**Лекция №1 03.09.2016**

**Применение и классификация информационных систем**

Информационная сеть – набор связанных между собой компьютеров из других устройств; два компьютера называются связанными, если они способны обмениваться данными.

Применение информационных сетей

* Корпоративная сеть (на сегодняшний день компьютеры объединены, как правило, в единую цель, независимо от размеров предприятия; первой целью такого объединения является предоставление различного доступа предприятиям; например, сетевой принтер, доступ к которому возможен с любого компьютера в корпоративной сети; кроме совместного использования различных устройств, возможно совместное использование данных, то есть пользователи удаленные от этих данных могут получать к ним доступ с компьютеров корпоративной сети; также создание коммуникационной среды, так как большинство людей на работе для общения предпочитает использовать электронную почту, помимо электронной почты в корпоративных сетях используется технология – видеоконференция; предоставление возможности совместной работы с документами (отдаленные друг от друга сотрудники имеют возможность составлять совместные документы, например, гугл добс);
* домашняя сеть (ПК, КПК, ноутбуки, периферийные устройств совместного доступа (МФУ, принтеры), приборы для развлечения (телевизоры, проигрыватели, различные видеокамеры, аудиосистемы, игровые консоли), телекоммуникационные устройства (стационарные и мобильные телефоны, факсы), бытовые приборы (микроволновая печь, холодильники, часы, система отопления и кондиционированная, системы освещения), измерительные приборы (счетчики, системы пожарной сигнализации, термостаты);
* всемирная паутина – это распределенная система, предоставляющая доступ к различным ресурсам, расположенных на компьютерах и подключенных к сети интернет;
* общение (огромное количество людей общаются с друг другом, используя программы мгновенного обмена сообщениями, но в отличии от электронной почты данный вид программы позволяют отправлять мгновенные сообщения, звук и видео сообщения);
* интерактивные развлечения (применяется сетей в области интерактивных развлечений приобретает всю большую популярность и способствует развитию данной области, это происходит потому, что современные информационные сети позволяют играть вместе пользователям со всего мира).

Классификация информационных сетей

В основном, информационные сети классифицируются по признакам структурной и функциональной организации сети.

1. По размерам сети:

* персональная сеть (предназначена для одного человека; например, сеть, соединяющая персональный компьютер и различные устройства связи (мобильный телефон));
* локальная сеть (объединяет компьютеры, как правило, в одном офисе, здании или на территории какого-либо предприятия);
* муниципальная сеть (объединяет компьютеры в пределах города; например, система предоставления высокоскоростного доступа к сети Интернет);
* глобальная сеть (охватывает значительную географическую территорию и включает в себя большое число компьютеров; например, крупнейшая глобальная сеть – Интернет).

1. По типу топологии сети (под топологией сети понимают способ описания конфигурации сети в виде графов, где вершинам узлам графа соответствуют различные сетевые устройства, например, компьютеры или коммуникационное оборудование, а ребрам графа соответствуют линии связи между сетевыми устройствами; линии связи могут быть физические, логические (обмениваются информацией) и информационные (показывают каким образом информационные потоки передаются в сети)):

* полносвязная (от выбора топологии характерно зависит характеристики сети, например, наличие между узлами резервных линий связи значительно повышает надежность сети; полносвязная топология соответствует сети, в которой каждый узел непосредственно связан с остальными, несмотря на логическую простоту этот вариант оказывается громоздким и не эффективным, дело в том, что в данной топологии для каждой пары узлов должна быть выделена независимая линия связи и каждый узел должен иметь достаточное количество сетевых интерфейсов для связи со всеми узлами сети; главным недостатком является сложное расширение сети, так как при добавлении нового узла его необходимо соединить с со всеми остальными узлами сети, поэтому данная топология применяется крайне редко и, в основном, в небольших сетях);
* ячеистая (получается из полносвязной удалением некоторых линий связи, эта топология, как правило, характерна для крупных сетей; в сетях с данной топологии отказ какой-либо линии связи, как правило, не приводит к потере связи между двумя узлами, так как всегда имеется одна или несколько резервных линий связи);
* общая шина (линия связи, которая соединяет все компьютеры в единую сеть; при таком соединении узлы могут передавать данные только по очереди, потому что линия связи единственная, в противном случае передаваемая информация будет искажаться; в сетях с данной топологией не страшны отказы отдельных узлов, так как оставшиеся узлы могут нормально обмениваться данными и без них, однако, если происходит отказ (обрыв или неспособность передавать данные) линии связи, то ни один из подключенных к ней узлов не сможет осуществлять передачу данных; данный факт делает эту топологию не очень надежной);
* кольцо (каждый узер соединен только с двумя соседними узлами, от одного он только принимает данные, от другому только передает; в сетях с такой топологией каждый узел получает данные от предыдущего узла и передает их далее, если они адресованы не ему; основным недостатком сетей с кольцевой топологией является то, что повреждении линии связи в одном месте или отказ какого-либо узла приводит к неработоспособности всей сети);
* звезда (все узлы подключаются непосредственно к общему центральному узлу; в функции центрального узла входят перенаправления данных одному или нескольким подключенным к нему узлам; в качестве центрального узла может выступать компьютер, тогда топология называется активная звезда или концентратор, или коммутатор – пассивная звезда; возможность наращивания количества узлов сети ограничивается количеством сетевых интерфейсов центрального узла; в сетях с данной топологии выход из строя одного узла не отображается на работе всей сети в целом, однако отказ центрального узла обернется неработоспособностью всей сети);
* дерево (иногда имеет смысл проектировать сеть с использование нескольких центральных узлов иерархически соединенных между собой связями типа звезда, в результате получается топология дерева, в настоящий момент топология дерева является наиболее распространенной и часто применяемой топологией);
* смешанная (в то время, как в небольших сетях используется типовая топология, то для крупных сетей характерно наличие произвольных линий связи между узлами, в таких сетях можно выделить отдельные подсети, имеющие типовую типологию, поэтому их называют сетями со смешанной топологией).

1. по типу функционального взаимодействия (сетевое устройство, которое предоставляет доступ к своим ресурсам, является серверов; устройство, которое обращается к сетевым ресурсам, является клиентом; в зависимости от того, как распределены функции между устройствами сети, информационные сети делятся на два класса):

* одноранговые сети (все узлы равны в правах доступа к ресурсам друг друга; в такой сети, как правило, отсутствуют устройства, которые выполняют только серверные функции, а каждый узел является как клиентом, так и сервером, одна из областей применения одноранговые сетей – это обмен файлами);
* сети с выделенным сервером (в сетях данного класса имеются устройства, которые предназначены только для выполнения серверных функции, исключая функции клиента; очевидно, что на серверах желательно устанавливать серверные ОС, специально оптимизированные для выполнения тех или иных серверных функции; серверы, как правило, являются специализированными: 1) файл сервер (это сервер, предназначенный для хранения различных файлов; как правило, файл-сервер оптимизирован для выполнения файловых операции и обладает большим объемом дискового пространства); 2) принт-сервер (сервер, представляющий собой сетевое устройство, позволяющее подключить обычный не сетевой принтер к сети для его совместного использования); 3) факс-сервер (сервер, предназначенный для централизованного управления факсимильными устройствами и предоставлению пользователям возможности отправки и получения факсов); 4) почтовый сервер (сервер, позволяющий организовать электронную почту, создает возможность завести личные почтовые ящики всем пользователям сети; почтовый сервер может контролировать переписки, то есть проверять все входящие и исходящие сообщения); 5) сервер баз данных (сервер, предназначенный для развертывания и обслуживания СУБД); 6) сервер приложений (сервер, предоставляющий интегрированную среду для развертывания и выполнения серверных приложений)).

1. по типу технологии передачи:

* широковещательные сети (обладают единым каналом связи, совместно используемым всеми узлами сети, поэтому данные, которые предназначены одному узлу, получают все узлы сети, при получении данных каждый узел выполняет проверку и, если эти данные были адресованы ему, то он их принимает и обрабатывает, в противном случае – данные игнорируются; широковещательные сети позволяют адресовать данные одновременно всем узлам сети, такая операция называется – широковещательная передача, если же данные адресуются под множество узлов, то такая операция называется – групповая передача);
* сети с передачей от узла к узлу (сети с передачей от узла к узлу состоят из большого числа соединенных между собой узлов, поэтому данным, чтобы быть доставленным до пункта назначения необходимо пройти через ряд промежуточных узлов; в сетях данного класса можно адресовать данные только одному узлу сети, такая операция называется однонаправленная передача).

1. по типу среды передачи (информационные сети могут отличаться физической среды, которую они используют для передачи данных, физическая среда может представлять собой кабельные линии связи или радиоканалы наземной, или спутниковой связи):

* проводная сеть;
* беспроводной сети.

1. по скорости передачи:

* низкоскоростная сеть (до 10Мбит/с);
* среднескоростная сеть (до 100Мбит/с);
* высокоскоростная сеть (до 1Гбит/с);
* сверхскоростная сеть (до 10Гбит/с).

Отметим, что единицы измерения, используемые для обозначения скорости передачи отличаются от принятых объемов памяти.

**Лекция №2 10.09.2016**

**Эталонные модели сети**

Эталонная модель сети отражает декомпозицию общей задачи, взаимодействие компьютеров и сетевых устройств на отдельные подзадачи.

Отдельные подзадачи организуют наборы уровней, образующих иерархию. Количество уровней, их название и назначение в разных моделях отличаются, однако во всех моделях целью каждого уровня является предоставление неких примитивных функции на вышестоящие уровни, набор таких операции и функции называется, межуровневый интерфейс.

**Протокол и стек протоколов**

Уровень N одного узла поддерживает только с уровнем N другого узла. Набор правил и соглашений, используемых в такой связи называется, протоколом.

Список используемых протоколов по одному на каждый уровень, называется стеком протоколов.

В действительности данные не пересылаются между соответствующими уровнями узлов сети, вместо этого каждый уровень передает данные уровню, расположенному ниже. Ниже первого уровня расположена линия связи, по которой и выполняется обмен.

**Эталонная модель OSI**

OSI – взаимодействие открытых систем

ISO – международная организация по стандарту

Эталонная модель OSI имеет 7 уровней:

1. Физический уровень (занимается, непосредственно, передачей данных по линии связи; на физическом уровне передаваемые данные представляю последовательность электрических или электромагнитных сигналов, передаваемых в физической среде; функции физического уровня реализуются на всех устройствах, подключенных к сети);
2. Канальный уровень (протокол канального уровня) (канальный уровень осуществляет передачу данных по сети, для этого передаваемые данные разбиваются на отдельные кадры, которые имеют заголовок и поле, содержащее часть передаваемых данных; заголовок кадры заполняется соответствующей служебной информацией, на основании которой различные сетевые устройства будут передавать кадр; например, адрес назначения; канальный уровень разделяют на два подуровня, которые также часто называют уровнями:

* уровень управления логическим каналом (осуществляет управление передачей данных);
* уровень управления доступом к среде (обеспечивает адресацию и механизмы управления доступом к линиям связи);

1. Сетевой уровень (предназначен для определения маршрута передачи данных; данные на сетевом уровне вместе с заголовками сетевого уровня образуют пакеты, если в сети одновременно присутствует слишком большое количество пакетов, то они могут образовывать «заторы» (неспособность справиться с потоком данных); недопущение подобной ситуации также является задачей сетевого уровня, другими словами сетевой уровень занимается синхронизацией процессов передачи пакетов в сети);
2. Транспортный уровень (транспортный протокол) (предназначен для обеспечения контроля передачи данных; в эталонной модели OSI определено множество классов транспортных протоколов, которые предоставляют различные функции для вышестоящих уровней, например, может осуществляться гарантированная доставка данных или обычная передача данных; на транспортном уровне передаваемые данные вместе с заголовками транспортного уровня образуют сегменты; начиная с транспортного уровня, все протоколы реализуются программными средствами, обычно включаемые в состав ОС);
3. Сеансовый уровень (обеспечивает управление диалогом для того, чтобы отслеживать очередность передачи данных, а также представляет средства синхронизации; последние позволяют устанавливать контрольные точки внутри длинных сообщений, чтобы в случае ошибки можно было продолжить передачу с того же места, в котором она оборвалась);
4. Уровень представления передаваемых данных (обеспечивает предоставление передаваемых данных, не меняя при этом их содержание, на этом уровне может осуществляться сжатия, распаковка, шифрование и дешифрование данных, кроме того, при необходимости, на этом уровне может выполнятся переводы от одного формата данных к другому);
5. Прикладной уровень (в действительности представляет собой набор разнообразных протоколов для работы в сетевых приложениях, как правило, прикладные протоколы используются для передачи файлов, сообщений электронной почты и тому подобное).

**Эталонная модель TCP/IP**

Была разработана министерством обороны США. Имеет 4 уровня:

1. Уровень сетевого доступа (обеспечивающих доставку данных к компьютерам и сетевым устройствам);
2. Межсетевой уровень (обеспечивающий маршрутизация пакетов данных в сети; функционально данный уровень близок к сетевому уровню эталонной модели OSI);
3. Транспортный уровень (обеспечивающий передачу данных с определенной степенью надежности, аналогично транспортному уровня модели OSI);
4. Прикладной уровень (состоящий из протоколов, осуществляющий обмен данными между сетевыми приложениями).

**Гибридная эталонная модель**

Несмотря на то, что эталонная модель OSI пользуется большой популярностью в сетях, протоколы OSI не имеют никакого распространения. Для TCP/IP верно обратное утверждение: модель практически не существует в то время, как протоколы TCP/IP чрезвычайно популярны.

В виду этого, на практике используются гибридную эталонную модель, состоящую из пяти уровней:

1. Физический уровень;
2. Канальный уровень;
3. Сетевой уровень;
4. Транспортный уровень;
5. Прикладной уровень.

Первые четыре уровня выполняют те же функции, что и одноименные функции модели OSI. Последний пятый прикладной уровень объединяет в себя три верхних уровня модели OSI (сеансовый, представления, прикладной). В качестве протоколов для взаимодействия между уровнями используется стеки протоколов TCP/IP.

**Сетевые устройства**

Устройства, подключенные к сети, называют, сетевыми устройствами. Эти устройства позволяют осуществлять связь с компьютерами и другими сетевыми устройствами. Выделяют активные и пассивные сетевые устройства.

**Сетевые адаптеры**

Для работы в сети каждый компьютер оснащен платой сетевого интерфейса (NIC), также называемой сетевым адаптером. Сетевой адаптер может представлять собой печатную плату, которая вставляется в слот на материнской плате компьютера или же может быть интегрирован в материнскую плату.

**Пассивные сетевые устройства**

Под пассивными сетевыми устройствами подразумеваются сетевые устройства, функционирующие только на физическом уровне и не наделенные интеллектуальными способностями, то есть они никак не анализируют поступающие данные. Такими устройствами являются:

1. Повторитель (репитер) (сетевое устройство, которое воспринимает входящие сигналы и восстанавливает все их характеристики в исходное состояние, что позволяет передавать сигналы в физической среде на большее расстояние; при передаче сигналов происходит их затухание, то есть ослабление и искажение, но, а для устранения этой ошибки и используется повторитель);
2. Концентратор (хаб) (сетевое устройство, предназначенное для подключения в сет нескольких компьютеров или сетевых устройств; концентратор повторяет сигнал, полученный от одного из подключенных к нему устройств на все остальные устройства, подключенные к этому концентратору; концентраторы и повторители очень похожи, разница между ними состоит только лишь в количестве портов, поэтому концентраторы иногда называются многопортовыми повторителями).

**Активные сетевые устройства**

Активные сетевые устройства анализируют поступающие данные и, на основании этого анализа, принимают решение, как передавать полученные данные. К таким устройствам относятся коммутаторы, мосты, маршрутизаторы:

1. Коммутатор (свитч) (сетевое устройства, функционирующее на канальном уровне и предназначенное для подключения в сеть нескольких компьютеров или сетевых устройств; в отличие от концентратора, который рассылает данные через все свои порты, коммутатор старается передавать данные через порт получателя (коммутатор хранит в своей памяти таблицу коммутации, в которой указывается соответствие физического адреса компьютера, подключенного к коммутатору к его порту; после включения коммутатора таблица коммутации пуста, когда коммутатор принимает кадр – из заголовка кадра извлекаются адреса отправителя и получателя, адрес отправителя заносится в таблицу коммутации и для него указывается порт, с которого был принят кадр; в последствии, если будет получен кадр, адрес назначения которого уже есть в таблице коммутации, кадр будет передан, непосредственно, через порт, указанный в таблице коммутации; если же адреса назначения нет в таблице коммутации, то кадр рассылается через все активные порты коммутатору, кроме порта, с которого он был получен);
2. Мост (сетевое устройство, функционирующее на канальном уровне и предназначенное для объединения сетей, получив кадр, мост определяет подсеть, в которую должен быть передан полученный кадр; мост передает кадры из одной подсети в другую, только если получатель и отправитель принадлежат разным подсетям; мосты и коммутаторы похожи в том, как они анализируют входящие кадры, однако мосты не предназначены для подключения в сеть компьютеров или других сетевых устройств; они используются исключительно для объединения сети);
3. Маршрутизатор (роутер) (сетевое устройство, функционирующее на сетевом уровне и предназначенное для объединения сетей; задачами маршрутизатора являются анализ входящих пакетов, выбор наилучшего пути передачи пакетов по сети и их коммутация на соответствующий выходной порт).

**лекция №3 17.09.2016**

**Линии и каналы связи**

При построении сетей применяются линии связи. Линия связи представляет собой совокупность устройств передачи данных и физической среды, по которой передаются информационные сигналы.

В зависимости от физической среды, линии связи можно разделить:

1. кабельные линии связи;
2. беспроводные линии связи.

Линии связи являются основной частью канала связи.

Канал связи (канал передачи данных) – это средство передачи данных по линии связи. Иногда один канал связи может включать в себя несколько линий связи, но чаще всего одна и та же линия связи используется несколькими каналами связи.

Каналы связи делятся на несколько типов в зависимости от того, могут ли они передавать данные в обоих направлениях или нет:

* дуплексный канал (канал, обеспечивающий одновременную передачу данных в обоях направлениях);
* полудуплексный канал (канал, обеспечивающий поочередную передачу данных в обоих направлениях, то есть в определенный момент времени передача осуществляется только в одном направлении);
* симплексный канал (канал, обеспечивающий одновременную передачу только в одном направлении).

Каналы связи образуют первичные и вторичные сети:

* первичная сеть представляет собой совокупность каналов без подразделения их по назначению и видам связи;
* местная первичная сеть (часть первичной сети, ограниченная территорией города или района);
* внутризоновая первичная сеть (часть первичной сети, обеспечивающая соединение между собой каналов связи разных местных первичных сетей одной зоны);
* магистральная первичная сеть (часть первичной сети, обеспечивающая соединение между собой каналов связи разных внутризоновых первичных сетей).
* вторичная сеть (состоит из каналов одного назначения, образуемых на базе первичной сети; например, телефонные и телеграфные каналы, телевизионные каналы и.т.д.).

**Кабельные линии связи**

Имеют довольно сложную структуру, кабель состоит из проводников, заключенных в несколько слоев изоляции.

