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Например, для пользователя paul машины с адресом polyn.net.kiae.su почтовый адрес будет выглядеть как:

paul@polyn.net.kiae.su

Такой тип адреса называется маршрутонезависимым (routeless). Т.е. в адресе нет прямого указания, через какие машины будет лежать маршрут почтового сообщения.

**Компоненты электронной почты Интернет**

На рис. 2 приведены компоненты системы электронной почты в рамках сети Internet. 

Термин "агент" довольно часто встречается в документации Интернет. "Агент" - это программа специального назначения, выполняющая действия для пользователя или другой программы. В большинстве случаев почтовая программа называется [агентом пользователя](http://www.icmm.ru/%7Emasich/win/lexion/mail/iface.html) (User Agent, UA). Точно так же [агент передачи почты](http://www.icmm.ru/%7Emasich/win/lexion/mail/sendmail.html) (Message Transfer Agent, МТА) представляет собой клиент или сервер, выполняющий задачи по доставке или получению почты на сетевом компьютере.

В принципе, пользовательский агент отделен от агента передачи почты. Конечно, их можно объединить в одной программе, но все равно это будут отдельные логические модули. Будучи взаимосвязаны, оба агента выполняют совершенно различные функции. Для пользователей Windows самой известной программой, возможно, является **Microsoft OutLook Express**. Все эти программы - агенты пользователя. Каждая обеспечивает интерфейс между пользователем и системой электронной почты Интернет.

Система электронной почты представлена агентами передачи почты, МТА. До того как обсудить задачи пользовательского агента, необходимо узнать немного больше о том, что же такое МТА. МТА умеют устанавливать соединение для связи с другими МТА. Протоколом этого соединения, как правило, является простой протокол передачи почты (Simple Message Transfer Protocol, [SMTP](http://www.icmm.ru/%7Emasich/win/lexion/mail/smtp.html)).

**Протоколы IMAP и POP**

IMAP и POP — это протоколы, которые загружают почту на пользовательские компьютеры, когда те подключаются к сети. Данные протоколы являются идеальным средством управления почтой, особенно для компьютеров, которые не всегда подключены к сети из-за того, что выключаются после работы или размещены дома и соединяются с сетью по телефонной линии.

Протокол IMAP (Internet Message Access Protocol — протокол доступа к сообщениям в сети Internet) является более предпочтительным средством. Он доставляет сообщения по одному, а не все сразу. Такой метод лучше для тех пользователей, которые подключены к медленной сети или имеют портативные компьютеры. Протокол IMAP управляет почтовыми папками в нескольких системах, например на почтовом сервере и персональном компьютере.

POP (Post Office Protocol — почтовый протокол) — это протокол реализующий похожую модель, в которой, однако, все сообщения загружаются с сервера на персональный компьютер. При этом они могут удаляться с сервера (зачастую безвозвратно) или сохраняться на нем (в этом случае хранилище почты будет постоянно увеличиваться в размерах). Метод доставки всей почты за раз не оптимален для сети и не очень гибок с точки зрения пользователя. Это особенно проявляется в коммутируемых соединениях.

## Протокол SMTP

Главной целью протокола Simple Mail Transfer Protocol (SMTP) является надежная и эффективная доставка электронных почтовых сообщений. SMTP - это довольно независимая субсистема, требующая только надежного канала связи. Средой для SMTP может служить отдельная локальная сеть, система сетей или вся сеть Internet.

Протокол SMTP базируется на следующей модели коммуникаций: в ответ на запрос пользователя почтовая программа-отправитель устанавливает двухстороннюю связь с программой-приемником . Получателем может быть оконечный или промежуточный адресат. SMTP-eiiaiau генерируются отправителем и посылаются получателю. Для каждой команды должен быть получен отклик.

Когда канал организован, отправитель посылает команду MAIL, идентифицируя себя. Если получатель готов к приему сообщения, он посылает положительный отклик. Далее отправитель посылает команду RCPT, идентифицируя получателя почтового сообщения. Если получатель может принять сообщение для оконечного адресата, он снова выдает положительный отклик. В противном случае он отвергает получение сообщения для данного адресата, но не вообще почтовой посылки.

SMTP-сервера могут вести диалог с несколькими оконечными пользователями. Любое почтовое сообщение завершается специальной последовательностью символов. Если получатель успешно завершил прием и обработку почтового сообщения, он посылает положительный отклик.

Протокол SMTP обеспечивает передачу почтового сообщения непосредственно конечному получателю, когда они соединены друг с другом. В противном случае пересылка может выполняться через одну (или более) промежуточную “почтовую станцию”.

Для решения поставленной задачи SMTP-сервер должен знать имя конечного получателя и название почтового ящика места назначения. Аргументом команды MAIL является адрес отправителя (обратный адрес); аргументом команды RCPT - адрес конечного получателя. Обратный адрес используется для посылки сообщения в случае ошибки.

Все отклики имеют цифровые коды. Команды, отклики и имена ЭВМ не чувствительны к тому, строчные или прописные символы использованы при их написании. Это не относится к написанию имен и адресов получателя.

SMTP имеет команды для проверки корректности имени адресата (VRFY) и расширения списка адресов (EXPN). Обе команды в качестве аргументов используют строки символов (в некоторых реализациях эти две команды по своей функции идентичны). Для команды VRFY параметром является имя пользователя, а отклик может содержать его полное имя и адрес его почтового ящика.