Три вида кабелей:

* витая пара (кабель, состоящий из одной или нескольких пар изолированных медных проводов, скрученных между тобой; скручивание проводов позволяет уменьшить влияние на них электромагнитных помех; в зависимости от структуры проводов, из которых состоит, витая пара их разделяют на:
* одножильный кабель (состоит из проводов, у которых сечение образовано одним проводником – жилой; следует отметить, что одножильный кабель не терпит перегибов);
* многожильный кабель (состоит из проводов, у которых сечение образовано несколькими, иногда переплетенными между собой, жилами; многожильные кабели обладают большей гибкостью и эластичностью, что приводит к большему затуханию сигналу; обычно многожильные кабели используются на маленьких расстояниях);
* коаксиальный кабель (кабель, предназначенный для передачи высокочастотных сигналов; коаксиальный кабель состоит из следующих элементов:
* оболочка изоляционного материала, которая служит от изоляции и защиты от внешних воздействий;
* экран в виде медной оплетки или алюминиевой фольги, или их комбинации;
* изоляция, выполненная в виде сплошного диэлектрического заполнения, которая обеспечивает постоянство взаимного расположения экрана и центрального проводника;
* центральный проводник в виде одножильного или многожильного центрального провода.);
* оптоволоконный кабель (кабель, состоящих из тонких зыбких стеклянных волокон, по которым распространяются световые лучи, а также из армирующих и защитных покровов; оптоволоконный кабель не восприимчив к электромагнитным помехам и способен обеспечивать высокую передачу, в отличии от витой пары или коаксиального кабеля; каждое волокно состоит из стеклянного сердечника и стеклянной оболочки, так как стеклянная оболочка обладает меньшим показателям преломления, чем сердечник, то распространяясь по сердечнику, световые лучи не выходят за его пределы, отражаясь от оболочки).

**Разновидности витой парой**

Для обеспечения лучшей защиты от электромагнитных помех используют экранирование. В зависимости от наличия экрана определяют следующие разновидности:

1. Незащищенная витая пара UTP, U/UTP (витая пара, в которой отсутствует защитный экран);
2. Защищенная витая пара STP, U/FTP (витая пара, в которой имеется индивидуальный экран в виде алюминиевой фольги);
3. Фольгированная витая пара FTP, F/UTP (витая пара, в которой имеется один общий экран в виде алюминиевой фольги);
4. Незащищенная экранированная витая пара SF/UTP (витая пара, в которой имеется двойной медный экран из медной оплетки и алюминиевой фольги);
5. Защищенная экранированная витая пара S/STP (витая пара, в которой имеется один общий экран и индивидуальный экран для каждой пары, выполненный в виде медной оплетки);
6. Фольгированная экранированная витая пара S/FTR (витая пара, в которой имеется один общий экран, выполненный в виде медной оплетки, а также индивидуальный экран для каждой пары из алюминиевой фольги).

ISO/IEC 11801:2002

XX/XXX

U – общий экран отсутствует

F – общий экран состоит из фольги

S – общий экран имеет вид медной оплетки

SF - двойной общий экран, медная оплетка плюс фольга

TP – витая пара

U – индивидуальный экран пары отсутствует

F – индивидуальный экран пары состоит из фольги

S – индивидуальный экран пары имеет вид медной оплетки

**Категории витой пары**

Витая пара различается на несколько категорий, причем, чем выше категория, тем выше скорость передачи, а также больше пар проводов и больше витков на единицу длины.

* Cat.1 (имеет 1 пару, скорость передачи до 1 Мбит/с, используется при построении телефонных и модемных линий связи);
* Cat.2 (имеет 2 пары, скорость передачи до 4 Мбит/с, сейчас не используется);
* Cat.3 (имеет 4 пары, скорость передачи до 10/100 Мбит/с, по технологии 100Base-T4 способен передавать данные со скоростью до 100 Мбит/с, используется при построении телефонных и компьютерных сетей);
* Cat.4 (имеет 4 пары, скорость передачи до 16 Мбит/с, сейчас не используется);
* Cat.5 (имеет 4 пары, скорость передачи до 100 Мбит/с, используется при построении телефонных и компьютерных сетей);
* Cat.5e (имеет 2/4 пары, скорость передачи до 100/1000 Мбит/с, используется при построении компьютерных сетей);
* Cat.6 (имеет 4 пары, скорость передачи до 1 Гбит/с, используется при построении компьютерных сетей);
* Cat.6a (имеет 4 пары, скорость передачи до 10 Гбит/с, используется при построении компьютерных сетей);
* Cat.7 (имеет 4 пары, скорость передачи до 10 Гбит/с, фактически то же самое, что и витая пара S/FTP Cat.6a, используется при построении компьютерных сетей).

**Обжим витой пары**

Витая пара подключается к оборудованию с помощью разъемов стандартно RJ – Registered Jack. Для фиксации разъемов в ответной части предусмотрена специальная защелка, расположенная на корпусе разъема. Например:

* RJ-11 [6P4C] (имеет 6 позиций, 4 контакта, используется при построении телефонных сетей);
* RJ-12 [6P6C] (имеет 6 позиций, 6 контактов, используется при построении телефонных сетей);
* RJ-45 [8P8C] (имеет 8 позиций, 8 контактов, используется при построении компьютерных сетей);
* RJ-50 [10P10C] (имеет 8 позиций, 8 контактов, используется при построении компьютерных сетей).

Все разъемы стандарта RJ производятся в неэкранированном и экранированном исполнении. Экранированные разъемы в отличии от неэкранированных имеют металлический кожух, который соединяется с экраном витой пары.

Все разъемы стандарта RJ допускают только однократную установку и для их обжима используются специальные обжимные клещи (кримпер).

Существует два стандарта обжима витой пары:

* EIA/TIA-568A (зелено-белый, зеленый, оранжево-белый, синий, сине-белый, оранжевый, коричнево-белый, коричневый);
* EIA/TIA-568B (оранжево-белый, оранжевый, зелено-белый, синий, сине-белый, зеленый, коричнево-белый, коричневый).

Для соединения компьютер-коммутатор используется «прямой» кабель, обжатый с обеих сторон одинаково. А для соединения компьютер-компьютер или коммутатор-коммутатор используется «перекрестный кабель», с одной стороны обжатый по стандарту EIA/TIA-568A, а с другой по стандарту EIA/TIA-568В. Чаще всего используется EIA/TIA-568В.

Витая пара, обжатая не по стандарту, может привести к нетрудоспособности сети.

**Типы коаксиальных кабелей**

1. Толстый коаксиальный кабель [RG-8, RG-11] (обладает волновым сопротивление 50 Ом, внешним диаметров примерно 12 мм и достаточно толстым центральным проводником 2,17 мм; имеет хорошие механические и электрические характеристики, однако этот кабель сложно монтировать, так как он весьма сложно гнется);
2. Тонкий коаксиальный кабель [RG – 58] (обладает волновым сопротивлением 50 Ом, внешним диаметров примерно 6 мм и тонким центральным проводником 0,89 мм; не так прочен, как толстый коаксиальный кабель, и его механические и электрические характеристики хуже, но зато он обладает большей гибкостью);
3. Телевизионный кабель [RG-6, RG-59] (обладает волновым сопротивлением 75 Ом, и широко распространяется в кабельном телевидении, также существуют стандарты, позволяющие использовать такой коаксиальный кабель в информационных сетях);
4. Твинаксиальный кабель (коаксиальный кабель с двумя, переплетенными между собой изолированными проводниками; волновое сопротивление составляет 150 Ом (по 75 Ом на каждый проводник), также существует разновидность твинаксильаного кабеля с 4 проводниками, который по своим возможностям похож на витую пару категории 5).

**Монтаж коаксиального кабеля**

Для подсоединения коаксиального кабеля к оборудованию используются специальные разъемные соединения, которые можно разделит на три группы:

1. Для пайки;
2. Под отжим;
3. Навинчивающиеся.

Перед подключением коаксиального кабеля к оборудованию необходимо его разделит и зачистить.

наиболее распространен вариант с использование обжима (на концах кабеля монтируются специальные разъемы; соединение отдельных отрезков коаксиального кабеля используются с помощью разъема BNC и BNC – i

**Оптоволоконный кабель**

* Одномодовое волокно (используется сердечник очень малого диаметра, при этом луч распространяется практически вдоль оптической оси сердечника, не отражаясь от внешней оболочки СХЕМА; используются для передачи данных со сверхвысокими скоростями в несколько десятков Гбит/с на расстояние до нескольких сотен километров, однако их изготовление представляет собой очень сложный технологический процесс, что делает его дорогим);
* Многомодовое волокно (используется более широкий сердечник, в котором одновременно существуют несколько световых лучей, отражающихся от внешней оболочки под разными углами СХЕМА; многомодовые волокна делятся на два типа:
* волокна со ступенчатым показателям преломления;
* волокна с плавным показателем преломления).

В качестве источников света в оптоволоконных проводах применяются светодиодные излучатели – лазеры. Для одномодовых кабелей, как правило, применяются только лазеры, а светодиодные излучатели используются только для многомодовых.

**Монтаж оптоволоконного кабеля**

Оптоволоконный кабель подключается к сетевому оборудованию с помощью оптических разъемов типа FC, SC, ST. Для сращивания отдельных частей оптоволоконных кабелей используется сварка его оптических волокон. Для сварки оптических волокон используются специальные сварочные аппараты, которые упрощают процесс соединения волокон с наименьшими потерями. Процесс сварки включает в себя: нагревания и плавление концов волокон с их последующим точным соединением.

**Беспроводные линии связи**

Строятся в соответствии с простой схемой, каждый узел оснащен антенной, который, одновременно, является передатчиком и приемником электромагнитных волн. Электромагнитные волны распространяются в атмосфере во всех направлениях или же в пределах определенного сектора. Направленность или не направленность зависит от типа антенны.

Параболическая антенна является направленной, а изотропные антенны представляют собой вертикальный проводник и являются ненаправленными, так как при ненаправленном распространении все пространство заполняется в пределах определенного радиуса, то это пространство может служить разделяемой средой.

**Радиосвязь**

Хотя природа всех радиоволн одинакова, их способность к распространению сильно зависит от длины волны и частоты.

Выделяют основные частотные диапазоны:

1. длинные волны (частотный диапазон: длиной 1-10 км, частота 300-30 кГЦ; могут распространяться на большие расстояния, так как способны огибать земную поверхность);
2. средние волны (частотный диапазон: длиной 100 м - 1 км, частота 3 МГц – 300 кГц; распространяются на небольшие расстояния, поскольку волны могут отражаться только от иона сферы);
3. короткие волны (частотный диапазон: длиной 10-100 м, частота 30-3 МГц; многократно отражают от поверхности земли и от ионосферы, благодаря чему распространяются на большие расстояния);
4. ультракороткие волны (частотный диапазон: длиной 0,1-10 м, частотой 3 ГГц – 30 МГц; распространяются только на расстояния прямой видимости).

**Спутниковая связь**

Один из видов радиосвязи, основанный на использовании, в качестве ретрансляторов, искусственных спутников Земли.

Спутник в себя несколько транспондеров. Транспондер – это приемно-передающее устройство, которое посылает сигнал в ответ на принятый сигнал. Каждый из транспондеров настроен на определенную частоту радиоволн. При передаче сигнала на Землю транспондер использует частоту такую, чтобы отправляемый сигнал не накладывался на получаемый.

Спутниковая связь осуществляется между земными станциями, которые могут быть как стационарные, так и подвижные.

Зоны, в котором возможен прием спутникового сигнала, называется зоной покрытия.

**Базовые сетевые технологии**

Сетевая технология – согласованный набор протоколов канального уровня (2 уровень) и реализующих их программно-аппаратных средств, достаточных для построения сети. Кроме того, сетевая технология определяет топологии сети, физическую среду передачи, формат сетевых кадров, способ кодирования сигналов и скорость передачи данных.

Большинство сетевых технологии стандартизировано серией стандартов IEEE 802 (институт инженеров по электротехнике и электронике).

**Технология Ethernet**

Является наиболее популярной сетевой технологией проводных сетей. В сетях применяется топология общая шина или звезда.

В качестве метода доступа к среде передачи используется метод доступа CSMA/CD.

**Метод доступа CSMA/CD:** в сетях с разделяемой средой, к которой имеют доступ все узлы сети, используется метод доступа к среде передачи, называемый метод доступа с распознаванием несущей и обнаружением коллизий. Канал связи может использоваться в режиме коллективного доступа сразу несколькими сетевыми устройствами, поэтому необходимо определить, что канал связи свободен, прежде, чем передавать кадр (совокупность данных на канальном уровне). Это достигается путем прослушивания и анализа возникающих в канале сигналах.   
Если опознается несущая частота, канал связи считается занятым и передача кадра откладывается до окончания чужой передачи. Кроме того, после окончания передачи, все узлы обязаны выдержать небольшую дополнительную паузу, называемую межкадровым интервалом. Эта пауза позволяет узлам принять и обработать кадр, а затем начать передачу своего кадра. Однако, механизм прослушивания каналов связи и пауза между кадрами не гарантирует исключения ситуации, в которой два или более узла сети решат, что канал связи свободен и начнут передавать свои кадры. Говорят, что при этом происходит коллизия.

Коллизия - состояние, когда кадры сталкиваются в разделяемой среде, что приводит к искажению передаваемой информации. При обнаружении коллизии, узел отправляет по каналу связи специальную 32-битную последовательность, называемую jam- последовательностью для более надежного распознавания коллизии всеми узлами сети (условия распознавания коллизии зависят от используемой физической среды). После обнаружения коллизии, каждый узел сети, который передавал кадр и столкнулся с коллизией, делает некоторую паузу и повторяет попытку, пока кадр не будет отправлен или пока не будет достигнуто максимальное число попыток.

**Формат кадра Ethernet**

На практике в сетях Ethernet используются кадры различных форматов.

* SNAP (протокол доступа к подсетям);

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | SFD | DA | SA | L | DSAP | SSAP | Control | OUI | T | Данные | FCS |

* Поле Preamble имеет 7 байт и принимает значения в двоичной системе для каждого байта 10101010. С помощью Preamble выполняется синхронизация кадров в сети;
* Поле Start of Frame Delimiter – это начальный ограничитель кадров длиной в 1 байт. Этот байт имеет значение 10101011 в двоичной системе и указывает на то, что следующий байт – это первый байт заголовка MAC;
* Поля Destination Address и Sourse Address длиной 6 байт каждый, содержат соответственно адрес получателя и адрес отправителя;
* Поле Length имеют длину 2 байта и содержит размер кадра, не считая таких полей, как P, SFD, DA, SA, L, FCS. Данное поле может принимать значения от 46 до 1500;
* Поля Destination Service Access Point и Sourse Service Access Point соответственно адрес точки доступа службы получателя и адрес точки доступа службы отправителя, оба этих поля имеют длину 1 байт, и предназначены для того, чтобы указать, какие протоколы верхнего уровня передают и получают данные с помощью этого кадра;
* Поле Control имеет длину 1 байт и зависит от того, в каком режиме управления логическим каналом LLC реализуется в процессе передачи данных;
* Полет Organizationally Unique Identefier имеет длину 3 байта и содержит идентификатор организации, которая контролирует входы протоколов, указанных в поле Type;
* Поле Type длиной 2 байта, предназначенный для тех же целей, что и поля DSAP и SSAP, то есть, указания протокола верхнего уровня;
* Поле Данные имеет переменную длину от 38 до 1492 байт, если размер передаваемых данных менее 38 байт, то поле Данные дополняются до минимально допустимых 38 байт, что позволяет канальному уровню выявлять конфликты в сети;
* Поле Frame Check Sequence длиной в 4 байта, содержит контрольную сумму, вычисленную для своего кадра по алгоритму CRC32, по значению данного поля получатель кадров может определить: были ли данные искажены в процессе его передачи.

**Спецификация Ethernet**

Технология Ethernet определяет большое количество спецификаций, которые описываю используемые при построении сети топологию, физическую среду и скорость передачи данных. В зависимости от скорости передачи данных, существует несколько вариантов данной технологии:

* Ethernet (до 10 Мбит/с);
* Fast Ethernet (до 100 Мбит/с);
* Gigabit Ethernet (до 1 Гбит/с);
* 10Gigabit Ethernet (до 10 Гбит/с).

**Ethernet 10Base – 5:** спецификация Ethernet для построения сетей на основе толстого коаксиального кабеля с топологией общая шина; данная идентификация описана стандартом IEEE 802.3; обозначение Ethernet 10Base – 5 говорит о том, что максимальная скорость передачи данных 10 Мбит/с, а максимальнаядлина сегмента 500 м; словом base указывает на то, что передаваемый сигнал не модулируется;

Для получения компьютера к толстому коаксиальному кабелю используются трансиверы, которые устанавливаются на коаксиальном кабеле не ближе 2,5 м друг от друга; трансивер соединяется с сетевым адаптером компьютера через трансиверный кабель с разъемом DB-15, длина трансиверного кабеля должна быть не менее 2 м и не более 50 м; трансиверы подключаются к толстому коаксиальному кабелю при помощи разъемов, имеющих названия «вампиры» - это из-за того, что при подключении разъем прокалывает кабель до центральной жилы. На концах толстого коаксиального кабеля устанавливаются терминаторы; без терминаторов сигнал отражается от концов кабеля из-за чего передача данных по кабелю становится невозможным; участок сети между двумя терминаторами называют сегментами; сегменты могут соединяться с помощью повторителей, при этом общая длина сети не может превышать 2,5 км. Один из сегментов должен быть заземлен с помощью терминатора, имеющего цепочку и контакт на конце.

**Ethernet 10Base – 2:** спецификация Ethernet для построения сетей на основе тонкого коаксиального кабеля с топологией общая шина; данная спецификация описано в стандарте IEEE 802.3a; компьютер подключается к тонкому коаксиальному кабелю с помощью специальных разъемов BNC – T, на концах разъемов BNC – T могут подключаться либо терминатор, либо коаксиальный кабель, обжатый разъемом BNC; максимальное число компьютеров в сегменте 30 штук, а минимальное расстояние между ними 0,5 м; допустимая максимальная длина сегмента 185 м, сегменты могут соединяться с помощью повторителей, общая длина сети не должна превышать 925 м (5 сегментов и 4 повторителя).

**Ethernet 10Base – T:** спецификация Ethernet для построения сетей на основе витой пары с топологией звезда; данная спецификация описана стандартом IEEE 802.3i; для подключения компьютера к концентратору используются витая пара категории 3, длиной не более 100 м; для связи используется 2 пары: одна на прием, другая на передачу данных. Витая пара подключается к компьютеру с помощью разъема RJ-45; участок сети, охватывающий концентратор со всеми подключенными к нему компьютерами, называется сегментом сети;

**Ethernet 10Base – FL:** спецификация Ethernet для построения сетей на основе оптоволоконного кабеля с топологией звезда; данная спецификация описана стандартом IEEE 802.3j; компьютер подключается к оптоволоконному кабелю с помощью оптоволоконного трансивера, оптоволоконные кабели подключаются к трансиверу и к концентратору с помощью разъема SC и ST; оптоволоконный трансивер соединяется с сетевым адаптером компьютера с помощью трансиверного кабеля длиной не более 25 м; для связи концентратора с трансивером используются два оптоволоконных кабеля длиной не более 2 км; один на прием данных RX, а второй на передачу данных TX; длина оптоволоконного кабеля, соединяющего трансивер и концентратор, может достигать 2 км; применяется многомодовый оптоволоконный кабель и свет длиной волны 850 нм.

В **Ethernet 10Base – FL** несколько концентраторов могут объединятся между собой для построения топологии «дерево», при этом максимальное число концентраторов между любыми двумя трансиверами не должно превышать четырех.

**FAST ETHERNET**

Все спецификации FAST ETHERNET описаны стандартом IEEE 802.3u и предназначены для построения сетей топологии «звезда».