**44. Протоколы передачи гипертекстовой информации (HTTP, HTTPS).**

**HTTP** — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам).  
  
Аббревиатура HTTP расшифровывается как *HyperText Transfer Protocol*, «протокол передачи гипертекста». В соответствии со спецификацией [OSI](http://en.wikipedia.org/wiki/OSI_model), HTTP является протоколом прикладного (верхнего, 7-го) уровня.   
Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.  
  
Задача, которая традиционно решается с помощью протокола HTTP — обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам (обычно это веб-браузер) и веб-сервером. На данный момент именно благодаря протоколу HTTP обеспечивается работа Всемирной паутины. Как правило, передача данных по протоколу HTTP осуществляется через TCP/IP-соединения.

Сам по себе протокол HTTP не предполагает использование шифрования для передачи информации. Тем не менее, для HTTP есть распространённое расширение, которое реализует упаковку передаваемых данных в криптографический протокол **SSL** или **TLS**.  
  
Название этого расширения — **HTTPS** (*HyperText Transfer Protocol Secure*). HTTPS широко используется для защиты информации от перехвата. На данный момент HTTPS поддерживается всеми популярными веб-браузерами.

**45. Проблемы безопасности в сетях. Методы и средства обеспечения безопасности.**

**Безопасность информации** - 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.

**Криптография** — наука об обеспечении безопасности данных.

Криптография и построенные на ее основе системы призваны решать следующие задачи:

 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.

 Аутентификация. Необходимо однозначно идентифицировать отправителя, при однозначной идентификации отправитель не может отказаться от послания.

 Обеспечивается электронной цифровой подписью и сертификатом.

 Целостность. Информация должна быть защищена от несанкционированного изменения как при хранении, так и при передаче.

 Обеспечивается электронной цифровой подписью.

**Методы обеспечения безопасности в Интернет**

В соответствии с названными задачами основными методами обеспечения безопасности выступают

 шифрование

 цифровая подпись

 сертификаты

**Шифрование**

Любая система шифрования работает по определенной методологии, включая в себя

 один или более алгоритмов шифрования (математических формул),

 ключи, используемые этими алгоритмами,

 также систему управления ключами

**Два основных типа криптографических алгоритмов**

1. Симметричные алгоритмы, основанные на использовании закрытых, секретных ключей, когда и шифрование, и дешифрирование производятся с помощью одного и того же ключа;

2. Асимметричные алгоритмы с открытым ключом, используются один открытый и один закрытый ключ, операции шифрования и дешифрования производятся с помощью разных ключей.

**Симметричные методы шифрования**

 Технология шифрования с секретным ключом требует, чтобы оба участника зашифрованной переписки имели доступ к одному и тому же ключу.

 Алгоритмы симметричного шифрования используют ключи не очень большой длины и могут быстро шифровать большие объемы данных.

**Порядок использования систем с симметричными ключами**

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.

**Асимметричные методы шифрования**

 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.

 Множество классов необратимых функций порождает все разнообразие систем с открытым ключом.

**Известно несколько криптосистем с открытым ключом:**

 RSA - наиболее разработана на сегодня система. Этот алгоритм стал мировым фактически признанным стандартом для открытых систем и рекомендован МККТТ (Международный Консультативный Комитет по телефонии и телеграфии),

 ECC (криптосистема на основе эллиптических кривых),

Алгоритмы систем шифрования с открытым ключом можно использовать в качестве следующих инструментов:

 самостоятельные средства защиты передаваемых и хранимых данных;

 средства для распределения ключей;

 средства аутентификации пользователей (для создания электронной цифровой подписи).

**Цифровая подпись**

Технология электронной подписи использует либо секретный ключ (оба участника сделки применяют один и тот же ключ), либо открытый ключ (требуется пара ключей — открытый и личный).

**Хэш-функции** используются для обнаружения факта модификации сообщения, то есть для электронной подписи.

**Существует несколько защищенных хэш-функций:**

 Message Digest 5 (MD-5),

 Secure Hash Algorithm (SHA) и др.

Хэш-функции гарантируют: что разные документы будут иметь разные электронные подписи, и что даже самые незначительные изменения документа вызовут изменение его дайджеста.

**Протоколы и стандарты безопасности**

Основные причины обеспокоенности безопасности

 Отсутствие гарантии конфиденциальности

 Недостаточный уровень проверки (аутентификации) участников операции

 Нет гарантии целостности данных

Наиболее распространенными механизмами, призванными обеспечить безопасность проведения электронных платежей через Интернет являются:

 Протокол SSL (Secure Socket Layer), обеспечивающий шифрование передаваемых через Интернет данных

 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.

 Является стандартом, основанным на криптографии с открытыми ключами.

 Обеспечивает защиту данных, передаваемых в сетях TCP/IP по протоколам приложений за счет шифрования и аутентификации серверов и клиентов.

 Позволяет решить часть названных проблем безопасности, однако его роль в основном ограничивается обеспечением шифрования передаваемых данных.

**46. Административный уровень информационной безопасности (политика безопасности предприятия/организации).**

К административному уровню информационной безопасности относятся действия общего характера, предпринимаемые руководством организации. Главная цель мер административного уровня – сформировать программу работ в области информационной

безопасности и обеспечить её выполнение, выделяя необходимые ресурсы и контролируя состояние дел. Основой программы является политика безопасности, отражающая подход организации к защите своих информационных активов. Политика безопасности строится на основе анализа рисков, которые признаются реальными для информационной системы

организации. Когда риски проанализированы и стратегия защиты определена, составляется программа обеспечения информационной безопасности. Под эту программу выделяются ресурсы, назначаются ответственные, определяется порядок контроля выполнения программы.