1. 100Base-TX (максимальная скорость 100 Мбит/с): спецификация FAST ETHERNET для построения сетей на основе витой пары; для подключения компьютера к коммутатору используется, витая пара 5 категории (Cat.5) длиной не более 100 м; для связи используется две пары: одна пара передачи, другая пара для приема;
2. 100Base-T4: спецификация FAST ETHERNET для построения сетей на основе витой пары; для подключения компьютера к коммутатору используется, витая пара 3 категории длиной не более 100 м; передача данных осуществляется в полудуплексном режиме с использованием четырех пар: три пары используются для передачи данных, а одна пара для выявления ошибок при передаче;
3. 100Base-FX: спецификация FAST ETHERNET для построения сетей на основе многомодового оптоволоконного кабеля; подключения компьютера к концентратору осуществляется с помощью двух оптоволоконных кабелей: один на прием, другой на передачу; оптоволоконный кабель подключается с помощью разъема SC; при дуплексной связи максимальное расстояние между компьютером и концентратором 2 км, а при полудуплексном 412 м.

**Gigabit Ethernet**

Почти все спецификации Gigabit Ethernet предназначены для построения сетей с топологией «звезда».

1. 1000Base-T: спецификация Gigabit Ethernet для построения сетей на основе витой пары категории 5 или выше; максимальное расстояние между компьютером и коммутатором – 100 м; данные передаются по 4 парам проводов: две на прием данных, две на передачу;
2. 1000Base-SX: спецификация Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; данная спецификация предусматривается использование коротковолнового лазера 850 нм, который обеспечивает передачу данных многомодовому оптоволоконному кабелю на расстояние 550 м;
3. 1000Base-LX: спецификация Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; данная спецификация предусматривает использование длинноволнового лазера 1300 нм, который обеспечивает передачу по многомодовому оптоволоконному кабелю до 550 м, по одномодовому до 5 км;
4. 1000Base-CX: спецификация Gigabit Ethernet для соединения сетевого оборудования; предусматривает использование короткого триансиального кабеля до 25 м;

**10Gigabit Ethernet**

Большинство спецификаций 10Gigabit Ethernet описано стандартом IEEE 802.3ae. Предназначено для построения сетей с топологией «звезда».

1. 10GBase-SR: спецификация 10Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; предусматривает использование коротковолнового лазера 850 нм, который обеспечивает передачу по многомодовому оптоволоконному кабелю на расстояние до 300 м;
2. 10GBase-LR: спецификация 10Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; данная спецификация предусматривает использование длинноволнового лазера 1300 нм, который обеспечивает передачу по одномодовому оптоволоконному кабелю на расстояние до 10 км;
3. 10GBase-ER: спецификация 10Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; предусматривает использование длинноволнового лазера 1550 нм, который обеспечивает передачу по одномодовому оптоволоконному кабелю на расстояние до 40 км.
4. 10GBase-LX4: спецификация 10Gigabit Ethernet для построения сетей на основе оптоволоконного кабеля; предусматривает использование длинноволнового кабеля 1300 нм, который обеспечивает передачу по многомодовому оптоволоконному кабелю на расстояние до 300 м и одномодовому до 10 км;
5. 10GBase-CX4: спецификация 10Gigabit Ethernet для соединение сетевого оборудования с помощью короткого коаксиального кабеля до 15 м, при этом используется твиаксиальный кабель, у которого 4 пары медных проводов;
6. 10GBase-T: спецификация 10Gigabit Ethernet для построения сетей на основе витой пары; для подключения компьютера к коммутатору используется витая пара категории 6a не более 100 м или витая пара просто 6 категории не более 550 м.

**Технологии Token Ring и FDDI**

Эти технологии также, как и Ethernet применяется для построения проводных сетей, однако, они являются функционально более сложными технологиями. В сетях используется маркерный метод доступа в среде передачи данных.

**Маркерный метод доступа**

Заключается в том, что в сети циркулирует специальный кадр, называемый маркером. Владение этим маркером гарантирует право передачи данных по сети.

Если узел, принимающий маркер, не имеет данных для отправки, он просто передает маркер следующему узлу. Каждый узел может удерживать маркер в течении определенного максимального времени.

Если узел, владеющий маркером, имеет данные для передачи, он захватывает маркер, преобразовывает его в информационный кадр, и передает этот информационный кадр следующему узлу сети.

Когда информационный кадр передается по сети, маркер отсутствует, поэтому другие узлы, желающие передать данные вынуждены ожидать.

Новый маркер будет создан только после завершения передачи данных, когда информационный кадр достигнет узла назначения.

Отметим, что в сетях с маркерным доступом не может возникнуть коллизий.

**Технология Token Ring**

Предусматривает построение сети на основе витой пары с физической топологией «звезда» и с логической топологией «кольцо».

На основе данной технологии был принят стандарт IEEE 802.5.

Сети Token Ring поддерживают две скорости передач: 4 и 16 Мбит/с. В сетях Token Ring каждый компьютер подключается к специальному устройству множественного доступа MAU, если произойдет, обрыв витой пары, то разорванное кольцо будет восстановлено путем включения внутри MAU обходного пути.

**Технология FDDI**

Эта технология позволяет создать сеть на основе оптоволоконного кабеля с топологией двойное кольцо. Одно кольцо считается основным (по нему передаются данные в обычном состоянии), а второе кольцо является вспомогательным (по нему данные передаются в случае отказа на первом кольце). При этом данные в кольцах циркулируют в разных направлениях (в разных направлениях для «обхода» вышедшего из строя устройства). Поскольку использование двух колец повышает надежность сети, технология FDDI успешно использовалась в магистральных каналах связи. Скорость передачи составляет 100 Мбит/с.

**БЕСПРОВОДНЫЕ ТЕХНОЛОГИИ**

В настоящее время существует достаточно много сетевых технологий для построения беспроводных сетей, среди которых с наибольшим распространением пользуются такие технологии, как Bluetooth и Wi-Fi.

**Bluetooth**

Технология Bluetooth предназначена для создания беспроводной персональной сети. В основном, Bluetooth используется для связи между мобильными телефонами, связи мобильных телефонов с компьютерами или для подключения беспроводных устройств к компьютерам. Но при необходимости Bluetooth можно использовать для соединения компьютеров.

**Архитектура Bluetooth:** посредством Bluetooth устройства объединяются в пикосеть, состоящую из одного ведущего узла и нескольких (до 7) ведомых узлов. Ведущий узел пикосети синхронизирует работу всех ведомых узлов, но, кроме 7 ведомых узлов, в пикосети допускается наличие до 255 отдыхающих ведомы узлов – они не учувствуют в обмене данными, но при этом находятся в синхронизации с ведущим узлом.

Несколько пикосетей (до 10) можно распределить в распределенную сеть. Для этого определенные ведомые узлы одной пикосети должны выполнять роль ведущего в другой. Отметим, что узлы из разных пикосетей, входящих в распределенную сеть, не могут взаимодействовать с друг другом.

**Передача данных в Bluetooth:** приемы и передачиBluetooth используют частотный диапазон от 2,4 до 2,4835 ГГц. При использовании данного диапазоны возникают две главные проблемы:

* большую проблему представляет собой высокий уровень помех из-за того, что в данном частотном диапазоне 2,4 ГГц работают много устройств различных беспроводных технологий, а также микроволновые печи; для противодействия помехам в Bluetooth совместно со схемой модуляции GFSK (Гауссовская манипуляция с частотным сдвигом) используется метод FHSS (псевдослучайная перестройка рабочей частоты), который заключается в скачкообразном переключении междудоступными каналами радиосвязи. При этом диапазон 2,4 ГГц разбивается на 79 каналов шириной 1 МГц, а смена каналов происходит 1600 раз в секунду. Схема переключения между каналами известна только приемнику и передатчику, которые синхронно переключаются от одного канала к другому. Таким образом, маловероятно, что работающие рядом Bluetooth-устройства будут друг другу мешать;
* радиоволны диапазона 2,5 ГГц легко поглощаются в окружающей среде, что ограничивает радиус действия приема передатчиков; данную проблему можно преодолеть с помощью усиления мощности приема передатчика.

Устройства Bluetooth делятся на 3 класса в зависимости от мощности и радиуса действия приемопередатчика. Чаще всего включаются устройства 2-ого класса (радиус действия 10 м). Также есть 1 класс (100 м) и 3 класс (1 м).

**Профили Bluetooth**

Каждое устройство Bluetooth поддерживает определенный набор профилей, которые определяют его функциональные возможности.

Чтобы задействовать какое-либо Bluetooth-устройство для выполнения определенной задачи, требуется наличие поддержки соответствующего профиля.

**Профили Bluetooth можно разделить на две группы:**

* базовые профили (поддерживаются всеми без исключения устройствами Bluetooth; к ним относятся GAP (профиль общего доступа; отвечает за поддержание связей между устройствами Bluetooth, выявление доступных профилей, а также за безопасность), GOEP (общий профиль обмена объектами; профиль определяет каким образом два устройства Bluetooth будут осуществлять обмен данными посредством протокола OBEX), SPP (профиль последовательного порта; определяет каким образом два устройства будут осуществлять обмен данными посредством эмуляции интерфейса USB, RS -232)) ;
* прикладные профили (существует огромное количество прикладных профилей, например, профиль DUN (профиль дозвона по сети; обеспечивает компьютеру доступ к сети Интернет посредством мобильного телефона), профиль HID (профиль устройств для взаимодействия с человеком; предназначен для подключения беспроводных устройств таких, как мышь, клавиатура, джойстик и.т.д.).

Существуют также профили Bluetooth для потоковой передачи аудио или видео от компьютера до беспроводных наушников или от цифровой видеокамеры к ТВ, также для передачи данных.

**Спецификации Bluetooth**

Все спецификации Bluetooth описаны стандартом IEEE 802.15.1.

В настоящее время наиболее распространенные спецификации Bluetooth:

* Bluetooth 2.0 + EDR (данная спецификация поддерживает технологию EDR, которая позволяет достичь скорости передачи 2,1 Мбит/с, теоретически 3 Мбит/с; данная спецификация поддерживает обратную совместимость со старыми спецификациями Bluetooth 1.0, Bluetooth 1.1, Bluetooth 1.2);
* Bluetooth 2.1 + EDR (является доработанной версией предыдущей спецификации, основными нововведениями являются введение NFX ( не более 10 см) и энергосберегающей технологии Sniff Subrating, которая позволила снизить энергопотребление в 3-10 раз по сравнению со старыми спецификациями);
* Bluetooth 3.0 + HS (высокая скорость) (данная спецификация поддерживает технологию AMP, позволяющую передавать данные со скоростью до 24 Мбит/с, также предусматривает использование двух приемопередатчиков: одного низкоскоростного со скоростью до 3 Мбит/с с невысоким энергопотреблением и одного высокоскоростного со скоростью до 24 Мбит/с и совместимого со стандартом IEEE 802.11; при передачи данных может использовать как низкоскоростной передатчик, так и высокоскоростной передатчик – это позволяет снизить уровень энергопотребления в тех случаях, когда не требуется высокая скорость передачи данных);
* Bluetooth 4.0 (поддерживает возможности предыдущих спецификаций Bluetooth, но, кроме того, предусматривает возможность передачи данных с низким энергопотреблением; снижение электропотребления достигается за счет снижения передачи данных (не более 1 Мбит/с), так и за счет того, что приемопередатчик не работает постоянно, а включается только на время приема данными; данная спецификация используется главным образом в миниатюрных цифровых устройствах и различных электронных датчиках);
* Bluetooth 4.1 (в основном, содержит улучшения возможностей предыдущих спецификаций, например, может взаимодействовать с беспроводной технологией LTE, чтобы координировать передачу данных и уменьшать вероятность помех).

**08.10.2016**

**Wi-Fi**

Технология Wi-Fi предназначена для беспроводных локальных сетей.

**Архитектура Wi-Fi**

Сети Wi-Fi могут использоваться в нескольких режимах:

1. «каждый с каждым» режим (компьютеры могут связываться непосредственно с друг другом без использования дополнительного оборудования; данный режим используется для создания временных сетей; для создания долговременных беспроводных сетей используются другие режимы СХЕМА);
2. инфраструктурный режим (для создания беспроводной сети используется точка доступа, подключенная к беспроводной сети СХЕМА; точка доступа одновременно служим коммутатором, который объединяет компьютеры в беспроводную сеть, и мостом, который объединяет беспроводную и проводную сеть);
3. распределенная беспроводная система (в режиме распределенной беспроводной системе точки доступа соединяются между собой, образуя мостовое соединение между проводными сетями, при этом точкам доступа, образующим мостовое соединение, подключение к компьютеру невозможно; данный режим служит только для объединения сетей и применяется там, где прокладка кабеля нежелательна или невозможна СХЕМА);
4. распределенная беспроводная система с точкой доступа (СХЕМА; в режиме распределенной беспроводной системы с помощью точек доступа можно организовать не только мостовое соединение, но и одновременно подключить к ним компьютеры, образуя беспроводную сеть).

**Стандарты Wi-Fi**

Основным стандартом беспроводных локальных сетей является стандарт IEEE 802.11, который предполагает использование диапазон 2,4 ГГц, а для противодействия помехам используется FHSS совместно с частотной модулей FSK или метод DSSS совместно с фазовой модуляцией PSK. Максимальная скорость передачи, согласно этому стандарту, составляет от 1 до 2 Мбит/с. Радиус сети теоретически может достигать 300 м, но на практике редко превышает 100 м.

В процессе развития Wi-Fi постепенно появилось несколько стандартов, дополняющих стандарт IEEE 802.11, эти стандарты можно разделить на три группы:

1. IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, IEEE 802.11ac (эти стандарты описывают работу беспроводного сетевого оборудования);
2. IEEE 802.11d, IEEE 802.11e, IEEE 802.11h, IEEE 802.11i, IEEE 802.11j, IEEE 802.11k, IEEE 802.11r (эти стандарты описывают параметры беспроводной среды, частоты каналов радиосвязи, средства безопасности и тому подобное);
3. IEEE 802.11f, IEEE 802.11c (эти стандарты описывают принципы взаимодействия между собой точек доступа, также организацию беспроводного мостового соединения и тому подобное).

**Метод доступа CSMA/CA**

Метод коллективного доступа с распознаванием несущей и предотвращения коллизий.

* узлы используют периоды молчания случайной длины, чтобы избежать коллизий;
* для того, чтобы узлы могли узнавать о коллизиях, которые распознать невозможно, применяется схема с подтверждением отправленного кадра.

СХЕМА

Когда канал связи свободен перед отправкой кадра, узел начинает период молчания, за исключением случаев, когда узел давно не использовал канал связи. При этом отсчет периода молчания приостанавливается на время занятости канала, когда узел завершит период молчания, он отправит свой кадр. Если кадр отправлен успешно адресат отправляет обратно подтверждение, но, если подтверждение отсутствует, то произошла ошибка (возможно, коллизия). В таком случае отправитель удваивает период молчания и повторяет попытку, пока кадр не будет доставлен или пока не будет достигнуто максимальное число попыток.

**Проблема скрытого узла**

В беспроводных сетях возможно возникновение проблемы скрытого узла, которое заключается в том, что не все узлы могут слышать друг друга, а передача кадра, идущая в одну часть беспроводной сети, может быть не воспринята узлом другой её части. СХЕМА

Может возникнуть ситуация, когда узел А передает данные точке доступа, а узел Б, прослушивая канал связи, делает ложный вывод о том, что канал связи свободен и можно начать передачу данных. Такое решение приведет к тому, что, скорее всего, возникнет коллизия.

Решением проблемы скрытого узла в Wi-Fi подразумевается, что каждый передаваемый кадр содержит поле NAV (вектор распределения сети), в котором указывается как долго будет передавать данные отправитель кадров. Все узлы, получившие кадр, понимают, что канал связи будет занят в течении некоторого времени, указанное в поле NAV, даже если не удается распознать несущую частоту. Кроме того, в поле NAV включается также время, необходимое для отправки подтверждения.

**АДРЕСАЦИЯ В ИНФОРМАЦИОННЫХ СЕТЯХ**

Для однозначной адресации в информационных сетях используются физический и сетевой адреса.

* **Физический адрес** (адрес, который используется некоторой сетевой технологией для адресации в пределах одной сети, однако существуют технологии, в которых используются другие физические адреса);
* **Сетевой адрес** (адрес, состоящий из идентификатора сети и идентификатора узла; сетевые адреса решают задачу объединения сетей, независимо от способа адресации в отдельных сетях, например, в сетях TCP/IP использует IP-адрес);

**Доменное имя** – это символьная форма адресации, которая является наиболее удобной для человека, нежели сетевой адрес. Однако, компьютеры работают с сетевыми адресами, поэтому для установления соответствия между доменным именем и сетевым адресом используются специальные таблицы.

**MAC-адрес**

- это физический адрес, который присваивается сетевому интерфейсу во время изготовления устройства и используется для передачи сетевых кадров. MAC-адрес представляет собой 48-битное число, которое записывается в виде 6 пар по 16-ричных цифр. Чтобы гарантировать однозначную адресацию сетевых интерфейсов, было предложено разделить MAC-адрес на несколько частей:

* первые два бита – флаги:
* I/G (указывает, является ли MAC-адрес индивидуальным или групповым);
* U/L (указывает, является ли MAC-адрес универсальным или локально-управляемым).

Универсальные MAC-адреса присваиваются производителем сетевых адаптеров и сетевых устройств, а локально-управляемые – администратором сети.

* OUI (уникальный 22-битный идентификатор организации, который выдается организацией IEEE каждому производителю сетевых адаптеров и устройств);

Каждый производитель отвечает за правильность присвоения значений 24-битного поля OUA, которые должны быть уникальными для созданных им устройств.

Существует также широковещательный MAC-адрес, который состоит только их двоичных единиц. Если кадр отправляется с широковещательным MAC-адресом, то его получают все узлы сети.

**IP-адрес**

- это сетевой адрес, который используется в протоколе IP, который входит в состав стека протоколов TCP/IP.

Существует несколько протоколов IP:

* IPv4 (адрес представляет собой 32-битное число, которое записывается в виде четырех десятичных чисел от 0 до 255, разделенных точками, например, 192.268.6.13;
* IPv6 (адрес представляет собой 128-битное число, которое записывается в виде восьми групп, разделенных между собой двоеточиями, по четыре 16-ричных цифр в каждой группе, например, FC00:0000:0000:0000:0123:4567:89AB:CDEF; поскольку многие адреса могут содержать большое количество нулей, то разрешены несколько методов сокращенной записи: 1) в каждый группе могут быть опущены ведущие нули; 2) одна или более групп, состоящие полностью из нулей, могут заменяться парой двоеточия, например, FC00: :0123:4567:89AB:CDEF).

**Классовая и бесклассовая адресация**

Поскольку IP-адрес является сетевым адресом, то он состоит из идентификатора сети и идентификатора узла. Запись IP-адреса не предусматривает специального различительного знака между идентификатором сети и идентификатором узла.

Существует два метода для определения какая часть адреса относится к идентификатору сети, а какая к идентификатору узла:

1. классовая адресация;
2. бесклассовая адресация.

**Классовая адресация**

IP-адресация осуществлялась на основе 5 классов. Первые биты определяли класс сети, по которому можно было сказать – сколько бит IP-адреса отведено под идентификатор сети и под идентификатор узла.

Класс А – к нему относятся все адреса, старший бит которых равен нулю; в этом классе под идентификатор сети относятся 7 бит, а под идентификатор узла 24 бита; IP-адреса класса А лежат в диапазоне 1.0.0.0 – 126.255.255.255; максимально количество узлов 16777214;

Класс В – к нему относятся все адреса, старшие биты которых имеют значение 1,0(один ноль); в этом классе под идентификатор сети относятся 14 бит, а под идентификатор узла 16 бит; IP-адреса класса В лежат в диапазоне 128.0.0.0 – 191.2555.255.255; максимально количество узлов 65534;

Класс С – к нему относятся все адреса, старшие три биты которых имеют значение 1,1,0; в этом классе под идентификатор сети относятся 21 бит, а под идентификатор узла 8 бит; IP-адреса класса С лежат в диапазоне 192.0.0.0 – 223.255.255.255; максимальное количество узлов 254;

Класс D – к нему относятся все адреса, старшие четыре биты которых имеют значение 1110; в этом классе под идентификатор группы относят 28 бит; предназначены для группы передачи, поэтому делить адрес на идентификатор сети и узла не нужно; IP-адреса класса D лежат в диапазоне 224.0.0.0 – 239.255.255.255;

Класс E – к нему относятся все адреса, которые начинаются с последовательности 11110; IP-адреса класса E лежат в диапазоне 240.0.0.0 – 247.255.255.255; адреса этого класса были зарезервированы для будущего использования.