**Политика безопасности** – совокупность документированных решений, принимаемых руководством организации и направленных на защиту информации. С практической точки зрения политика безопасности рассматривается на трёх уровнях. К верхнему уровню

относятся решения, затрагивающие организацию в целом. Они носят общий характер и исходят от руководства организации.

Для политики **верхнего уровня** цели организации в области информационной безопасности формулируются в терминах целостности, доступности и конфиденциальности. Если организация отвечает за поддержание критически важных баз данных, на первом плане может стоять уменьшение числа потерь, повреждений

или искажений данных.

На верхний уровень выносится управление защитными ресурсами и координация использования этих ресурсов, выделение специального персонала для защиты критически важных систем и взаимодействие с другими организациями, обеспечивающими или контролирующими режим безопасности. Политика верхнего уровня должна чётко очерчивать сферу своего влияния.

В политике определяются обязанности должностных лиц по выработке программы безопасности и претворению её в жизнь.

К **среднему уровню** можно отнести вопросы, касающиеся отдельных аспектов информационной безопасности, но важные для различных эксплуатируемых организацией систем. Примеры таких вопросов – отношение к передовым технологиям, доступ в Internet (как совместить свободу доступа к информации с защитой от внешних угроз?), использование домашних компьютеров, применение пользователями неофициального программного обеспечения.

Политика безопасности **нижнего уровня** относится к конкретным информационным сервисам. Она включает в себя два аспекта – цели и правила их достижения, поэтому её порой трудно отделить от вопросов реализации. В отличие от двух верхних уровней рас-

сматриваемая политика должна быть определена более подробно. Есть много вещей, специфичных для отдельных видов услуг, которые нельзя единым образом регламентировать в рамках всей организации. В то же время эти вещи настолько важны для обеспечения режима безопасности, что относящиеся к ним решения должны приниматься на управленческом, а не техническом уровне. Приведем несколько примеров вопросов, на которые следует дать ответ в политике безопасности нижнего уровня:

1 Кто имеет право доступа к объектам, поддерживаемым

сервисом?

2 При каких условиях можно читать и модифицировать дан-

ные?

3 Как организован удалённый доступ к сервису?

**47. Процедурный (организационный) уровень информационной безопасности.**

На процедурном уровне рассматриваются меры безопасности, которые ориентированы на людей, а не на технические средства.

На процедурном уровне можно выделить следующие классы мер:

– управление персоналом;

– физическая защита;

– поддержание работоспособности;

– реагирование на нарушения режима безопасности;

– планирование восстановительных работ.

**Управление персоналом**

Управление персоналом начинается с приёма нового сотрудника на работу и даже раньше – с составления должностной инструкции. Уже на данном этапе желательно подключить к работе специалиста по информационной безопасности для определения компьютерных привилегий, ассоциируемых с должностью. Существует два общих принципа, которые следует иметь в виду:

– разделение обязанностей;

– минимизация привилегий.

**Принцип разделения обязанностей** предписывает так распределять роли и ответственность, чтобы один человек не мог нарушить критически важный для организации процесс. Например, нежелательна ситуация, когда крупные платежи от имени организации выполняет один человек. Надежнее поручить одному сотруднику оформление заявок на подобные платежи, а другому – заверять эти заявки.

**Принцип минимизации привилегий** предписывает выделять пользователям только те права доступа, которые необходимы им для выполнения служебных обязанностей. Назначение этого принципа очевидно – уменьшить ущерб от случайных или умышленных некорректных действий.

Предварительное составление описания должности позволяет оценить её критичность и спланировать процедуру проверки и отбора кандидатов. Чем ответственнее должность, тем тщательнее нужно проверять кандидатов: навести о них справки, быть может, побеседовать с бывшими сослуживцами.

Когда кандидат определён, он, вероятно, должен пройти обучение; по крайней мере его следует подробно ознакомить со служебными обязанностями, а также с нормами и процедурами информационной безопасности.

С момента заведения учётной записи начинается его администрирование, а также протоколирование и анализ действий пользователя. Постепенно изменяется окружение, в котором работает пользователь, его служебные обязанности. Всё это требует соответствующего изменения привилегий.

Ликвидация учётной записи пользователя, особенно в случае конфликта между сотрудником и организацией, должна производиться максимально оперативно. Возможно и физическое ограничение доступа к рабочему месту.

**Физическая защита**

Безопасность информационной системы зависит от окружения, в котором она функционирует. Необходимо принять меры для защиты зданий и прилегающей территории, поддерживающей инфраструктуры, вычислительной техники, носителей данных. Основной принцип физической защиты, соблюдение которого следует постоянно контролировать, формулируется как «непрерывность защиты в пространстве и времени».

Мы кратко рассмотрим следующие направления физической защиты:

– физическое управление доступом;

– противопожарные меры;

– защита поддерживающей инфраструктуры;

– защита от перехвата данных;

– защита мобильных систем.

Меры **физического управления доступом** позволяют контролировать и при необходимости ограничивать вход и выход сотрудников и посетителей. Контролироваться может всё здание организации, а также отдельные помещения. Нужно сделать так, чтобы посетители по возможности не имели непосредственного доступа к компьютерам или в крайнем случае позаботиться о том, чтобы от окон и дверей не просматривались экраны мониторов и принтеры. Необходимо, чтобы посетителей по внешнему виду можно было отличить от сотрудников. Если отличие состоит в том, что посетителям выдаются идентификационные карточки, а сотрудники ходят «без опознавательных знаков», злоумышленнику достаточно снять карточку, чтобы его считали «своим». Очевидно, соответствующие карточки нужно выдавать всем.

Необходимо установка **противопожарной** сигнализации и автоматических средств пожаротушения. Отдельную проблему составляют аварии водопровода. Они происходят нечасто, но могут нанести огромный ущерб. При размещении компьютеров необходимо принять во внимание расположение водопроводных и канализационных труб и постараться держаться от них подальше.

Мобильные и портативные компьютеры – заманчивый объект кражи. Их часто оставляют без присмотра, в автомобиле или на работе, и похитить такой компьютер совсем несложно. Необходимо шифровать данные на жёстких дисках таких компьютеров.

**Поддержание работоспособности**

Существует ряд рутинных мероприятий, направленных на поддержание работоспособности информационных систем. Именно здесь таится наибольшая опасность. Нечаянные ошибки системных администраторов и пользователей грозят повреждением аппаратуры, разрушением программ и данных; в лучшем случае они создают бреши в защите, которые делают возможной реализацию угроз.

Можно выделить следующие направления повседневной деятельности:

– поддержка пользователей;

– поддержка программного обеспечения;

– конфигурационное управление;

– резервное копирование;

– регламентные работы.

**Поддержка пользователей** подразумевает прежде всего консультирование и оказание помощи при решении разного рода проблем. Иногда в организациях создают для этой цели специальный «справочный стол», но чаще от пользователей «отбивается» системный администратор.

**Поддержка программного обеспечения**. Прежде всего необходимо следить за тем, какое программное обеспечение установлено на компьютерах. Если пользователи будут устанавливать программы по своему усмотрению, это может привести к заражению вирусами, а также появлению утилит, действующих в обход защитных средств.

Второй аспект поддержки программного обеспечения – контроль за отсутствием неавторизованного изменения программ и прав доступа к ним. Сюда же можно отнести поддержку эталонных копий программных систем. Обычно контроль достигается комбинированием средств физического и логического управления доступом, а также использованием утилит проверки и обеспечения целостности.

**Резервное копирование** необходимо для восстановления программ и данных после аварий. И здесь целесообразно автоматизировать работу, как минимум сформировав компьютерное расписание создания полных и инкрементальных копий, а как максимум – воспользовавшись соответствующими программными продуктами. Нужно также наладить размещение копий в безопасном месте, защищенном от несанкционированного доступа, пожаров, протечек, то есть от всего, что может привести к краже или повреждению носителей. Целесообразно иметь несколько экземпляров резервных копий.

**Реагирование на нарушения режима безопасности**

Программа безопасности, принятая организацией, должна предусматривать набор оперативных мероприятий, направленных на обнаружение и нейтрализацию нарушений режима информационной безопасности. Важно, чтобы в подобных случаях последовательность действий была спланирована заранее, поскольку меры нужно принимать срочные и скоординированные.

В организации должен быть человек, доступный 24 часа в сутки, который отвечает за реакцию на нарушения. Все должны знать координаты этого человека и обращаться к нему при первых признаках опасности. В общем, как при пожаре, нужно знать, куда звонить и что делать до приезда пожарной команды.

**Планирование восстановительных работ**

Ни одна организация не застрахована от серьёзных аварий, вызванных естественными причинами, действиями злоумышленника, халатностью или некомпетентностью. Планирование восстановительных работ позволяет подготовиться к авариям, уменьшить ущерб от них и сохранить способность к функционированию хотя бы в минимальном объёме.

**48. Основные меры программно-технического уровня информационной безопасности.**

Программно-технические меры, то есть меры, направленные на контроль оборудования, программ и данных, образуют последний и один из самых важных рубежей информационной безопасности. Как показывает статистика, ущерб наносят в основном действия легальных пользователей, по отношению к которым процедурные регуляторы малоэффективны. Главные «враги» – некомпетентность и неаккуратность при выполнении служебных обязанностей, и только программно-технические меры способны им противостоять.

Для программно-технического уровня основным является понятие

сервиса безопасности, в который входят:

– идентификация и аутентификация;

– управление доступом;

– протоколирование и аудит;

– шифрование;

– контроль целостности;

– экранирование;

– анализ защищенности;

– обеспечение отказоустойчивости;

– обеспечение безопасного восстановления;

– туннелирование;

– управление.

**Особенности современных информационных систем, существенные с точки зрения безопасности**

Информационная система современной организации является весьма сложным образованием, которое пользуется многочисленными внешними сервисами и в свою очередь предоставляет собственные сервисы вовне.

Следует учитывать ещё по крайней мере два момента. Во-первых, для каждого сервиса основные грани ИБ (доступность, целостность, конфиденциальность) трактуются по-своему. Целостность с точки зрения системы управления базами данных и с точки зрения почтового сервера – вещи принципиально разные. Бессмысленно говорить о безопасности локальной или иной сети вообще, если сеть включает в себя разнородные компоненты. Следует анализировать защищенность сервисов, функционирующих в сети. Для разных сервисов и защиту строят по-разному. Во-вторых, основная угроза информационной безопасности организаций по-прежнему исходит не от внешних злоумышленников,

а от собственных сотрудников.