**Бесклассовая адресация**

Деление IP-адреса в бесклассовой адресации происходит на основе маски подсети.

Маска подсети – это число, применяемое в паре с IP- адресом и двоичная запись которого содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться, как идентификатор сети. Границы между последовательностями единиц и нулей в маске соответствуют границам идентификаторам сети и узла IP-адреса.

Число бит в IP-адресе, определяющих идентификатор сети, называется префиксом сети. Префикс сети указывают через косую черту после IP-адреса.

**Специальные IP-адреса**

Существуют ограничения для IP-адресов, а именно: идентификатор сети и/или идентификатор узла не могут состоять только из одних двоичных единиц или нулей. Однако, некоторые IP-адреса интерпретируются особым образом и на них это ограничения не распространяется.

Например, неопределенный адрес, который состоит только из двоичных нулей. Такой адрес в некоторых случаях используется в заголовках IP-пакетов. Например, ::/128 – неопределенный адрес. Подробно записаны в документах RFC 4193, RFC 4291, RFC 5735.

1. **Публичные и частные адреса**;

* публичные адреса, которые распространяются между узлами глобальных сетей и контролируются организацией IANA;
* частные адреса, которые могут назначаться узлам локальных сетей их администраторами и не контролируются организацией IANA; в IP4

для частных адресов выделено три диапазона: 10.0.0.0 – 10.255.255.255 [10.0.0.0/8], 172.16.0.0. – 172.31.255.255 [172.16.0.0/12], 192.168.0.0. – 182.168.255.255 [192.168.0.0/16].

В IP6 все частные адреса имеют префикс FC00::/7 и называются уникальные локальные адреса (ЮЛА), такие адреса имеют специальный формат: первые 7 бит – префикс, L – флаг (указывает на то, что значение так называемого глобального идентификатора является псевдо случайное число, которые было сгенерировано по стандартному алгоритму, описанным в документе RFC 4193), наличие глобального идентификатора позволяет свести к минимуму риск конфликтом IP-адресов при объединении сети;

1. **широковещательные адреса** (в IP4 есть два специальных адреса для широковещательных передач):

* ограниченный широковещательный адрес – это IP-адрес, состоящий только из двоичных единиц 255.255.255.255/32; отправляемый на такой адрес IP-пакет будет получен всеми узлами локальной сети (но за пределы сети отправляться не будет);
* широковещательный адрес – IP-адрес, в котором идентификатор узла состоит только из двоичных единиц 192.168.5.255/24; отправленный на такой адрес IP-пакет будет получен всеми узлами локальной сети, идентификатор которой указан в IP-адресе; в IP6 нет широковещательных адресов, но вместо них используются групповые адреса;

1. **групповые адреса** (IP-адрес, называемый групповым адресом представляет собой идентификатор группы узлов в сети; отправляемый на групповой IP-адрес пакет будет получен всеми узлами, входящими в соответствующую группу; в IP4 для групповых адресов выделен диапазон 224.0.0.0-239.255.255.255, например, 224.0.0.1 – этот адрес идентифицирует группу, в которую входит все узлы локальной сети, данный адрес можно использовать аналогично ограниченному широковещательному адресу; или второй пример, 224.0.0.2 – этот адрес идентифицирует группу, в которую входят все маршрутизаторы сети, с помощью данного адреса можно сделать рассылку по маршрутизаторам, присутствующим в локальной сети;   
   групповые адреса в IP6 имеют префикс FF00::/8, например, FF02::1 – идентифицирует группу, в которую входят все узлы локальной сети; FF02::2 – идентифицирует группу, в которую входят все маршрутизаторы сети; регистрацией и контролем групповых адресов является IANA);
2. **адрес обратной петли** (когда IP-пакет отправляется по такому адресу, то он не передается в сеть, а возвращается обратно на верхний уровень как будто бы был получен из сети, таким образом, маршрут перемещения данных образует петлю; адрес обратной петли используются для тестирования и для организации работы различных сетевых приложений, установленных на одном компьютере; в IP4 для этого подходит любой адрес из диапазона 127.0.0.1 – 127.255.255.254, но обычно используют 127.0.0.1/32; в IP6 используют только один адрес ::1/128);
3. **адреса IP4, отображенные в IP6** (в IP6 определен специальный тип адреса, который используется для отображения адресного пространство IP4 в адресное пространство IP6; такими адресами идентифицируются узлы, не поддерживающие IP6, если с ними необходимо организовать взаимодействие для устройств, поддерживающих только IP6; в таком адресе младшие 32 бита представляют адрес IP4, кроме того, этот адрес имеет префикс ::FFFF::/96 ( ::FFFF:C0A8:6D или 192.168.6.13).

**Формат IP-пакета**

Протокол IP определяет формат пакета, **который** содержит **заголовок и блок данных**. Для различных версий протокола IP имеются различные форматы заголовка IP-пакета.

1. Заголовок пакета IP4;

заголовок пакета IP4 содержит обязательно 20-байтную часть и необязательную переменную длины.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | Версия | | | | IHL | | | | Тип службы | | | | | | | | Длина пакета | | | | | | | | | | | | | | | |
| 32 | Идентификатор | | | | | | | | | | | | | | | | Флаги | | | Смещение фрагмента | | | | | | | | | | | | |
| 64 | TTL | | | | | | | | Протокол | | | | | | | | Контрольная сумма заголовка | | | | | | | | | | | | | | | |
| 96 | Адрес отправителя | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 128 | Адрес получателя | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 160 | Необязательная часть (переменной длины) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Тип службы – предназначен для различных классов обслуживания; теоретически это позволяет маршрутизаторам выбирать между различными каналами связи, отличающихся пропускной способностью и задержкой; на практике маршрутизаторы обычно игнорируют данное поле.

Длина пакета – содержит размер всего IP пакета, включая заголовок и данные; максимальный размер пакет не должен превышать 65 535 байт.

Идентификатор – позволяет узлу назначения определять какому сегменту данных верхнего уровня принадлежит полученный пакет; слишком длинные сегменты данных разбиваются на фрагменты, которые передаются в виде отдельных IP пакетов. Все фрагменты одного сегмента данных содержат одно и тоже значение идентификатора; после получения всех фрагментов, собранный сегмент данных передается на верхний уровень.

Флаги – первый бит н используется и равен нулю; второй флаг – DE, указывающий на то, что IP пакет не является фрагментом; третий флаг – MF, указывающий, что IP пакет не является последним в последовательности пакетов.

Смещение фрагмента – указывает положение фрагменты в исходном сегменте данных; смещение задается в блоках по 8 байт; первый фрагмент последовательности имеет 0-вое смещение.

Поле TTL – время жизни, длиной 1 байт – представляет собой счетчик, ограничивающий собой время жизни пакеты; значение этого поля уменьшается маршрутизаторами при передаче IP пакета; когда значение поля TTL становится равным нулю, пакет уничтожается, а отправитель рассылает специальный ICMP-пакет, сообщающий об ошибке. Таким образом, удается избежать вечного странствования IP пакета.

Протокол – длиной 1 байт – предназначено для указания протокола верхнего уровня, которому следует передать данные после того, как они будут собраны из полученных объектов.

Контрольная сумма – позволяет получателю IP пакета определить были ли искажены значения полей заголовка в процессе передачи IP пакета.

Необязательная часть – существует для того, чтобы с появлением новых вариантов протокола IP4 не пришлось вносить в заголовок поля, отсутствующих в нынешнем формате. Размер необязательной части должен быть кратен 4 байтам.

1. **Заголовок IP6 (имеет фиксированную длину 40 байт);**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | |  | | | | | | | |  | | | | | | | |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | Версия | | | | Приоритет | | | | Метка потока | | | | | | | | | | | | | | | | | | | | | | | |
| 32 | Длина полезной нагрузки | | | | | | | | | | | | | | | | Следующий заголовок | | | | | | | | Число переходов | | | | | | | |
| 64 | Адрес отправителя (16 байт) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 192 | Адрес получателя (16 байт) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Версия (4 бита) – содержит версию протокола, которому принадлежит пакет. Для IP6 значение поля равно 6. На период перехода с IP4 на IP6 маршрутизаторы по значению данного поля могут различать IP-пакеты разных версий.

Приоритет – используется при определении порядка передачи IP-пакета; все передаваемые IP пакеты относятся к одной из двух категорий: управляемые и неуправляемые IP пакеты; неуправляемые IP пакеты всегда передаются перед управляемые; управляемые IP пакеты реагируют на проблемы, связанные с перегруженностью сети; если сеть перегружена, пересылка IP пакета приостанавливается до устранения проблемы; категории управляемых IP пакетов существует несколько классов, уточняющих их приоритеты:

* 0 – приоритет не задан;
* 1 – фоновая пересылка данных;
* 3- не используется;
* 4 – обслуживаемая массовая пересылка данных;
* 6 – интерактивная передача данных;
* 7 – пересылка управляющих данных.

Метка потока – используется для идентификации потока данных в сети, если всем IP пакетам в контексте одной передачи данных будет задано одно значение потоковой метки, то маршрутизаторы смогут запомнить по какому маршруту должны передаваться все IP пакеты с этой меткой. В дальнейшем при получении IP пакета с тем же значением меткой потока маршрутизатора не приходится вычислять маршрут заново, тем самым ускоряется прохождение IP пакетов через маршрутизатор. Чтобы предотвратить чрезмерное разрастания устаревших маршрутов, предполагается, что маршруты не должны хранится более 6 секунд. Если за это время не будет получен IP пакет с определенной меткой потока, маршрут будет удален.

Длинна полезной нагрузки (2 байта) – содержит количество байт, которое следует зафиксированным заголовком IP пакета.

Следующий заголовок (1 байт) – определяет какой из дополнительных заголовков следует за фиксированным заголовком IP пакета; значение этого поля идентифицирует заголовок расширения; к фиксированному заголовку может быть присоединено несколько заголовков расширения, каждое из которых имеет поле следующий заголовок, в котором указывается тип следующего заголовка расширения. Последний заголовок расширения должен иметь в поле значение 59, которое указывается на отсутствие следующего заголовка, либо значение, указывающее протокол верхнего уровня, которому следует передать данные после того, как они будут собраны из полученных фрагментов

Число переходов – поле длиной 1 байт, определяет максимальное число маршрутизаторов, которое может пройти IP пакет; при прохождении маршрутизатора значение этого поля уменьшается на 1 и по достижению 0 IP-пакет уничтожается.

Адрес отправителя (16 байт)

Адрес получателя (16 байт).

**Система доменных имен (Domain name system)**

Первоначально, когда сети были относительно небольшими, для установления соответствия между доменным именем и сетевым адресом, использовался файл hosts, в котором содержался список всех доменных имен и соответствующих им адресам. Этот файл хранится на каждом компьютере в сети.

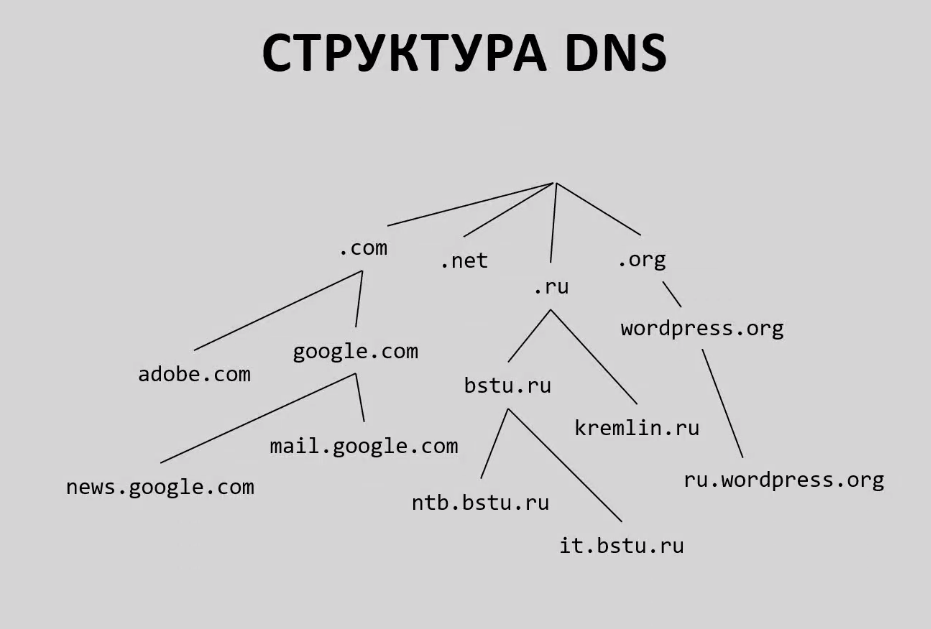
Однако, по мере добавления в сеть новых компьютеров, обновление этого файла становилось всё более трудной задачей. Ситуация кардинально изменилась, после создания системы доменных имен.

DNS – это распределенная система, предназначенная для получения информации о сетевом узле по его доменному имени.

На сегодняшний день файл hosts продолжает использоваться, но обычно только для того, чтобы предотвратить поиск локальных сетевых узлов в DNS, так как запрос к этому файлу имеет приоритет перед обращением к DNS.

**Структура DNS:**

DNS имеет иерархическую структуру. Каждый уровень в этой иерархии называется доменом. При записи домены отделяются друг от друга точкой.



Самый верхний уровень этой иерархии называется корнем.

Следующий уровень в иерархии – домены верхнего уровня (com, net, org), которые описывают категорию, относящуюся к тому или иному пространству доменных имен. Домены верхнего уровня общедоступны и могут использоваться любыми людьми и организациями.

Также существуют домены верхнего уровня, в которых отражается код страны.

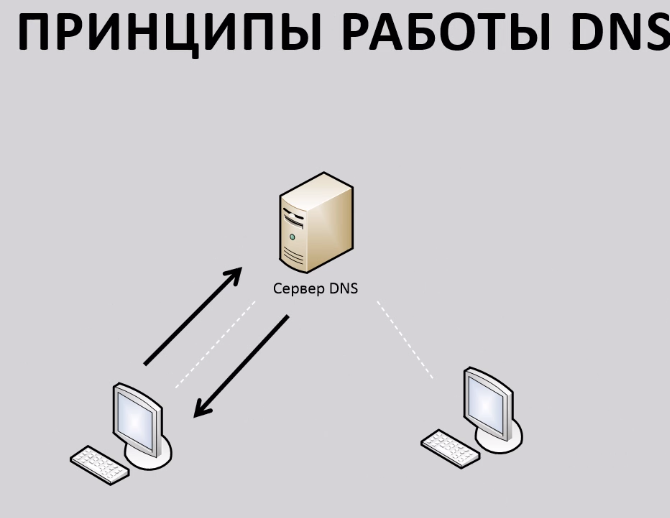
За управление доменами верхнего уровня отвечает организация IANA.

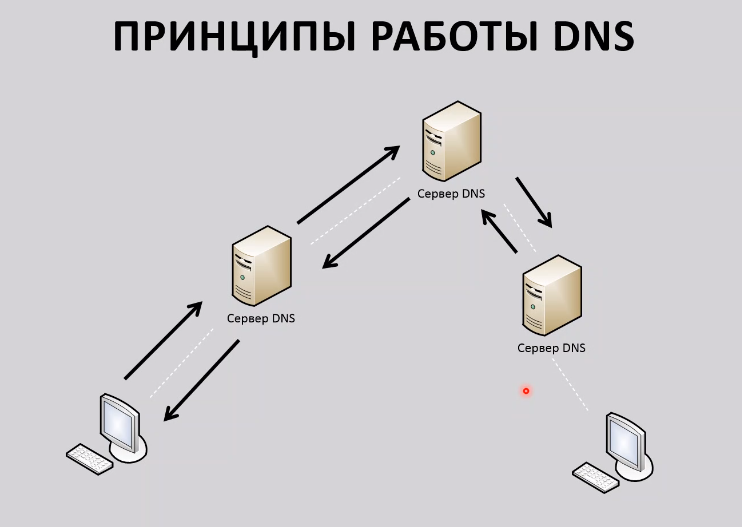
После доменов верхнего уровня следуют домены второго уровня (google.com).

В иерархии DNS также могут создаваться различные поддомены.

**Принципы работы DNS:**

Структура DNS позволяет серверам делегировать ответственность за часть доменов, подчиненным, по отношению к ним, серверам. Технически делегирование выражается в выделении множества доменов в отдельную зону. Зоной в DNS называется множество доменов, за управление которыми отвечает один или группа DNS-серверов. Зона может содержать один или несколько доменов. В случае их делегирования, эти домены образуют отдельные зоны, которые будут иметь собственные DNS-серверы.





Если необходимо получить IP-адрес какого-либо домена, отправляется запрос к DNS-серверу на поиск этого IP-адреса. В DNS реализуется три сценария поиска:

1. Если домен в той же зоне, что и компьютер, который посылал запрос. В таком случае, локальный DNS-сервер, имеющий в своей базе данных адрес домены, возвращает запрашиваемый IP-адрес.
2. Если необходимо получить IP-адрес домена в другой зоне. В этом случае локальный DNS-сервер обнаруживает, что нужный домен находится в другой зоне, и, в зависимости от того, где находится эта зона, формирует запрос родительскому или дочернему серверу. Если этот DNS-сервер обнаруживает, что нужный домен также находится в другой зоне, тогда формируется новый запрос, который отправляется дальше. Если же этот DNS-сервер отвечает за зону, в которой находится домен, искомый IP-адрес возвращается DNS-серверу, который посылал запрос. Когда IP-адрес достигнет локального DNS-сервера, он отправит IP-адрес компьютеру, с которого был подан запрос.
3. Если необходимо повторно получить IP-адрес домена в другой зоне. В таком случае локальный DNS-сервер проверяет наличие искомого IP-адреса в своем кэше. Если IP-адрес все еще в кэше, он посылается запрашиваемому компьютеру.   
   По мере обработки запросов, DNS-сервер сохраняет все ответы на них (по умолчанию 24 часа).

Кроме прямого запроса, который пытается найти IP-адрес по известному доменному имени, существует обратный запрос, который решает противоположную задачу, т.е. поиск доменного имени по известному IP-адресу.

**База данных DNS**

представляет собой текстовый файл, состоящий из записей ресурсов, которые описывают домены и их функции в зоне. DNS-сервер выдает записи ресурсов по запросу компьютера или других серверов.

Запись ресурса, как правило, состоит из 5 полей:

1. Доменное имя – определяет домен, к которому относится запись ресурса.  
   Есть две формы записи доменных имен в базе – полное имя домена с точкой на конце (it.bstu.ru) или только имя поддомена (it).
2. За полем доменного имени может следовать необязательно поле «время жизни», которое указывает, в течении которого промежутка времени информация в этой записи должна считаться действительной.
3. Класс адреса – необязательное и всегда принимает значение IN. Данное поле используется только для совместимости со старыми версиями DNS-серверов и DNS -клиентов.
4. Тип записи ресурсов – указывает тип записи ресурсов. Существует более 20 типов.
5. Данные – поле, которое содержит значения специфические для указанного типа записи ресурса.

**Типы записи ресурса:**

**Запись начала полномочий (SOA)** – в базе данных DNS показывает, какой сервер является ответственным за конкретною зону, кроме того содержит адрес электронной почты, отвечающего за работу DNS, и другая критически важная информация.

Name – имя зоны, которые далее описывается в базе данных

origin – имя сервера, где ведется эта зона

person – адрес электронной почты администратора данной зоны

serial – текущая версия БД DNS

refresh – временной интервал (сек), через который сервер должен обновлять содержимое своей БД

retry – временной интервал, через который сервер обновит данные после неудачной попытки

- временной интервал в секундах, в течении которого содержимое БД считается достоверным

minimum – время жизни, которое возвращается для всех запросов к БД

**Запись адреса** – является наиболее распространенной, и связывает имя домена с IP-адресом.

A – для IPv4, AAAA – для IPv6.

host [время жизни] [класс адреса] A address

host – имя домена, address – IP-адрес этого домена

**Запись канонических имен** – запись канонического имени домена позволяет создавать псевдоним домена и позволяет любому домену упоминаться в БД под несколькими именами. По сути, эта запись перенаправляет адресуемые ей запросы к записи адреса, соответствующего ей домена.

name [время жизни] [класс адреса] CNAME host

**Записи указателя** – используются для определения имени домена по его IP-адресу

arpa [время жизни] [класс адреса] PTR host

arpa – доменное имя из пространства имен домена, которое формируется с используется

host – имя домена

in-addr.arpa – IPv4

ip6.arpa - IPv6

Обратный порядок в доменном имени необходим, так как, в отличии от доменных имен, при чтении адресов слева-направо, они воспринимаются в обратном порядке.

**Запись обмена почтой** – указывает, какие домены позволяют принимать сообщения электронной почты. Записи MX настраиваются на уровне зоны, то есть вся электронная почта, отправляемая домену с именем зоны будет перенаправляться тому домену, который указан в соответствующей записи МХ.

name MX preference host

name – имя зоны. Если оно опущено, запись МХ относится к зоне, указанной в записи начала полномочий

preference - целое число, обозначающее порядок подключения, если используется несколько почтовых серверов

host - имя домена, на котором установлен почтовый сервер

**Запись сервера имен (NS)** – указывает, какой домен в БД DNS является сервером для конкретной зоны.

host [время жизни] [класс адреса] NS server

host - имя домена в зоне, за которую отвечает сервер. Если имя домена опущено, запись NS относится к зоне, указанной в записи начала полномочий.

**Протокол DNS**

является протоколом прикладного уровня стека протоколов TCP IP и использует клиент-серверную архитектуру, где клиент посылает запрос, а сервер отвечает либо информацией, полученной из базы данных, либо сообщением об ошибки, в котором указывается причина отказа в обработке запроса. Передача этих данных производится при помощи протоколов транспортного уровня TCP и UDP. При этом используется порт с номером 53. Запрос и ответ DNS имеют одинаковую структуру и состоят из заголовка и четырех блоков данных:

1. Заголовок
2. Блок запросов
3. Блок ответов
4. Блок авторитетных серверов
5. Блок дополнительной информации

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | | 3 | | | | | | | |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | Идентификатор | | | | | | | | | | | | | | | | QR | | OPCODE | | | | АА | ТС | RD | RA | зарезервированы | | | RCODE | | | |
| 32 | Кол-во записей в блоке запросов | | | | | | | | | | | | | | | | | Кол-во записей в блоке ответов | | | | | | | | | | | | | | | |
| 64 | Кол-во записей в блоке авторитетных запросов | | | | | | | | | | | | | | | | | Кол-во записей в блоке доп. информации | | | | | | | | | | | | | | | |

**Блоки данных в сообщении DNS**

В блоке запросов указывается какие записи ресурсов хочет получить DNS-клиент, в блоке ответов содержатся записи ресурсов, доступные в базе данных DNS на момент запроса DNS-клиента. Если в запросе не установлен флаг AA, то в блоке авторитетных серверов будут указаны DNS-серверы, к которым DNS-клиент может обращаться за авторитетными ответами. В блоке дополнительной информации содержатся записи ресурсов, которые могут иметь отношение к запрашиваемым ресурсам.

Каждая запись **блока запроса** состоит из трех полей:

1. первое поле – доменное имя (имеет переменную длину и специальный формат записи; перед именем каждого домена ставится однобайтное значение, которое определяет длину имени этого домена, конец списка обозначается нулем);
2. второе поле – тип записи ресурсов (имеет длину 2 байта и указывает тип запрашиваемого ресурса; например, SOA = 1, A=1, AAAA=28, CNAME=5, PTR=12, MX=15, NS=2);
3. третье поле – класс адреса (имеет длину 2 байта и всегда принимает значение равное 1, что соответствует классу IN=1).

**Последние три блока** данных в DNS-сообщении имеют одинаковый формат. Каждая запись в этих блоках имеет 6 полей:

1. первое поле – доменное имя, переменной длины, записывается в том же формате, что и в запросе;
2. второе поле – тип записи ресурсов, длинной 2 байта, содержит те же значения, что и в запросе;
3. третье поле – класс адреса, всегда принимает значение равное 1;
4. четвертое поле – время жизни, имеет длину 4 байта, берется из базы данных;
5. пятое поле – размер данных, длиной 2 байта;
6. шестое поле – поле данных, переменной длины, содержат данные соответствующие записи ресурсов.

**Протокол DHCP**

Сетевые адрес могут назначаться статически и динамически.

При использовании первого способа вручную назначает адрес каждому устройству, вместо этого можно назначать адреса динамически, например, посредством протокола DHCP.

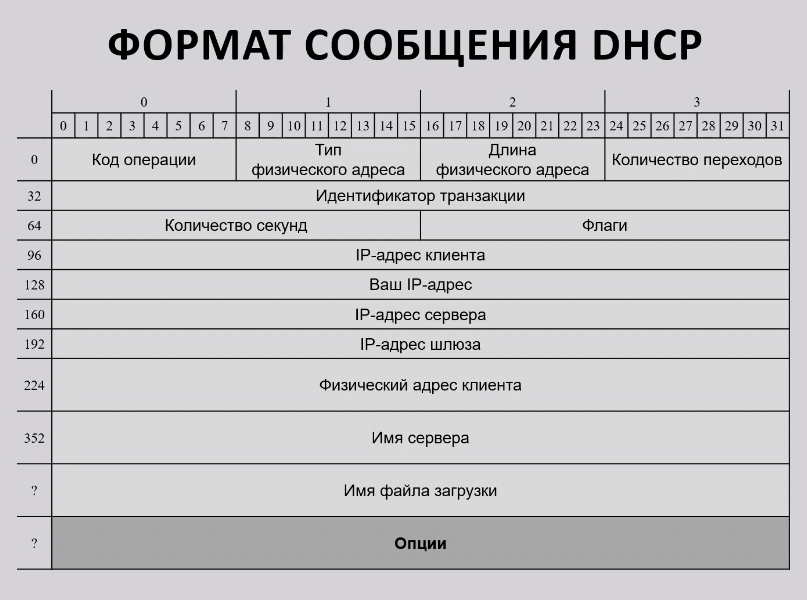
**Протокол DHCP – протокол прикладного уровня стека протоколов DHCP-IP, позволяющим динамически назначать для сетевых узлов IP-адрес и другие параметры конфигурации (маску подсети, DNS-сервера, IP-адреса основного шлюза).**

Протокол DHCP использует клиент-серверную архитектуру, где клиент посылает запрос, а сервер возвращает IP-адрес и другие параметры кофигурации.

DHCPv6 – предназначен для IP-адресов 6 версии.

Протокол DHCP описан в документе RFC 2131, а шестая версия в RFC 3315.

Формат сообщения DHCP



Передача сообщений DHCP осуществляется посредством транспортного протокола UDP. При этом сервер принимает значения от порта 67, а отправляет от порта 68.

Первое поле – код операции – 1 байт – указывает является ли сообщение запросом или ответом (1=запрос, 2=ответ);

Второе поле – тип физического адреса – 1 байт – указывает тип физического адреса клиента (MAC-адрес – 1);

Третье поле – длина физического адреса – 1 байт – указывает размер физического адреса (MAC-адрес = 6 байт);

Четвертое поле – количество переходов – 1 байт – содержит количество промежуточных маршрутизаторов, через которое прошло сообщение.

Поле идентификатор транзакции – 4 байта – позволяет соотнести ответы и вопросы в рамках одной DHCP-транзакции., значение этого поля выдается DHCP-клиентов в начале процесса получения IP-адреса.

Поле количество секунд – 2 байта – содержит время в секундах с момента начала получения IP-адреса, это поле может не использоваться, тогда оно должно принимать значение равное 0.

Поле флаги – старший бит данного поля определен, как флаг B, а остальные биты зарезервированы для будущего применения и должны быть равны 0. Флаг B устанавливает в 1, если DHCP-клиент требует широковещательного ответа.

Поле IP-адрес клиента – 4 байта – выполняется только в том случае, если узел уже имеет IP-адрес. Это возможно, если DHCP-клиент выполняет процедуру обновления IP-адреса по истечению срока аренды.

Поле Ваш IP-адрес – 4 байта – содержит адрес, предлагаемый DHCP-сервером.

Поле IP-адрес сервера – заполняется DHCP-сервером при ответе на запрос DHCP-клиента.

Поле IP-адрес шлюза – 4 байта – содержит адрес агента DHCP-ретрансляции, который ретранслирует сообщение DHCP между DHCP-клиентом и DHCP-сервером, если они находятся в разных сетях.

Поле физический адрес клиента – 16 байт – содержит физический адрес DHCP-клиента (обычно MAC-адрес и тогда использует 6 байт).

Поле имя сервера – имеет переменную длину и является необязательным (до 64 байт) – содержит имя сервера в виде ASCIZ-строки.

Поле имя файла загрузки – имеет переменную длину до 128 байт и является необязательным – содержит имя файла на сервере, используемое бездисковыми рабочими станциями.

Поле опции – имеет переменную длину – содержит различные дополнительные параметры (эти параметры описаны в документе RFC 2132). В начале этого поля содержатся четыре магических числа 99,130,83,99, которые позволяют DHCP-клиенту и DHCP-серверу определять наличие поля опции.

Аренда IP-адреса

Выдача IP-адреса в аренду производится DHCP-сервером по запросу DHCP-клиента. DHCP-сервер гарантирует, что выданный им IP-адрес до срока аренды не будет выдан другому узлу. При повторных запросах DHCP-сервер старается предложить IP-адрес, которым узел пользовался ранее. DHCP-клиент может запросить продление срока аренды IP-адреса, либо наоборот отказаться от него. При острой нехватки адресов DHCP-сервер может сократить срок аренды IP-адреса.

1 этап. Старт инициализации: на этом этапе DHCP-клиент отправляет сообщение DHCPPOISCOVER на ограниченный широковещательный адрес обнаружить доступные DHCP-серверы, при этом, если DHCP-клиент не имеет IP-адресов, то в поле IP-адрес клиента указывается неопределенный IP-адрес и такой же адрес указывается в IP-пакете, как адрес отправителя. Кроме того, чтобы DHCP-серверы как-то могли идентифицировать клиента, в поле физический адрес клиента помещается его MAC-адрес.

2 этап. Определение конфигурации: получив сообщение от DHCP-клиента, DHCP-сервер определяет требуемую конфигурацию клиента в соответствии с указанными администратором сети настройками. После чего DHCP-клиенту на его MAC-адрес отправляется сообщение типа DHCPOFFER, в котором предлагается IP-адрес и параметры конфигурации. Предлагаемый IP-адрес указывается в поле Ваш IP-адрес, а прочие параметры конфигурации указываются в виде Опций в соответствующем поле. Кроме того, в поле IP-адрес сервера указывается адрес DHCP-сервера.

3 этап. Выбор конфигурации: на этом этапе DHCP-клиент может получить несколько различных предложений от DHCP-серверов. Выбрав один из предложенных вариантов, DHCP-клиент отправляет на ограниченный широковещательный IP-адрес сообщение DHCPREQUEST, в котором в поле Опций указывается IP-адрес, выбранного DHCP-сервера.

4 этап. Выполнение конфигурации и завершение инициализации: выбранный DHCP-сервер завершает процесс выдачи IP-адреса, отправляя DHCP-клиенту на его MAC-адрес сообщение DHCPACK. Причем в поле Опций указывается предложенный IP-адрес и прочие параметры конфигурации. Получим такое сообщение, DHCP-клиент должен настроить сетевой интерфейс узла, используя предложенный IP-адрес и параметры конфигурации.

**Протокол ARP**

Для доставки кадров внутри сети нужно определить соответствие между физическим адресом узла и его сетевым адресом. С этой целью был разработан протокол ARP.

Протокол ARP – протокол сетевого уровня стека протоколов TCP/IP, предназначенный для определения физического адреса узла по его сетевому адресу.

Необходимость протокола ARP продиктована тем обстоятельством, что сетевые адреса назначаются узлам, независимо от их физических адресов.

Для IP6 протокол ARP не применяется, его функции возможны на протокол ICMPv6.

Формат ARP-пакета



Поле Тип физического адреса – 2 байта – указывает тип физического адреса; поле Тип сетевого адреса – 2 байта – указывает тип сетевого адреса (для MAC=1, IP=2048);

Поле Длина физического адреса (1 байт) и поле Длина сетевого адреса (1 байт) для MAC адреса 6 байт, для IP адреса 4 байта.

Поле Код операции – 2 байта – указывает содержит ли ARP-пакет запрос или ответ (запрос=1, ответ=2).

Поле Физический адрес отправителя, поле Сетевой адрес отправителя, поле Физический адрес получателя, поле Сетевой адрес получателя – имеет переменную длину.

**Определение MAC-адреса для заданного IP-адреса**

Когда требуется определить MAC-адрес для заданного IP-адреса 4 версии, ищется соответствующая запись в **специальной ARP-таблице, которую имеет каждый узел сети**, который поддерживает сетевой уровень. В ARP-таблице содержатся IP-адреса и соответствующие им MAC-адреса всех узлов сети.

Когда ARP-таблице не удается найти заданный IP адрес:

1. Узел, которому нужно определить MAC-адрес по IP-адресу **формирует ARP-пакет**, указывая в нем **искомый** IP-адрес и свой MAC-адрес, и отправляет его на **широковещательный** MAC-адрес (6 пар FF);
2. Все узлы сети, получая ARP-пакет, **сравнивают указанный в нем IP-адрес** с собственным IP-адресом;
3. Узел, обладающий искомым IP-адресом, формирует ARP-пакет, в котором указывает свой MAC-адрес и IP-адрес, и отправляет его по MAC-адресу, который был получен в ARP-пакете, при этом **узел добавляет в свою ARP-таблицу новую запись** с MAC- и IP-адресом отправителя в свою ARP-таблицу;
4. Получив ответный ARP-пакет, узел сможет добавить в свою **ARP-таблицу новую запись с искомым IP-адресом и полученным MAC-адресом.**

В случае, если в сети нет узла с искомым IP-адресом, не будет и соответствующей записи в ARP-таблице, поэтому ARP-пакеты, отправляемые по этому адресу, будут уничтожаться.

**Протокол ICMP/ICMPv6**

Протокол ICMP – **протокол сетевого уровня стека протоколов TCP/IP, применяемый для передачи сообщений об ошибках, возникших при передачи данных.** Описан в документах RFC 792, RFC 950.

ICMP пакет инкапсулируется в IP-пакет, но не смотря на это, он является неотъемлемой частью протокола IP4. Для протокола IP6 предусмотрен аналогичный протокол ICMP6, который описан в документе RFC4443.

Формат ICMP-пакета

Протокол ICMP и ICMP6 определяют формат пакета, который состоит из заголовка и блока данных. Заголовок имеет фиксированную длину 8 байт.

Поле Тип (1 байт) – указывает тип сообщения, который определяет формат блока данных ICMP-пакета;

Поле Код (1 байт) – обеспечивает дополнительный уровень детализации сообщения; значение данного поля зависит от типа сообщения;

Поле Контрольная сумма (2 байта) – содержит значения, которые представляют собой результат вычисления контрольной суммы всего ICMP-пакета.

Ещё 4 байта после зарезервированы для других форматов сообщения.

Типы сообщений ICMP:

* 0 – эхо-ответ;
* 3 – узел назначения недостижим;
* 5 – перенаправление маршрута;
* 8 – эхо-запрос;
* 11 – истечение времени;
* 13 – запрос отметки времени;
* 14 – ответ отметки времени;
* 15 – информационный запрос;
* 16 – информационный ответ;
* 17 – запрос маски;
* 18 – ответ маски.

**Протоколы ICMP и ICMP6 предоставляют возможность тестирования достижимости узлов сети посредством двух типов сообщений: эхо-запрос и эхо-ответ.**

Формат ICMP-пакета. Эхо-сообщения



Поле Тип – указывает является ли пакет эхо-запросом или эхо-ответом;

Поле Код – содержит нулевое значение;

Поле Идентификатор (2 байта) – содержит значение, используемое для идентификации сеансов;

Поле Порядковый номер (2 байта) – указывает номер сообщения в последовательности передаваемых эхо-сообщений.

Поле Данные – имеет переменную длину и содержит информацию, которая должна быть возвращена в эхо-ответе.

Тестирование узла сети на его достижимость заключается в следующем: узлу, достижимость которого нужно проверить, посылается эхо-запроса. Если узел получил эхо-запрос, он формирует и отправляет эхо-ответ, так как ICMP-пакет эхо-запрос и эхо-ответ **передает внутри IP-пакета**, их успешная доставка будет обозначать доступность узла в сети.

**Недостижимость узла назначения**

Когда маршрутизатор **не может доставить IP-пакет, он отсылает узлу, отправившему этот IP-пакет, специальный ICMP-пакет, сообщающий о том, что узел назначения недостижим**.



Поле Тип – принимает значение 3, которое определяет тип сообщения, как сообщение по недостижимости узла назначения;

Поле Код – содержит значение, уточняющее причину, по которому IP-пакет не был доставлен, кроме причины ошибки ICMP-пакет включает заголовок недоставленного IP-пакета + его первые 64 бита блока данных.

Недостижимость узла назначения:

* 0 – сеть недоступна;
* 1 – узел назначения недоступен;
* 2 – протокол недоступен;
* 3 – порт недоступен;
* 4 – необходима фрагментация, но установлен флаг DF;
* 5 – ошибка в маршруте, заданном источником.

В протоколе ICMP6 сообщение по недостижимости узла назначения имеет формат схожий с форматом аналогичного сообщения в формате ICMP. За исключением того, что в него включается часть IP-пакета такого размера, чтобы ICMP пакет имел длину не более минимального размера блока данных протокола IP6. Также в протоколе ICMP6 для такого сообщения используются другие значения полей Тип и Код.

**Определение MAC-адреса для заданного адреса IP6**

Для IP6 при необходимости определить MAC-адрес по его IP-адресу используется протокол NDP, который дополняет протокол ICMP6. Описан в документе RFC 4861.

Протокол NDP определяет 5 новых типов сообщений ICMP6, два из которых используются для определения MAC-адреса для заданного IP-адреса. Это Запрос соседа и Объявление соседа.

Чтобы получит MAC-адрес целевого узла, необходимо отправить ICMP-пакет с Запросом соседа (это тип сообщения) на групповой адрес FF02::1 (получает все узлы сети). Получив такой ICMP-пакет, целевой узел в ответ на него посылает ICMP-пакет, содержащий сообщение Объявления соседа, в котором указывает свой MAC-адрес.

**Формат сообщения Запрос соседа**

Поле Тип – принимает значение 135, которое определяет тип сообщения, как Запрос соседа;

Поле Код – принимает нулевое значение;

Поле Целевой адрес (16 байт) – содержит IP-адрес целевого узла, MAC-адрес которого необходимо определить;

Поле Опции – переменной длины, может содержать MAC-адрес отправителя ICMP-пакета; MAC-адрес отправителя не должен включаться, когда IP-адрес является неопределенным.