**Архитектурная безопасность**

Сервисы безопасности, какими бы мощными они ни были, сами по себе не могут гарантировать надежность программно-технического уровня защиты. Только проверенная архитектура способна сделать эффективным объединение сервисов, обеспечить управляемость информационной системы, ее способность развиваться и противостоять новым угрозам при сохранении таких свойств, как высокая производительность, простота и удобство использования.

Если непрерывность нарушена, тогда у злоумышленника или недовольного пользователя появится возможность миновать защитные средства, он, разумеется, так и сделает.

Следование признанным стандартам и использование апробированных решений повышает надежность ИС и уменьшает вероятность попадания в тупиковую ситуацию, когда обеспечение безопасности потребует непомерно больших затрат и принципиальных модификаций.

Иерархическая организация ИС с небольшим числом сущностей на каждом уровне необходима по технологическим соображениям. При нарушении данного принципа система станет неуправляемой и, следовательно, обеспечить ее безопасность будет невозможно. Применительно к программно-техническому уровню принцип минимизации привилегий предписывает выделять пользователям и администраторам только те права доступа, которые необходимы им для выполнения служебных обязанностей. Этот принцип позволяет уменьшить ущерб от случайных или умышленных некорректных действий пользователей и администраторов.

Принцип разделения обязанностей предполагает такое распределение ролей и ответственности, чтобы один человек не мог нарушить критически важный для организации процесс или создать брешь в защите по заказу злоумышленников. В частности, соблюдение данного принципа особенно важно, чтобы предотвратить злонамеренные или неквалифицированные действия системного администратора.

**49. Основные пути решения задач системы защиты информации.**

Технологические, производственные и коммерческие данные, которые используют предприятия, обладают высокой стоимостью, а их утрата или утечка может привести к серьезным финансовым потерям. Поэтому одной из целей для предприятий отрасли является создание надежной системы защиты информации (СЗИ).

Система защиты информации — это комплекс организационных и технических мер, направленных на обеспечение информационной безопасности предприятия. Главным объектом защиты являются данные, которые обрабатываются в автоматизированной системе управления (АСУ) и задействованы при выполнении бизнес-процессов.

Основные угрозы для информационной безопасности любой компании связаны с кражей данных (например, промышленный шпионаж), использованием непроверенного программного обеспечения (например, содержащего вирусы), хакерскими атаками, получением спама (также может содержать вирусы), халатностью сотрудников. В результате компании несут значительные потери.

Процесс создания системы защиты информации можно разделить на три этапа:

формирование политики предприятия в области информационной безопасности;

выбор и внедрение технических и программных средств защиты;

разработка и проведение ряда организационных мероприятий.

Фундаментом для создания системы защиты информации является документ, в котором формулируются принципы и основные положения политики предприятия в области информационной безопасности. Какие вопросы он охватывает?

1. Разработка правового обеспечения защиты информации. Фактически это система нормативно-правовых документов, актуальных для деятельности предприятия.

2. Определение потенциальных угроз безопасности информации. Их можно разделить на три группы — это угрозы, возникающие:

вследствие действий

вследствие некорректной работы или отказа технических или программных средств

из-за стихийных бедствий, природных катаклизмов, форс-мажорных обстоятельств

3. Составление перечня данных, подлежащих защите. Информация, которая используется на предприятии, может быть открытой (доступна для всех) или закрытой (доступна для ограниченного круга лиц). К первому типу относятся сведения, которые не составляют государственной или коммерческой тайны, не относятся к категории конфиденциальной информации (согласно законодательству или внутренним документам предприятия).

Ко второму типу относятся:

данные, являющиеся государственной тайной — их перечень определяется законодательством;

коммерческие или служебные сведения — любая информация, связанная с производством, финансами, использующимися технологиями, утечка или утрата которой может нанести ущерб интересам предприятия;

персональные данные сотрудников.

4. Создание подразделения, ответственного за вопросы защиты информации.

5. Определение основных направлений обеспечения информационной безопасности. В рамках решения этой задачи, в частности, обозначаются компоненты, которые нуждаются в защите, определяются необходимые программные и технические средства, формулируются организационные меры, направленные на защиту информации.

**50. Основные аспекты информационной безопасности (доступность, целостность, конфиденциальность).**

**Информационная безопасность** – многогранная, можно даже сказать, многомерная область деятельности, в которой успех может принести только системный, комплексный подход.

Спектр интересов субъектов, связанных с использованием информационных систем, можно разделить на следующие категории: обеспечение **доступности**, **целостности** и **конфиденциальности** информационных ресурсов и поддерживающей инфраструктуры.

**Доступность** – это возможность за приемлемое время получить требуемую информационную услугу. Под **целостностью** подразумевается актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения. Наконец, **конфиденциальность** – это защита от несанкционированного доступа к информации.

Информационные системы создаются (приобретаются) для получения определенных информационных услуг. Если по тем или иным причинам предоставить эти услуги пользователям становится невозможно, это, очевидно, наносит ущерб всем субъектам информационных отношений. Поэтому, не противопоставляя доступность остальным аспектам, мы выделяем ее как важнейший элемент информационной безопасности.

Особенно ярко ведущая роль доступности проявляется в разного рода системах управления – производством, транспортом и т.п. Внешне менее драматичные, но также весьма неприятные последствия – и материальные, и моральные – может иметь длительная недоступность информационных услуг, которыми пользуется большое количество людей (продажа железнодорожных и авиабилетов, банковские услуги и т.п.).

**Целостность** можно подразделить на статическую (понимаемую как неизменность информационных объектов) и динамическую (относящуюся к корректному выполнению сложных действий (транзакций)). Средства контроля динамической целостности применяются, в частности, при анализе потока финансовых сообщений с целью выявления кражи, переупорядочения или дублирования отдельных сообщений.

Целостность оказывается важнейшим аспектом ИБ в тех случаях, когда информация служит "руководством к действию". Рецептура лекарств, предписанные медицинские процедуры, набор и характеристики комплектующих изделий, ход технологического процесса – все это примеры информации, нарушение целостности которой может оказаться в буквальном смысле смертельным. Неприятно и искажение официальной информации, будь то текст закона или страница Web-сервера какой-либо правительственной организации.

**51. Злоумышленники и информационная безопасность.**

Задача защиты информации, хранимой в компьютерных системах, от несанкционированного доступа (НСД), является весьма актуальной. Для решения этой задачи используется целый комплекс средств, включающий в себя технические, программно-аппаратные средства и административные меры защиты информации.  
  
Построение надежной защиты компьютерной системы невозможно без предварительного анализа возможных угроз безопасности системы. Этот анализ должен включать в себя:  
  
• оценку ценности информации, хранящейся в системе;  
  
• оценку затрат времени и средств на вскрытие системы, допустимых для злоумышленников;  
  
• оценку характера хранящейся в системе информации, выделение наиболее опасных угроз (несанкционированное чтение, несанкционированное изменение и т.д.);  
  
• построение модели злоумышленника - другими словами, оценка того, от кого нужно защищаться - от постороннего лица, пользователя системы, администратора и т.д.;  
  
• оценку допустимых затрат времени, средств и ресурсов системы на организацию ее защиты.  
  
При проведении такого анализа защищенной системы эксперт фактически ставит себя на место хакера, пытающегося преодолеть ее защиту

Хотя слово "хакер" в последнее время встречается в литературе очень часто, у употребляющих его авторов до сих пор не сложилось единое понимание того, кто же такие хакеры.   
В отношении атакуемой системы хакер может выступать в одной из следующих ролей:  
  
• постороннее лицо, не имеющее легального доступа к системе. Хакер может атаковать систему только с использованием общедоступных глобальных сетей;  
• сотрудник организации, не имеющий легального доступа к атакуемой системе.   
• пользователь системы, обладающий минимальными полномочиями. Хакер может атаковать систему, используя ошибки в программном обеспечении и в администрировании системы;  
•- администратор системы. Хакер имеет легально полученные полномочия, достаточные для того, чтобы успешно атаковать систему. Для нейтрализации этой угрозы в системе должны быть предусмотрены средства противодействия несанкционированным действиям администраторов;  
• разработчик системы. Хакер может встраивать в код системы "люки" (недокументированные возможности), которые в дальнейшем позволят ему осуществлять несанкционированный доступ (НСД) к ресурсам системы.  
  
Тем не менее, даже в самом общем случае существуют определенные правила, которых целесообразно придерживаться при организации защиты. Эти правила сформулированы ниже в виде практических советов.  
  
1. Всегда будьте в курсе последних новинок науки и техники в области компьютерной безопасности. Регулярно просматривайте материалы хакерских серверов Internet, подпишитесь на хакерские телеконференции (newsgroups). Подпишитесь на несколько журналов по компьютерной безопасности. Будьте образованнее вашего противника.  
  
3. Храните в тайне информацию о реализации защиты. Чем меньше хакер знает об атакуемой системе, тем труднее ему осуществить атаку.  
  
5. Постарайтесь максимально ограничить объем защищаемой сети. Не подключайте защищаемую сеть к Internet без крайней необходимости.  
  
6. Перед тем, как приобрести новое программное обеспечение, почитайте о нем на хакерских серверах Internet.  
  
7. Размещайте серверы БД в охраняемом помещении. Никто не должен иметь доступ к базам данных иначе чем через сеть.  
  
10. Все без исключения пакеты, передаваемые по открытым каналам связи, должны шифроваться.  
  
11. Все без исключения пакеты должны подписываться цифровой подписью независимо от того, по какому каналу связи передается пакет.  
  
16. Регулярно проверяйте целостность программного обеспечения.  
  
18. Создайте в системе несколько ловушек для хакеров. В роли ловушки может выступать, например, документ с заманчивым именем, прочитать который невозможно без использования программной закладки. Если система аудита зафиксировала успешное обращение к этому документу, значит, хакер сумел внедрить закладку в защищаемую систему.  
  
**52. Понятие угрозы безопасности в компьютерных сетях.**

Компьютерные вирусы, сетевые черви, троянские программы и хакерские атаки давно перестали ассоциироваться с фантастическими боевиками голливудского производства. Компьютерная "фауна", хулиганство и преступления — сейчас это обыденные явления, с которыми регулярно сталкиваются пользователи современных компьютерных систем.