**Формат сообщения Объявление соседа**

Сообщение Объявление соседа также используется при обнаружении недоступности соседа.

Поле Тип – принимает значение 136, которое определяет тип сообщения, как Объявление соседа;

Поле Код – содержит нулевое значение;

Класс R – указывает на то, что отправителем пакета является маршрутизатор;

Класс S – указывает на то, что данное сообщение было отправлено в ответ на сообщение Запрос соседа;

Класс О – указывает необходимо ли перезаписывать в кэше существующую запись о соседнем узле, когда класс О не установлен, запись не нужно обновлять, хотя запись может быть добавлена, если в кэше ещё не было указанного MAC-адреса;

Поле Целевой адрес (16 байт) – содержит IP-адрес целевого узла из ICMP-пакета, который вызвал объявление соседа;

Поле Опции – переменной длины, содержат MAC-адрес целевого узла, то есть отправителя объявления соседа.

**Объединение сетей**

Потребность в объединении сети связана с ограничениями протяженности сети и числа узлов в ней, а также в необходимости снизить нагрузку в сети.

Совокупность нескольких сетей, соединенных между собой, называется объединенной сетью.

При создании объединенных сетей требуются специальные устройства, которые позволяют подключать к себе две или более сети/подсети.

К таким устройствам относятся **мосты и маршрутизатора.**

**Объединение сетей с помощью мостов**

**Мост** – обеспечивает объединение сетей на канальном уровне, обеспечивая передачу данных из одной подсети в другую. Мосты способны объединять сети разных сетевых технологий.

В сетях Ethernet чаще всего применяется прозрачное мостовое соединение;

В сетях TokenRing применяется мостовое соединения с маршрутизацией от источника;

В объединенных сетях используется трансляционное мостовое соединение.

**Прозрачное мостовое соединение**

Каждый мост в объединенной сети анализирует проходящие через него кадры и изучает состав сети.

Для каждого из них сеть представляется таблицей MAC-адресов, связанных с определенным портом этого моста.

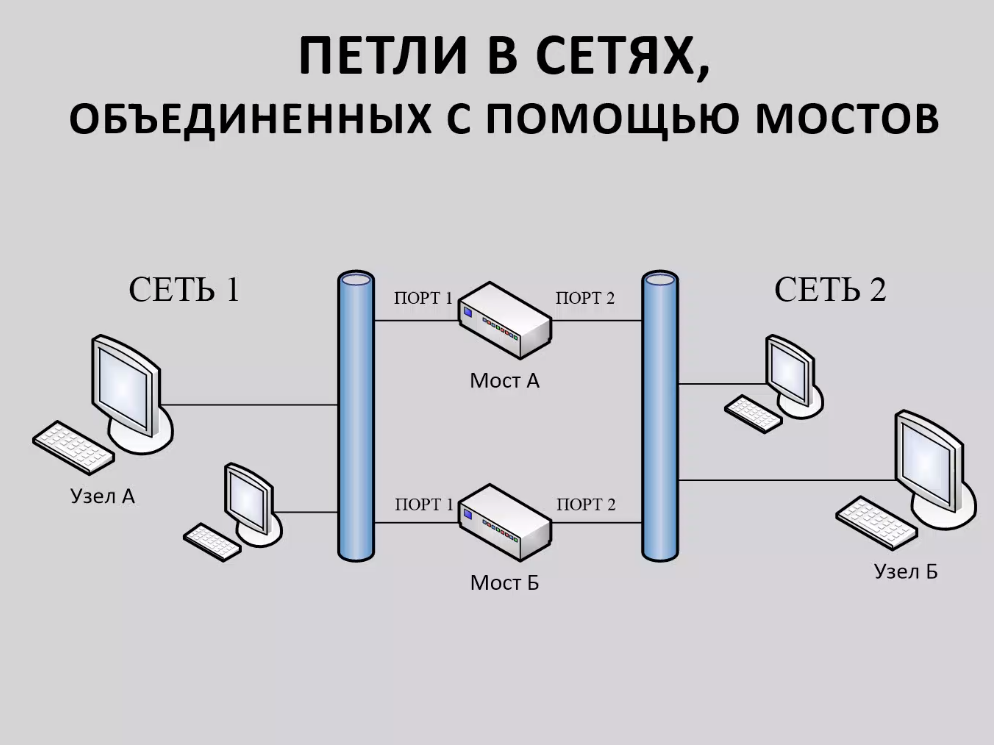
Изначально мост ничего не знает о том, какой подсети принадлежат узлы. Однако, постепенно изучая передаваемые кадры мост заполняет свою таблицу MAC-адресов, записывая в неё адрес отправителя кадра и порт, от которого этот кадр был получен.

Каждый раз при получении кадра, мост ищет адрес назначения этого кадра в своей таблице MAC-адресов. Если таблица MAC-адресов содержит адрес, то этот кадр передается в другую сеть через указанный порт.

Если не найдено никакой связи, то кадр рассылается через все активные порты, кроме порта, из которого кар был получен.

**Петли в сетях, объединенных с помощью мостов**

В сетях, объединенных с помощью мостов, **петлей называется состояние сети, при котором происходит бесконечная пересылка кадров между мостами, подключенными в общую сеть.**



Петля может образовываться следующим образом:

1. узел А отправляет кадр узлу Б;
2. мост А получает данный кадр и заносит в свою страницу MAC-адресов адрес узла А и связывает его с портом 1;
3. если мост А, пока не знает местоположение получателя кадров узла назначения, он рассылает кадр через все свои активные порты, кроме порта от которого этот кадр был получен;
4. мост Б также получает кадр от узла А и выполняет те же действия, что и мост А;
5. узел Б получит две копии кадров: одну от моста А, другую от моста Б;
6. одновременно с этим через сеть 2 мост Б получает копию кадра от моста А и производит следующие действия:
   1. изменяет соответствующую запись в таблице MAC-адресов, чтобы связать адрес узла А с портом 2;
   2. рассылает кадр через все активные порты, кроме порта, через который кадр был получен;
7. аналогичная ситуация происходит с мостом А, когда он получит копию кадра от моста Б.

В результате, при ответе узла Б на кадр узла А оба моста проигнорируют эти ответы, так как будут считать, что их получатель (узел А) находится в той же сети, что и их отправитель (узел Б).

Если в объединенной сети, которая имеет петли, какой-нибудь узел отправит кадр широковещательным адресом назначения, то это приведет к ситуации **широковещательного шторма, когда всем узлам сети будут бесконечно рассылаться копии кадров.**

Однако, наличие избыточных связей, которые и образуют петлю, может быть полезным для повышения надежности сети, за счет образования резервных путей.

**Протокол связующего дерева STP**

Протокол STP – протокол канального уровня, предназначенный для устранения петель в сетях Ethernet, объединенных с помощью мостов.

Описан стандартом IEEE 802.1D.

Первым **делом STP определяет роли мостов и портов, а затем блокирует часть портов так, чтобы в итоге получилась топология дерева, которая не будет иметь петель**.

Для того, чтобы определить какие порты блокировать, мосты периодически обмениваются сообщениями BPDU (протокольный блок данных моста). Для передачи BPDU-сообщений используется групповой MAC-адрес 01-80-С2-00-00-00.

**1 этап. Выбор корневого моста.**

Для того, чтобы получить топологию дерева в сети выбирается **один корневой мост, от которого будет строиться данная топология**.

Для **выбора корневого моста каждому мосту назначается уникальный идентификатор**. Идентификатор моста представляет собой 8-байтное число, в котором 6 младших байт содержит MAC-адрес моста, а 2 старших байта приоритет данного моста. Корневым мостом выбирается мост с наименьшим идентификатором. После **включения, каждый мост сначала считает себя корневым, но только до тех пор, пока не выяснится обратное**. Поэтому каждый мост рассылает через все **свои активные порты BPDU-сообщение,** в котором указывает свой идентификатор в качестве идентификатора корневого моста. Принимая BPDU-сообщение, мост сравнивает значение идентификатора корневого моста, полученного из сообщения, со значением корневого моста, хранящегося в памяти. Если полученное значение меньше, то оно сохраняется в качестве идентификатора корневого моста, а затем мост снова рассылает BPDU-сообщение, но уже с новым идентификатором корневого моста. Потом все мосты определяться: кто из них является корневым мостом.

Можно влиять на процесс выбора корневого моста, изменяя приоритет. Так как приоритет находится в старших разрядах идентификатора моста, то его значение подавляется значение MAC-адреса. Если же назначить всем мостам одинаковый приоритет, то будет выбран мост с наименьшим значением MAC-адреса.

**2 этап. Определение корневых портов.**

Для каждого моста, кроме корневого моста, **определяется корневой порт, то есть порт через который мост будет обмениваться кадрами с корневым мостом.**

Корневым портом выбирается порт, который имеет наименьшую суммарную стоимость пути в корневом мосту.

|  |  |  |
| --- | --- | --- |
| Сетевая технология | Скорость | Стоимость |
| Ethernet | До 10 Мбит/с | 100 |
| Fast Ethernet | До 100 Мбит/с | 19 |
| Gigabit Ethernet | До 1 Гбит/с | 4 |
| 10Gigabit Ethernet | До 10 Гбит/с | 2 |

Когда мост получает BPDU-сообщение, он увеличивает стоимость пути для порта, через который было получено данное сообщение, на соответствующую ему величину затрат.

Если найдется несколько портов с одинаковой стоимостью, **то корневым портом будет порт с наименьшим идентификатором.** Идентификатор порта представляет собой двухбайтовое число, в котором старший байт представляет собой

**Выбор назначенных мостов**

Для каждой подсети выбирается назначенный мост, который будет передавать кадры между узлами данной сети.

Мост может быть назначен сразу нескольким подсетями, но для подсети может быть назначен только один мост.

Среди подходящих мостов выбирается мост, имеющий корневой порт с наименьшей стоимостью пути к корневому мосту. Если таких мостов несколько, то назначен будет мост с наименьшим идентификатором.

**4 этап. Определение назначенных портов.**

Для всех мостов определяется назначенный порт, то есть порт, через который мост обменивается кадрами с узлами подключенной к нему подсети. У корневого моста, как правило, все порты являются назначенными. Если мост подключен к подсети с помощью нескольких портов, то назначенным портом будет выбран порт с наименьшим идентификатором. Все порты, которые не являются корневыми или назначенными, блокируются.

**Формат BPDU-сообщения**

****

Поле Идентификатор протокола (2 байта) – содержит нулевое значение;

Поле Версия (1 байт) – содержит нулевое значение;

Поле Тип сообщения (1 байт) – содержит нулевое значение;

Поле Флаги (1 байт) – используется только старших два бита TCN (этот класс указывает на то, что сообщение является уведомлением о изменении топологии, как правило, мост рассылает такие уведомления, если обнаружит отказ моста или изменения/отказ линии порта или изменения состояния порта) и TCA (класс указывает на то, что сообщение является подтверждением изменения топологии, которое отправляется в ответ на уведомление об изменении топологии;

Поле Идентификатор корневого моста (8 байт) – содержит идентификатор корневого моста (старшие 2 байта – приоритет, младшие 6 байт – MAC-адрес);

Поле Стоимость пути до корня (4 байта) – содержит суммарную стоимость пути до корневого моста;

Поле Идентификатор моста (8 байт) – содержит идентификатор моста, пославшего сообщение;

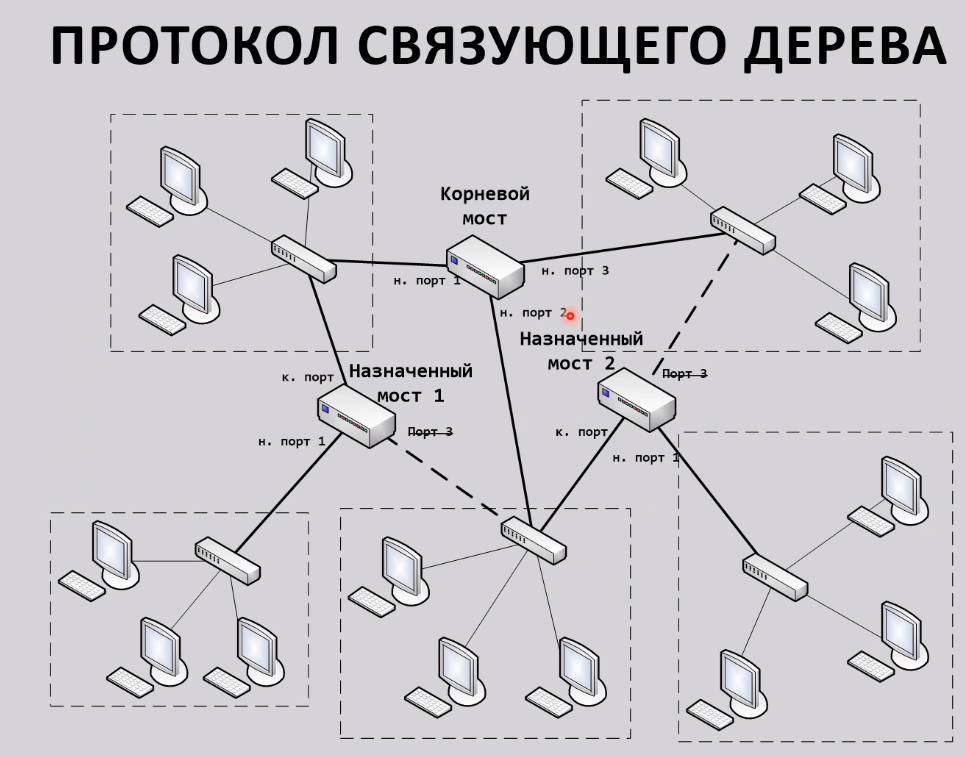
Поле Идентификатор порта (4 байта) – содержит идентификатор порта, через который было послано сообщение;

Поле Возраст сообщения (2 байта) – указывает время, прошедшее с тех пор, как корневой мост послал сообщение, послужившего основой для данного сообщения;

Поле Максимальный возраст (2 байта) – указывает временной интервал, определяющий, как долго мост может блокировать порты;

Поле Время приветствия (2 байта) – определяет, как часто корневой мост посылает сообщения;

Поле Задержка переходов (2 байта) – указывает временной интервал, через который мост должен изменить расстояние соответствующих портов после изменения топологии.



**Объединение сетей с помощью маршрутизатора**

Маршрутизатор **обеспечивает объединение сетей на сетевом уровне, посредством передачи пакетов из одной подсети в другую**.

При передаче пакетов маршрутизатор решает задачу выбора маршрута передачи. **Данный процесс называется маршрутизацией.** Маршрутизация выполняется на основании имеющейся у маршрутизатора таблицы маршрутизации.

**Таблица маршрутизации – база данных, расположенная в памяти маршрутизатора. Каждая запись таблицы маршрутизации содержит: адрес сети или узла, адрес следующего маршрутизатора на пути к пункту назначения, а также метрику (числовой показатель**), задающую предпочтительность маршруту.

Маска подсети

Алгоритм

1. Когда маршрутизатор получает пакет,
2. он извлекает адрес назначения из заголовка пакета,
3. Использует этот адрес и просматривает таблицу маршрутизации в поисках нужной записи.
4. Возможны два варианта:
   1. При наличии нескольких подходящих записей, как правило, выделяется запись с наименьшим значением метрики.
   2. Если же поиск не дал результатов, то полученный пакет отбрасывается.

Таблица маршрутизации может заполняться вручную или же автоматически с помощью протокола маршрутизации.

В первом случае речь идет о статической маршрутизации, а в другом о динамической.

Преимущество **статической маршрутизации –**

**это контроль**.

В отношении статических маршрутов администратор сети **может определять точную конфигурацию, которая не будет подвергаться изменениям**.

Недостатками такой маршрутизации становятся очевидными в крупной сети, а именно:

**трудоемкость процессов ввода** для каждого маршрутизатора,

а также при любой модификации сети **нужны ввести изменения в каждый маршрутизатор сети**.

Преимущества и недостатки динамической маршрутизаторы прямо противоположны преимуществам и недостаткам статической маршрутизации.

**Алгоритмы маршрутизации**

Основой любого протокола маршрутизации являются алгоритмы маршрутизации, которые применяются для определения оптимальных маршрутов.

Наиболее используемые:

1. **Дистанционно-векторный алгоритм маршрутизации (каждый маршрутизатор периодически рассылает своим соседним маршрутизатор вектор расстояния (Линия и дистанция) до всех известных ему сетей;**

**получив такой вектор, маршрутизатор обновляет соответствующие записи в таблице маршрутизации**;

при этом маршрутизатор добавляет к компонентам полученного вектора расстояние маршрутизатора-соседа, от которого был получен этот вектор;

в конце концов, каждый маршрутизатор будет владеть информацией обо всех имеющихся сетях и расстояниях до них;

если какие-либо записи не обновляются в течении определенного времени, то они будут удалены из таблицы маршрутизации);

1. **Алгоритм маршрутизации по состоянию канала**

**(каждый маршрутизатор поддерживает, так называемую, базу данных состояний каналов, в которой содержится информация обо всех маршрутизаторов объединенной сети (например, состояние и доступность соседей);**

**на основе базе данных состояния каналов маршрутизатор на основе алгоритма Дейстры строит дерево кратчайших путей, корнем которого является сам маршрутизатор;**

**с помощью этого дерева маршрутизатор может заполнять свою таблицу маршрутизации;**

после включения маршрутизатор рассылает всем маршрутизатором сети информацию о непосредственно подсоединенным к ним подсетям;

таким образом, у всех маршрутизаторов сети будет идентичная база данных состояния каналов;

Маршрутизатором нет необходимости периодически обмениваться маршрутной информацией, как в дистанционно-векторном алгоритме маршрутизации, вместо этого они рассылают **сообщения об обновлении состояния канала** (LSA), получив которые другие маршрутизаторы могут обновить свою базу данных состояния канала.

**Сходимость маршрутов – сходимостью маршрутов называют процесс распространения новой маршрутной информации в объединенной** сети. В идеале объединенная сеть должна находиться в состоянии, когда все маршрутизаторы располагают полной информацией обо всех подсетях объединенной сети.

В алгоритме маршрутизации по состоянию канала сходимость маршрутов происходит быстрее, чем в дистанционно-векторном. Это связано с тем, что в случае изменения сети маршрутизаторы узнают об этом практически незамедлительно, а не периодически).

**Протоколы маршрутизации**

Маршрутизация осуществляется, как между отдельными узлами, так и между автономными системами.

**Автономная система** – совокупность узлов, находящихся под единым административным управлением, и имеющих согласованную политику маршрутизации.

Протокол маршрутизации, используемый **внутри автономной системы, называется внутренним шлюзовым протоколом.**

Протокол маршрутизации, предназначенный **для объединения автономных систем, называется внешним шлюзовым протоколом.**

**Протоколы RIP**

**Протоколы RIP - прикладной протокол стека протоколов TCP/IP**, являющийся внутренним шлюзовым протоколом, основанный **на дистанционно-векторном алгоритме**.

Протоколы RIP предназначены для использования в небольших сетях. Поддерживают ряд таймеров, влияющих на процесс маршрутизации:

* таймер обновления

(задает частоту, с которой рассылаются сообщения об обновлении маршрутной информации (по умолчанию 30 секунд));

* таймер действительности маршрута

(задает временной интервал, в течении которого маршрутизатор не получает информацию об обновлении по определенному маршруту, не объявляет маршрут недействительным (по умолчанию 180 секунд));

* таймер сброса маршрута

(задает временной интервал, по истечении которого маршрут будет удален, если маршрутизатор не получал сообщение об обновлении этого маршрута (по умолчанию 240 секунд)).

Для предотвращения петель и улучшения сходимости маршрутов могут использоваться **механизмы разделения горизонта и негативный отклик**.

Механизм разделения горизонта основан на предположении, что не целесообразно отправлять информацию по маршруту в том же направлении, в котором она была получена.

Негативный отклик заключается в рассылке маршрутизатором уведомлений о недоступности того или иного маршрута.

Существует версия RIP, как для протокола IP4, так и для IP6.

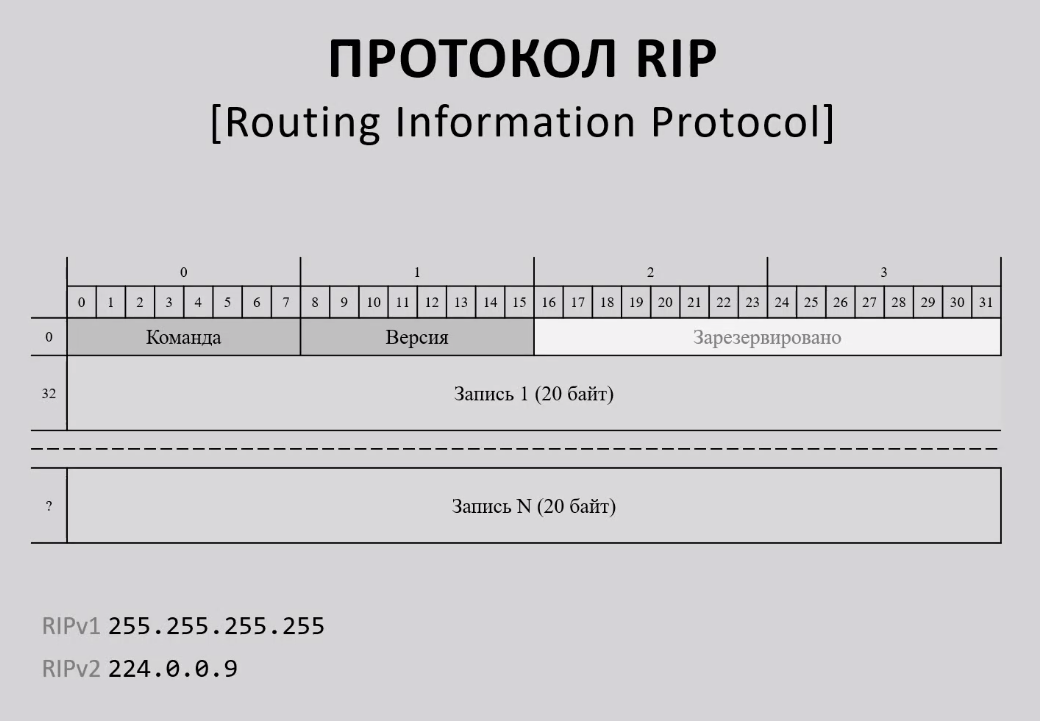
Для IP4:

* RIPv1 (используется классовая адресация);
* RIPv2 (используется бесклассовая адресация).

Обе эти версии используют транспортный протокол UDP с номером порта 520.

Одно сообщение R**IP может содержать до 25 записей. При этом, максимальный размер сообщения не превышает 512 байт.**

Для передачи сообщений в RIP 1 версии используется широковещательный адрес 255.255.255.255; в RIP 2 версии групповой адрес 224.0.0.9;



Поле Команды (1 байт) – определяет является ли сообщений запросов или ответов (1=запрос, 2= ответ);

Поле Версия (1 байт) – содержит информацию о версии протокола;

Всем зарезервированным полям присваивается нулевые значения. Каждая запись имеет фиксированную длину 20 байт.

**Запись в сообщении RIPv1**

****

Поле Идентификатор семейства адресов (2 байта) – определяет протокол сетевого уровня, используемый для адресации (для IP4 это поле принимает значение 2);

Поле Сетевой адрес (4 байта) – содержит адрес сети или узла назначения;

Поле Метрика (4 байта) – определяет число переходов, то есть **количество маршрутизаторов через которое может пройти пакет по маршруту (в запросах данное поле не используется, маршруты к сетям непосредственно подключенные к маршрутизатору имеет метрику** 0, а недоступные маршруты имеют метрику 16; небольшой диапазон метрики делает RIP непригодный для больших сетей).

**Запись в сообщении RIPv2**



Поле Идентификатор семейства адресов, поле Сетевой адрес, поле Метрика такие что и в Запись в сообщении RIPv1.

Но

Поле Признак маршрута (2 байта) – используется для того, чтобы отличить **внутренние маршруты, полученные внутри автономной системы, от маршрутов внешних, которые были получены с помощью внешних шлюзовых протоколов** (если маршрут является внутренним, то поле имеет нулевое значение; если же внешним, то ненулевое значение);

Поле Маска подсети (4 байта) – **содержит маску подсети, связанную с адресом, указанным в поле Сетевой адрес (если в поле Сетевой адрес указан адрес узла, а не сети, то используется маска 255.255.255.255).**

Поле Следующий маршрутизатор (4 байта) – содержит адрес следующего маршрутизатора на пути к пункту назначения, если **данное поле содержит неопределенный адрес 0.0.0.0, то следующий маршрутизатор является маршрутизатором отправителя сообщения.**

В RIPv2 можно использовать первую и только первую запись сообщения для аутентификации, таким образом, на маршрутные данные остается только 24 записи.

Формат первой записи



Первые 2 байта содержат значения FFFF, которые указывают, что следующие 16 байт содержат данные аутентификации.

Поле Тип аутентификации (2 байта) – определяет тип аутентификации (в настоящее время есть только один тип – использование простого нешифрованного пароля и значение данного поля тогда = 2);

Поле Информация о аутентификации (16 байт) – содержит пароль (если пароль короче 16 байт, то он дополняется справа нуль-символами).

**Протокол RIPng**

Для IP6 был разработан протокол RIPng, описанный в документе RFC 2080. Протокол RIPng не сильно отличается от предыдущих версий протокола IP.

RIPng имеет такой же формат сообщений, в котором поле «Версия» содержится значение равное 1. Для передачи сообщений RIPng используется транспортный протокол UDP 521. А также групповой адрес FF02::9.\



Поле Сетевой адрес (16 байт) – содержит адрес сети или узла назначения;

Поля Признак маршрута и Метрика – аналогичны одноименным полям в записи RIPv2. За исключением того, что метрика имеет размер 1 байт;

Поле Префикс сети (1 байт) – содержит префикс сети, связанный с указанным сетевым адресом.



В сообщениях RIPng адрес следующего маршрутизатора к месту назначения включается в специальную запись назначенного маршрута. И все записи, для которых адрес следующего маршрутизатора является вышеуказанным, группируются после записи назначенного маршрута. Другие группы записей, если они есть, аналогично указываются после записи назначенных маршрутов.

Последний байт в записи назначенного маршрута содержит значение FF (число, которое соответствует восьми 1), указывающее на то, что данная запись является записью указанного маршрута. Также имеется поле Следующий маршрутизатор (16 байт), которое содержит адрес следующего маршрутизатора на пути к пункту назначения (если данное поле содержит запись ::/128 (неопределенный адрес), то следующим маршрутизатором является маршрутизатор – отправитель сообщений).

**Протокол OSPF**

Протокол OSPF – это протокол сетевого уровня стека протоколов TCP/IP, являющийся внутренним шлюзовым протоколом маршрутизации, основанным на алгоритме маршрутизации по состоянию каналу.

Протокол OSPF преимущественно используется в крупных сетях, так как для этого у него нет ограничений, как в протоколе RIP, например, по числу переходов. Однако, в небольших сетях из-за сложности в настройке маршрутизатора, администратора, привыкшие к простоте протокола RIP, не всегда согласны на использование протокола OSPF.

Существует несколько версий протокола OSPF:

* для IP4 используется OSPFv2 (для IPv4);
* для IP6 используется OSPFv3 (для IPv6).

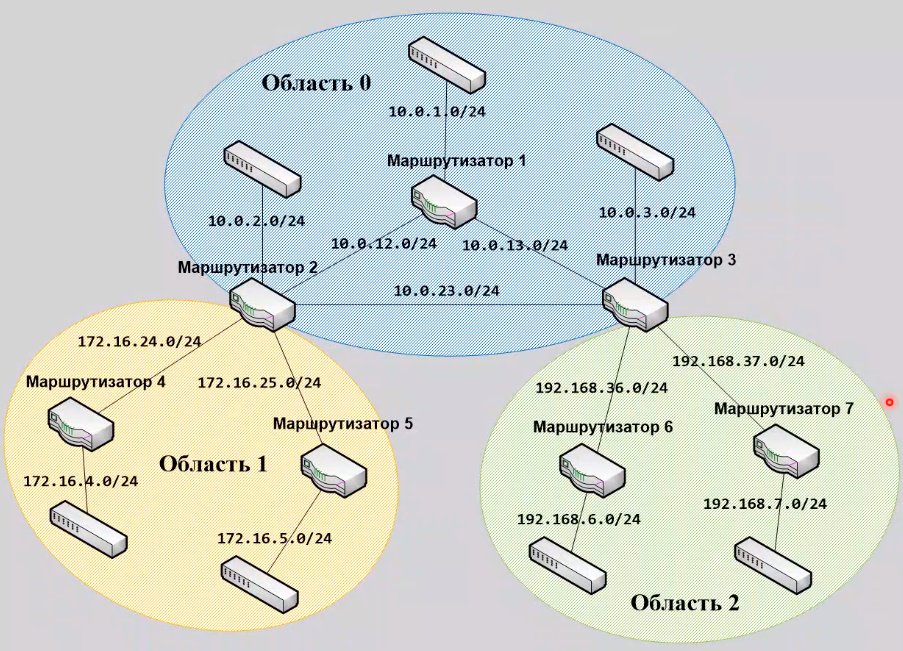
Протокол OSPF позволяет объединять смежные сети в области. Внутренние маршрутизаторы области не знают ничего о топологии за пределами своей области. Такая локализация сведений о топологии позволяет сильно снизить объем передаваемой служебной информации протокола OSPF.

При использовании областей, допущение идентичности баз данных состояния каналов на всех маршрутизаторах перестает быть верным. Каждый маршрутизатор теперь имеет базу данных состояний каналов, соответствующую области, к которой он относится. Маршрутизаторы относятся к одной области и имеют идентичные базы данных состояний каналов. Маршрутизаторы, связывающие области, называют граничными маршрутизаторами областей.

В протоколе OSPF существует специальная область, называемая магистральной, в которой соединены все остальные области. Магистраль отвечает за распространение маршрутной информации между остальными областями.

В этом примере магистральной является Область 0, при этом граничными маршрутизаторами областей являются маршрутизаторы 2, 3. Например, в маршрутизаторе 2 все записи, связанные с сетями области 1 объединены в единую запись 172.16.0.0/16, подобным образом и сети всех остальных областей представлены в записях соответствующих маршрутизаторов областей.

Чтобы уменьшить число сообщений об обновлении состоянии канала (LSA), передаваемых между маршрутизаторами, в протоколе OSPF применяется принцип назначенного маршрутизатора. На каждую область приходится по одному назначенному маршрутизатору, который выбирается исходя из своего приоритета по шкале от 0 до 255, где большие числа являются предпочтительнее.



Маршрутизатор со вторым по значимости приоритетом становится резервным назначенными маршрутизатором.

Основная функция назначенного маршрутизатора – рассылка сообщений LSA, между всеми маршрутизаторами области.

Над значением резервного назначенного маршрутизатора является обеспечение быстрого восстановления функции назначенного маршрутизатора, если тот выйдет из строя.

Протокол OSPF работает на базе протокола IP, при этом в IP-пакете для идентификации протокола OSPF, как протокола верхнего уровня, используется значение 89. Все пакеты протокола OSPF используют однотипные заголовки.

Заголовок пакета OSPFv2

СХЕМА

Поле Версия (1 байт) – содержит версию протокола OSPF;

Поле Тип (1 байт) – определяет назначение пакета OSPF, данное поле может принимать следующие значения: 1) 1 - приветственный пакет (используется для определения соседних маршрутизаторов); 2) 2 – описание базы данных (используется для распространения информации из базы данных состояний каналов); 3) 3 – запрос состояния канала (используется для осуществления запросов информации из баз данных состояния каналов); 4) 4 – обновление состояния канала (используется для распространения сообщений LSA); 5) 5 – подтверждение состояния канала (используется для подтверждения приема сообщения LSA).

Поле Длина пакета (2 байта) – содержит размер OSPF пакетов в байтах с учетом заголовка;

Поле Идентификатор маршрутизатора (4 байта) – содержит идентификатор маршрутизатора отправителя пакета;

Поле Идентификатор области (4 байта) – содержит идентификатор области, к которой относится маршрутизатор;

Поле Контрольная сумма (2 байта) – содержит значение, которое представляет собой результат вычисления контрольной суммы для всего пакета, начиная с заголовка OSPF, но без учета поля Аутентификации;

Поле Тип аутентификации (2 байта) – определяет тип применяемой в OSPFv2 аутентификации; может принимать следующие назначение: 0- не используется, 1 – простая парольная аутентификация, 2 – криптографическая аутентификация;

Поле Аутентификация (8 байт) – используется для аутентификации.

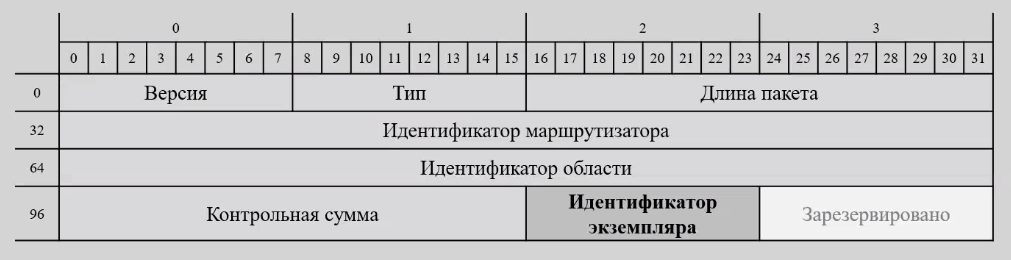


Заголовок пакета OSPFv3

В OSPF3 были удалены встроенные средства аутентификации, а вместо них для аутентификации используются средства IP6.

Поле Идентификатор экземпляра (1 байт) – используется OSPF3 для управления взаимодействием маршрутизаторов одной области; маршрутизатора могут стать соседними, только если имеют одинаковый идентификатор экземпляра.

Пакеты протокола OSPF передаются с помощью группового адреса 224.0.0.5 (OSPF2) или FF02::5 (OSPF3). Для передачи OSPF пакетов только назначенным и резервным назначенным маршрутизаторам используются другие групповые адреса.



**Внешние шлюзовые протоколы**

Основной функцией **протоколов является обмен информацией о доступности сети между граничными маршрутизаторами автономных систем**. Информация о доступности сети включается список автономных сетей, через которых проходит эта информация. Эти сведения, например, могут использоваться для построения топологии автономных систем и исключения петель.

1. EGP (внешний шлюзовой пакет) – последняя его версия описана в документе RFC 904;
2. BGP (протокол граничащего шлюза) – описан в документах RFC 17771, RFC 4271; прикладной протокол стека протоколов TCP/IP.

**Преобразование сетевых адресов**

При объединении сетей с помощью маршрутизаторов может возникнуть нехватка доступных адресов. В сетях, которые используются адреса IP4 теоретически можно иметь около четырех миллионов уникальных адресов. Однако, из-за наличия специальных IP-адресов фактическое число доступных адресов фактически меньше. Для решения данной проблемы применяется технология **NAT**, которая допускает преобразование внутренних адресов сети в адреса внешней сети, и наоборот, NAT позволяет подключить к глобальной сети локальные сети, в которых используются IP-адреса, не являющиеся уникальными в глобальной сети. При этом, узлы локальной сети могут совместно использовать один или несколько публичных IP-адресов глобальной сети. Кроме того, NAT позволяет скрыть внутреннюю структуру локальной сети. Описание в RFC 1631.

Потребность NAT отпадает при использовании IP6, когда каждый узел имеет уникальный IP-адрес.

**В NAT используются следующие понятия:**

* внутренний локальный адрес (IP-адрес, назначенный узлу во внутренней сети, как правило, этот адрес является частным);
* внутренний глобальный адрес (публичный IP-адрес, который представляет 1 или более внутренних локальных IP-адресов для внешней сети);
* внешний локальный адрес (IP-адрес, который представляет узел внешней сети для внутренней сети; этот адрес берется из адресного пространства внутренней сети);
* внешний глобальный адрес (IP-адрес, назначенный узлу во внешней сети).

Каждый **пакет из внутренней локальной сети имеет внутренний локальный адрес**, в качестве адреса отправителя, и **внешний локальный адрес, в качестве адреса получателя.**

Пока пакет находится во внутренней сети, он имеет **внутренний локальный адрес**. При переходе такого пакета во внешнюю сеть, адрес отправителя пакета **заменяется внутренним глобальным адресом**, а адрес получателя внешним глобальным адресом. Когда пакет находится во внешней сети, в качестве адреса отправителя используются внешний глобальный адрес, а в качестве адреса получателя используется внутренний глобальный адрес. Когда такой пакет оказывается во внутренней сети, адрес отправителя пакета заменяется внешним локальным адресом, а адрес получателя внутренним локальным адресом.

Обычно NAT используется маршрутизаторами, которые объединяют локальные и глобальные сети и преобразуют частные IP-адреса локальной сети в публичные IP-адреса глобальной сети и обратно. При этом все преобразования IP-адресов отслеживаются в соответствующих таблицах NAT, что позволяет при получении ответного пакета выполнить обратное преобразование адресов.

**Преобразование внутренних адресов**

В NAT различают два способа преобразования внутренних адресов:

1. Статическое преобразование (устанавливает взаимно-однозначное соответствие между внутренними локальными адресами и внутренними глобальными адресами; оно полезно, когда узел внутренней сети должен был доступен из не по фиксированному адресу);
2. Динамическое преобразование (устанавливает соответствие между внутренними локальными адресами и некоторым множеством внутренних глобальных адресов).

Пример маршрутизатора, преобразующего внутренние адреса при переходе пакета из внутренней сети во внешнюю и обратно.

Получая пакет, маршрутизатор заменяет внутренний локальный адрес отправителя 10.10.10.1 на глобальный адрес (в соответствии с записью таблицы NAT), и посылает пакет по назначению во внешнюю сеть. Если соответствующей записи в таблице NAT не существует, маршрутизатор выбирает адрес из адресного пространства внешней сети и создает новую запись. Когда маршрутизатор получает пакет с глобальным адресом, он использует таблицу NAT и преобразует адрес во внутренний глобальный адрес и посылает пакет по значению во внутреннюю сеть.

**Перегрузка глобальных адресов**

Технология NAT позволяет маршрутизаторам использовать один глобальный адрес для нескольких локальных адресов.

Когда нескольким локальным адресам ставится в соответствие один глобальный адрес, маршрутизатор использует информацию от протоколов транспортного уровня (например, номера портов TCP или UDP), чтобы перевести глобальный адрес обратно в нужный локальный адрес.

Когда маршрутизатор получает пакеты из внешней сети, он сверяется с таблицей NAT, используя протокол, глобальный адрес, номер порта, а затем преобразует глобальный адрес во внутренний локальный адрес и посылает пакет по назначению во внутреннюю сеть.

**Преобразование при перекрытии адресов**

Возможна ситуация, в которой адреса, используемые легально и нелегально во внутренней сети также используются в другой сети. Технологию NAT можно использовать, чтобы **преобразовать внутренние адреса, которые перекрываются с внешними адресами.**

**Протоколы транспортного уровня**

Протоколы транспортного уровня обеспечивают контроль над передачей данных.