Фактически во многих случаях мы оказываемся "заложниками" современных технологий, зачастую не вполне понимая того, что эти технологии не являются в достаточной мере защищенными от вирусных и/или хакерских атак. Что работоспособность этих систем напрямую зависит от "эпидемиологической ситуации в компьютерном мире" в данный текущий момент времени, которая к тому же может измениться в любую минуту, как это уже происходило неоднократно.

**основные классы угроз**  
- вредное программное обеспечение;  
- спам;  
- глобальные сетевые атаки.  
К первой категории относятся вирусные и троянские программы (включая сетевых червей), а также сетевые пакеты, которые используются в хакерских атаках. Ущерб, который наносится вредным программным обеспечением, можно классифицировать следующим образом:  
**Неавторизованный доступ** к персональной/корпоративной/государственной информации, т.е. ее уничтожение, изменение (включая намеренное искажение информации в злоумышленных целях), передача (отсылка, т.е. организация утечки информации, включая конфиденциальной/секретной).  
**Нештатное поведение программного обеспечения и железа,** т.е. сбои и/или замедление работы компьютерных систем, зависания элементов систем, и т.п.  
**Использование вычислительных, дисковых и прочих ресурсов** систем в чужих интересах и/или в ущерб интересам владельца ресурсов.  
Вторая категория угроз (спам) появилась значительно позднее, чем первая (вредоносные программы), однако по своей важности уверенно приближается к ней. Назойливость спама является далеко не единственной причиной, по которой к нему следует относиться как к отдельной категории современных угроз.

Спам — это:  
**Увеличение нагрузки на почтовые серверы,** что влечет за собой неоправданное увеличение средств, вкладываемых в организацию инфраструктуры сетей, а также увеличение численности персонала, эти сети обслуживающего.  
**Риск потери важной информации** по причине того, что она просто затерялась среди спама (если, например, 99 из 100 писем — это спам, то "заодно" уничтожится и нужное письмо). Возможно, также, что письмо потеряется провайдером по причине того, что почтовый размер ящика переполнен спамом.  
**Пустая трата времени:** для фильтрации спама сотрудники компаний (и домашние пользователи) тратят свое рабочее (и личное) время, доля которого возрастает с возрастанием доли спама в корреспонденции.   
**Риск стать жертвой мошенников**(привлечение к финансовым аферам или пирамидам), появление в почте нежелательной информации (если, например, домашний компьютер и почта используется также и детьми) и прочие подобные крупные, средние и мелкие неприятности.

Последняя (третья) категория угроз — это глобальные сетевые атаки, возникающие в результате запланированных действий хакера или группы хакеров, или как результат неконтролируемого распространения сетевых вирусов-червей.

Сетевые атаки приводят к следующим **последствиям:**  
**Отказ клиентских и серверных компонентов сетей** по причине перегрузки чрезмерным количеством запросов на обслуживание.  
**Заметное замедление работы** локальных и глобальных сетей по причине перегрузки каналов, работа которых блокируется чрезмерным количеством передаваемых данных, т.е. "флудинг" (от flood) каналов передачи информации.

**причины возникновения угроз**

Поскольку вирусы не возникают сами по себе в результате электромагнитных коллизий, а создаются людьми, то для ответа на этот вопрос следует разобраться в психологии тех индивидуумов, которые создают "вредное" программное обеспечение, в обиходе именуемое "вирусами". Наиболее вероятными причинами, толкающими вирусо-писателей на создание и распространение вредоносного программного обеспечения являются следующие:  
Мошенничество с целью присвоения ресурсов жертвы: незаметное управление пораженным компьютером, воровство паролей доступа в Интернет, средств с "кошельков" WebMoney и даже кодов доступа к персональным банковским счетам (в том случае, если жертва использует данный сервис).

Спаммеры занимаются массовой рассылкой электронной рекламы с единственной целью — получение прибыли, поскольку. Подобный метод маркетинга товаров и услуг, как оказалось, достаточно хорошо работает — очень небольшой (но все же не нулевой) процент получателей спама воспринимает его. В результате заказчик спама получает клиента, а спаммер получает дополнительные заказы на повторную рассылку спама — и круг "заказчик-спаммер-клиент" замыкается, что доказывает непобедимость спама в современных сетях при существующих правилах работы в сети.

**53. Атаки и информационная безопасность.**

Для организации коммуникаций в неоднородной сетевой среде применяются набор протоколов TCP/IP, обеспечивая совместимость между компьютерами разных типов. Данный набор протоколов завоевал популярность благодаря совместимости и предоставлению доступа к ресурсам глобальной сети Интернет и стал стандартом для межсетевого взаимодействия.  
  
Однако повсеместное распространение стека протоколов TCP/IP обнажило и его слабые стороны. В особенности из-за этого удалённым атакам подвержены распределённые системы, поскольку их компоненты обычно используют открытые каналы передачи данных, и нарушитель может не только проводить пассивное прослушивание передаваемой информации, но и модифицировать передаваемый трафик.  
  
Трудность выявления проведения удалённой атаки и относительная простота проведения (из-за избыточной функциональности современных систем) выводит этот вид неправомерных действий на первое место по степени опасности и препятствует своевременному реагированию на осуществлённую угрозу, в результате чего у нарушителя увеличиваются шансы успешной реализации атаки.  
  
Существует множество специальных технологий, с помощью которых злоумышленники могут атаковать удаленные компьютеры. Они подразделяются на несколько категорий:

1. **DoS-атаки** (атаки типа отказ в обслуживании) представляют собой попытку сделать компьютер или сеть недоступными тем пользователями, для которых они предназначены. Обмен данными между пользователями пораженного компьютера затруднен или невозможен в приемлемом режиме.  
     
   Компьютеры, подвергшиеся действию [DoS-атаки](http://www.securrity.ru/terms/51-dos-ataka.html), обычно должны быть перезагружены для восстановления нормальной работы. В большинстве случаев объектами этой атаки становятся веб-серверы, а целью является вывод их из строя и, как следствие, их недоступность на некоторое время.
2. **Атака путем подделки записей кэша DNS** (сервер доменных имен) позволяет злоумышленникам убедить DNS-сервер любого компьютера в том, что предоставляемые подложные данные являются истинными. Ложная информация кэшируется на определенное время, давая злоумышленникам возможность перезаписать ответы DNS-сервера с IP-адресами. В результате при попытке посещения веб-сайтов пользователь загружает компьютерные вирусы и черви вместо исходного содержимого.
3. **Атаки червей**  
   Компьютерные черви — это содержащие злонамеренный код программы, которые атакуют главные компьютеры и распространяются через сеть. [Сетевые черви](http://www.securrity.ru/terms/63-setevoj-cherv.html) используют сетевые уязвимости различных приложений.   
   Многих из атак червей (Sasser, SqlSlammer) можно избежать, используя настройки персонального файервола по умолчанию или с помощью блокировки незащищенных и неиспользуемых портов. Очень важно регулярно устанавливать новейшие пакеты обновления операционной системы.
4. **Сканирование портов** используется, чтобы определить, какие порты компьютера открыты на узле сети. Сканер портов представляет собой программное обеспечение, которое предназначено для поиска таких портов.  
     
   Компьютерный порт является виртуальной точкой, которая управляет сетевым трафиком в обоих направлениях. Это является критичным с точки зрения сетевой безопасности. В больших сетях данные, которые собираются с помощью сканера портов, могут помочь выявить потенциальные уязвимости компьютерных систем. Такое использование является допустимым.  
     
   Однако сканеры часто используются злоумышленниками для взлома систем безопасности. Первым шагом отправляется серия пакетов на каждый из портов. В зависимости от полученных ответов определяется, какой из портов можно использовать. Сканирование не причиняет вреда само по себе, но следует иметь в виду, что такая активность зачастую является признаком попытки выявления уязвимости и последующей атаки злоумышленников на систему. Сетевые администраторы обычно советуют блокировать все неиспользуемые порты и защищать используемые от неавторизованного доступа.
5. **Атаки по протоколу ICMP**  
   Протокол ICMP является популярным и широко используемым протоколом Интернета. Применяется он преимущественно подключенными к сети компьютерами для отправки сообщений об ошибках.  
     
   Удаленные злоумышленники пытаются использовать уязвимости протокола ICMP. Протокол ICMP предназначен для передачи данных в одном направлении без аутентификации. Это позволяет злоумышленникам организовывать DoS-атаки (отказ в обслуживании) или атаки, предоставляющие не имеющим на это права лицам доступ ко входящим и исходящим пакетам.  
     
   Типичными примерами атак по протоколу ICMP являются [ping-флуд](http://www.securrity.ru/terms/55-ping-flud.html), флуд эхо-запросов по протоколу ICMP и smurf-атаки. Компьютеры, подвергающиеся атаке по протоколу ICMP, значительно замедляют свою работу (это касается всех приложений, использующих Интернет), и у них возникают проблемы при подключении к Интернету.

**54. Поддержка взаимодействия с вычислительной сетью в операционных системах.**  Сетевая операционная система составляет основу ПО любой вычислительной сети.

В узком смысле сетевая операционная система — это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети. Есть и более развернутое определение сетевой операционной системы: сетевая операционная система — это система программных средств, управляющих процессами в сети и объединенных общей архитектурой, определенными коммуникационными протоколами и механизмами взаимодействия вычислительных процессов. Она обеспечивает пользователям стандартный и удобный доступ к разнообразным сетевым ресурсам и обладает высоким уровнем прозрачности, т.е. изолирует от пользователя все различия, особенности и физические параметры привязки процессов к обрабатываемым ресурсам.

В сетевой операционной системе отдельной машины можно выделить несколько частей:

Средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных операционных систем.

Средства предоставления собственных ресурсов и услуг в общее пользование — серверная часть операционной системы (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.

Средства запроса доступа к удаленным ресурсам и услугам и их использования — клиентская часть операционной системы (редиректор). Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в другой форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.

Коммуникационные средства операционной системы, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., то есть является средством транспортировки сообщений.

В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.

Если выполнение каких-либо серверных функций является основным назначением компьютера (например, предоставление файлов в общее пользование всем остальным пользователям сети или организация совместного использования факса), то такой компьютер называется выделенным сервером. Выделенный сервер не принято использовать в качестве компьютера для выполнения текущих задач, не связанных с его основным назначением, так как это может уменьшить производительность его работы как сервера.

Также может возникнуть и другая ситуация, когда локальный пользователь не возражает против использования его ресурсов и сам не исключает возможности обращения к другим компьютерам. ОС, устанавливаемая на его компьютере, должна включать и серверную, и клиентскую части.

Сетевые операционные системы имеют разные свойства в зависимости от того, предназначены они для сетей масштаба рабочей группы (отдела), для сетей масштаба офиса или для сетей масштаба предприятия.