В настоящее время наиболее распространены два стандартных протокола транспортного уровня – TCP и UDP.

**Порты**

После того, как данные будут доставлены на транспортный уровень интерфейса получателя, необходимо, определить – какой прикладной процесс поступает в качестве их получателя. Протоколы TCP и UDP обладают средствами идентификации прикладных процессов по номерам портов (не следует путать с портами сетевых устройств и сетевых адаптеров).  
Номера портов можно разделить на три диапазона:

1. Хорошо известные номера портов 0-1023 (номера портов данного диапазона присваиваются базовым системным службам (например, DHCP-клиент использует порт с номером 68));
2. Зарегистрированные номера портов 1024-49151 (номера портов из данного диапазона присваиваются различным коммерческим приложениям (например, порт с номером 1433 - присвоен Microsoft SQL-Server); некоторые ОС могут использовать данный диапазон для назначения временных номеров портов);
3. Динамические номера портов 49152- 65535 (если прикладной процесс явно не указывает номер порта, то ОС назначает ему временный номер порта из данного диапазона).

Регистрацией и выдачей номеров портов из первого и второго диапазонов занимается организация IANA.

Если при отправке данных, указать неиспользуемый номер порта узла назначения, то есть номер, с которым не связан ни один прикладной процесс, то эти данные будут отвержены, а в ответ отправлен ICP-пакет, сообщающий о недоступности порта узла назначения.

**Протокол UDP**

Протокол UDP – протокол транспортного уровня стека протоколов TCP/IP, описан в RFC 768. Является протоколом без установления соединения между отправителем и получателем, то есть работает в режиме простой передачи данных, поэтому UDP эффективен в приложениях, работающих по схеме запрос-ответ, так как при этом не нужно тратит ресурсы на открытие и закрытие соединения.

Другое преимущество UDP проявляется при групповой рассылке. Например, использование UDP для рассылки данных большому числу получателей можно просто отправить эти данные на широковещательный или групповой адрес. Таким образом, для отправки данных не требуется создавать множество соединений. А затем передавать данные в каждом соединении по отдельности и закрывать соединения.

Но следует иметь в виду, что UDP не обеспечивает упорядоченный прием данных. Поэтому порядок принятых данных может отличатся от порядка их отправления. В приложениях, в которых требуется сохранение порядка при получении данных, необходимо реализовать механизм упорядочивания этих данных.

Также UDP не имеет механизмов, позволяющий разбивать достаточно крупные данные на отдельные фрагменты и проводить сборку данных из последовательности фрагментов после их получения.

**UDP-дейтаграмма**

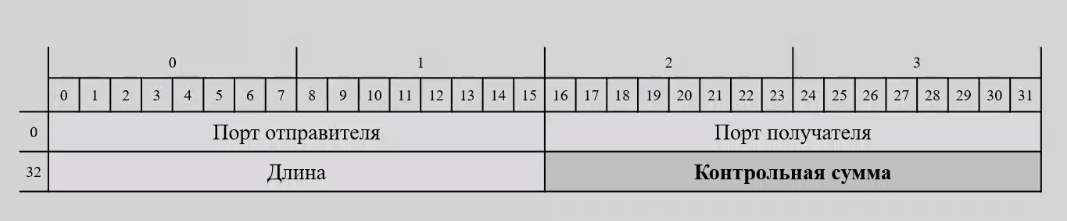
В UDP передаваемые данные инкапсулируются в дейтаграммы. Заголовок UDP-дейтаграммы имеет фиксированную длину 8 байт.

СХЕМА

Поля Порт отправителя, Порт получателя (2 байта) – идентифицируют соответственно процесс отправителя и получателя (поле порт отправителя может иметь нулевое значение);

Поле Длина (2 байта) – содержит размер UDP-дейтаграммы в байтах, учитывая размер заголовка и данных;

Поле Контрольная сумма (2 байта) – является необязательным, но если используется, то вычисление контрольной суммы вычисляется по отношению ко всему содержимому UDP-дейтаграммы, в которую входит UDP-заголовок (кроме поля контрольная сумма) и данные, а также псевдо-заголовок, получаемый из IP-заголовка, который используется только для вычисления контрольной суммы.





**Протокол TCP**

Протокол TCP – протокол транспортного уровня стека протоколов TCP/IP, описан в RFC 793.

Основным отличием TCP от UDP является то, что протокол TCP обеспечивает надежную доставку данных. Протокол TCP решает задачу обеспечения надежной доставки данных путем установления логического соединения между узлами сети. Благодаря этому соединению, TCP может следить, чтобы передаваемые данные не были потеряны, не были продублированы и пришли к получателю в том же порядке, в каком они были отправлены.

Все эти TCP-соединения образуют дуплексный канал связи. При этом широковещательная рассылка протокола TCP не поддерживается.

**TCP-сегмент**

Протокол TCP рассматривает поступающие данные, как неструктурированный поток байтов. Перед передачей на сетевой или прикладной уровень, данные помещаются в специальные буферы TCP. Для передачи на сетевой уровень формируется TCP-сегмент, который состоит из заголовка и данных из буфера. Размер сегментов определяется реализацией протокола TCP. Минимальный размер заголовка TCP-сегмента занимает 20 байт.

СХЕМА

Поля Порт отправителя и Порт получателя (2байта) – идентифицируют соответственно процесс отправителя и получателя;

Поле Порядковый номер (4 байта) – указывает номер байта, который определяет смещение TCP-сегмента в потоке передаваемых данных; данное поле используется получателем для восстановления исходных данных из полученных TCP-сегментов.

Поле Номер подтверждения (4 байта) – содержит номер байта, ожидаемого в следующем сегменте; данное поле действительно и используется только с флагом ACK;

Поле Длина заголовка (4 бита) – указывает размер заголовка TCP-сегмента, измеряемый в 32-битных словах; размер заголовка зависит от значений, задаваемых в поле Параметр; для выравнивания заголовка к размеру кратному 4 байтов, используется дополнение в виде последовательности нулей;

Поле Зарезервировано (6 бит) – зарезервировано;

Поле Флаги (6 бит) – содержит шесть флагов, содержащих служебную информацию о типе данного сегмента:

1) URG(срочное сообщение);

2) ACK (подтверждение принятия сегмента = 1);

3) PSH (флаг, с помощью которого отправитель просит получателя передавать данные прикладному уровню сразу после получения сегмента, а не хранить их в буфере);

4) RST (используется для отказа от неверного сегмента или от попытки создать соединение);

5) SYN (используется для установления соединения);

6) FIN (используется для завершения соединения).

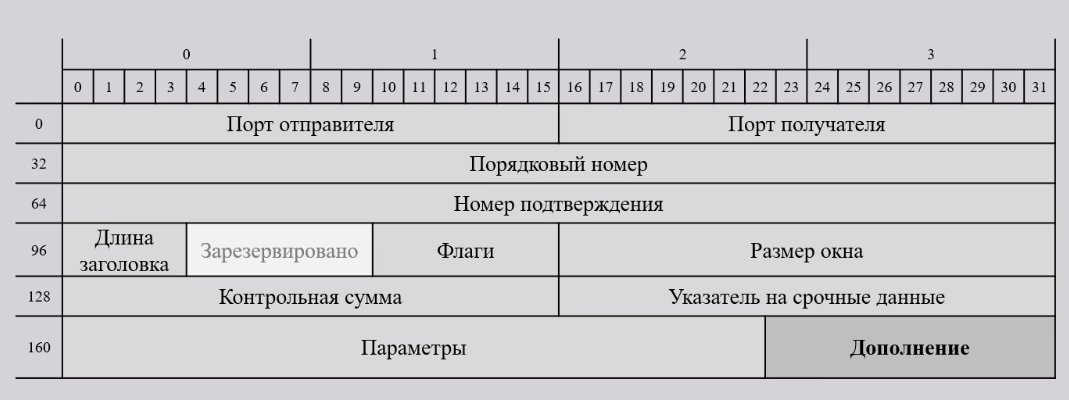
Поле Размер окна (2 байта) – используется для указания информации о том, какой размер данных узел готов принять в следующем TCP-сегменте;

Поле Контрольная сумма (2 байта) – предназначена для выявления ошибок; контрольная сумма вычисляется из заголовка TCP-сегмента (кроме поля контрольная сумма), данных и псевдо-заголовка; для TCP используется аналогичный псевдо-заголовок, что и для UDP, при этом в поле Протокол псевдо-заголовка указывается значение равное 6, которое соответствует протоколу TCP;

Поле Указатель на срочные данные – указывает на последний байт срочных данных в сегменте; данное поле используется только вместе с флагом URG, который устанавливается, если какие-то данные нужно отправить вне очереди;

Поле Параметры (переменная длина) – используется для согласования параметров при установлении TCP-соединения; например, с помощью данного поля можно указать максимальный размер сегмента; при отсутствии параметров данное поле имеет длину 1 байт и содержит нулевое значение;

Поле Дополнение – с помощью него дополняется размер заголовка до числа, краткого 4 байтам.



**Процесс соединения TCP-соединения**

В TCP определены два режима установления TCP-соединения: активный и пассивный.

В активном режиме одна сторона (клиент) отправляет другой стороне (серверу) TCP-сегмент с запросом на установление соединения.

В пассивном режиме одна сторона (сервер) ожидает запрос на установление TCP-соединения от другой стороны (клиент).

В TCP установление соединения представляет собой процесс, состоящий из нескольких этапов:

1. Сервер ожидает от клиента TCP-сегмент с запросом на установление соединения;
2. Клиент для установления соединения отправляет серверу SYN-сегмент , то есть TCP-сегмент, в котором установлен флаг синхронайз и сброшен флаг ; в син-сегменте так же указываются номера портов получателя и отправителя, начальный порядковый номер и максимальный размер сегмента;
3. После получения син-сегмента, сервер для подтверждения открытия соединения отправляет клиенту TCP-сегмент с флагами синхронайз и ACK; в этом TCP сегменте сервер также указывает свой начальный порядковый номер и максимальный размер сегмента;
4. После получения TCP-сегмента, подтверждающего открытие соединения, клиент выполняет одно из следующих действий:

* если клиент всё ещё желает установить соединение, то серверу отправляется TCP-сегмент с флагом S для завершения процесса установления соединения;
* если клиент, по каким-либо причинам, более не хочет устанавливать соединение, серверу отправляется TCP-сегмент с флагом резет (для отказа от установления соединения).

Для завершения TCP-соединения отправляет так называемый FIN-сегмент, то есть TCP-сегмент с установленным флагом FIN. Когда отправленный FIN-сегмент будет подтвержден, направление передачи закрывается, тем не менее, данные ещё могут передаваться в противоположном направлении, то есть другая сторона всё ещё может отправлять данные.

TCP-соединение считается завершенным, только если оба направлении передачи закрыты.

Для завершения TCP-соединения требуются 4 сегмента: по одному FIN-сегменту и по одному TCP-сегменту с флагом ACK, в каждом из двух направлений.

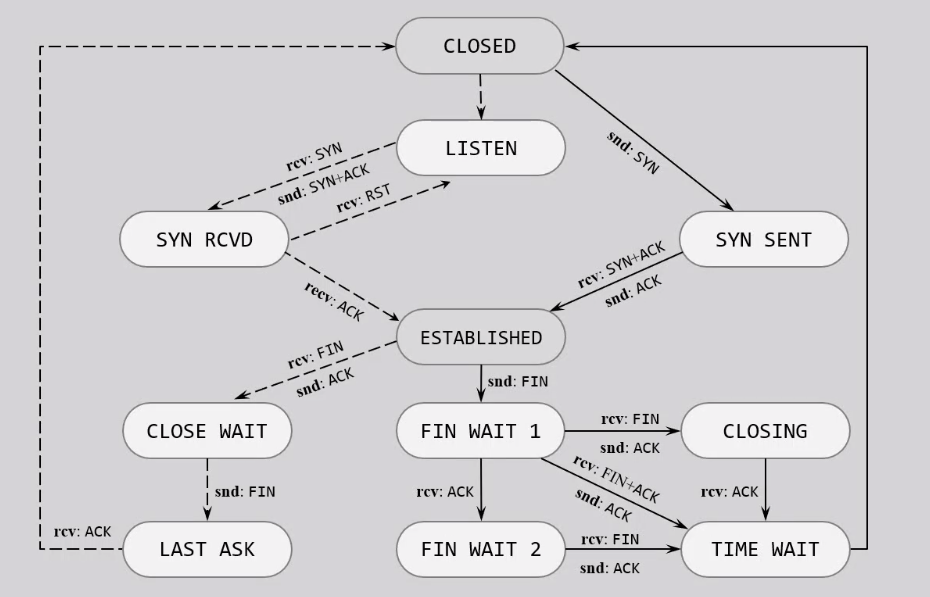
Иногда первый сегмент с флагом ACK и второй FIN-сегмент могут объединяться в один сегмент, чтобы уменьшить их количество до трех.

Если подтверждение на FIN-сегмент не будет получено в течении удвоенного максимального времени жизни MSL-сегмента, то отправитель FIN-сегмента завершается TCP-соединение. По умолчанию максимальное время жизни сегмента 4 минуты. Если другая сторона заметит, что ей никто не отвечает, и тоже завершит TCP-соединение.

**Состояния TCP-соединения**

Для TCP-соединения определено 11 различных состояний.

1. Каждое новое TCP-соединение начинается с фиктивного состояния CLOSED (представляет состояние, когда не существует TCP-соединение);
2. Состояние LISTEN (сервер ожидает SIN-сегмент);
3. Состояние SYN SENT (клиент отправил SIN-сегмент и ожидает его подтверждение);
4. Состояние SYN RCVD (сервер получил SIN-сегмент, отправил TCP-сегмент с флагами FIN и ACK и ожидает подтверждения);
5. Состояние ESTABLISHED (соединение установлено);
6. Состояние FIN WAIT 1 (клиент сообщил о завершении соединения, отправив FIN-сегмент, и ожидает подтверждения);
7. Состояние FIN WAIT 2 (клиент получил подтверждение FIN-сегмент);
8. Состояние CLOSING (клиент, получив вместо подтверждения FIN-сегмент без флага ACK от сервера, продолжает ожидать подтверждения своего FIN-сегмента; это означает, что обе стороны решили «одновременно» завершить соединение);
9. Состояние TIME WAIT (клиент, прежде чем окончательно завершить соединение, выдерживает паузу, которая будет длиться два максимальных времени жизни сегмента);
10. Состояние CLOSE WAIT (сервер получит FIN-сегмент);
11. Состояние LAST ACK (сервер ожидает подтверждения FIN-сегмента).



**Передача данных в TCP**

При передачи данных используется механизм, который гарантирует, что каждый отправленный TCP-сегмент будет обязательно доставлен.

После того, как TCP-сегмент был отправлен, начинается отсчет, так называемого таймаута ретрансмиссии (RTO). Когда TCP-сегмент будет получен, для подтверждения обратно отправляется другой TCP-сегмент с флагом ACK. Если до того, как закончится время таймаута ретрансмиссии так и не будет получение подтверждение считается, что произошла ошибка, и TCP-сегмент отправляется повторно.

Отметит, что истечении времени таймаута ретрансмиссии не означает, что отправленный TCP-сегмент не дошел до получателя. Например, TCP-сегмент мог задержаться в сети дольше, чем это предусмотрено, или же мог потеряться TCP-сегмент с подтверждением.

Кроме того, при передачи данных в TCP используется **механизм скользящего окна,** которой гарантирует, что даже если TCP-сегменты были получены не в том порядке, в котором были отправлены, получатель все равно сможет собрать из них исходные данные. Для этого каждому байту в потоке передаваемых данных присваивается порядковый номер (он не передается вместе с каждым байтом, вместо этого в заголовке TCP-сегмента указывается порядковый номер начального байта, а порядковые номера остальных байтов можно легко вычислить).

Каждый участник TCP-соединения имеет **окно-приема**, представляющего собой диапазон порядковых номеров байтов. Наименьшее значение соответствует левой границе окна приема – это порядковый номер следующего ожидаемого байта. Наибольшее значение соответствует правой границе окна приемника. Это порядковый номер последнего байта, для которого в TCP-соединении есть место в буфере.

Когда принимается TCP-сегмент все байты вне окна приема отбрасываются. Это касается, как уже принятых ранее байтов (с номерами левее окна приема), так и байтов, для которых нет места в буфере (с номерами правее окна приема). Если порядковый номер начального байта, в TCP-сегменте больше, чем порядковый номер следующего ожидаемого байта, то, значит, что данный сегмент принят не по порядку. В большинстве реализации TCP такой сегмент помещается в очередь и находится в ней, пока не будут получены пропущенные байты. Если же порядковый номер первого байта в сегменте совпадает со следующим ожидаемым, то окно приема сдвигается вправо на число байт в принятом сегменте.

Помимо окна приема у каждого участника TCP-соединения также имеется **окно передачи**. Окно-передачи разделено на 2 части:

1. В одной расположены отправленные байты, которые ещё не подтверждены;
2. А в другой байты, которые ещё не отправлены.

**Прикладные протоколы TCP/IP**

Прикладные протоколы обеспечивают взаимодействие различных сетевых приложений.

**Протокол FTP**

Протокол FTP – это прикладной протокол стека протоколов TCP/IP, предназначенный для передачи файла, описан в RFC 959.

Передача файлов по протоколу FTP осуществляется с использованием протокола TCP/IP. Между FTP-клиентом и FTP-сервером устанавливаются два TCP-соединения (командное соединение и соединение для передачи данных).

Командное соединение представляет собой TCP-соединение, по которому FTP-клиент отправляет команды, а FTP-сервер ответы на полученные команды. По умолчанию FTP-сервер для данного соединения использует порт с номером 21. Командное соединение начинается FTP-клиентом, а завершается FTP-сервером по команде от клиента.

Соединение для передачи данных представляет собой TCP-соединение, установленное специально для передачи файлов между клиентом и сервером. Если данное соединение устанавливается в активном режиме, то FTP-клиент обычно использует порт с номером 20. Соединение для передачи данных инициируется FTP-клиентом, а заканчивается FTP-сервером при окончании процесса передачи данных.

Команды FTP – представляют собой текстовые строки, содержащие ключевое слово, за которым может следовать один или несколько аргументов. Строка завершается символом возврата каретки CR (13) и символом перевода строки LF (10).

Команда ABOR – отменяет выполнение предыдущей команды.

Команда CDUP – изменяет рабочий каталог на вышестоящий.

Команда CWD каталог – изменяет рабочий каталог.

Команда DELE файл – удаляет файл.

Команда HELP [команда] – получает сведения о доступных командах на сервере.

Команда LIST [каталог] – получает список содержимого каталога (список передается через соединение для передачи данных).

Команда MKD каталог – создает новый каталог.

Команда NLST [каталог] – получает список содержимого каталога в более краткой форме, чем команда LIST (список передается через соединение для передачи данных).

Команда NOOP - проверяет связь с ftp сервером

Команда PASS пароль – указывает пароль пользователя.

Команда PASV – указывает, что следующее соединение должно быть установлено сервером в пассивном режиме. Сервер вернет IP-адрес и номер порта, к которому нужно подключиться.

Команда PORT h1, h2, h3, h4, p1, p2 – команда, через которое сервер будет устанавливать активное положение.

Команда PWD – возвращает рабочий каталог.

Команда QUIT – завершает командное соединение.

Команда REIN – перезапускает командное соединение.

Команда RETR файл – получает файл с FTP-сервер (перед командой должна следовать команда REIN или PWD).

Команда RMD каталог – удаляет каталог.

Команда STOR файл – передает файл на FTP-сервер.

Команда STOU – передает файл на FTP-сервер с присвоением уникального символа, имя нового файла возвращается в ответе.

Команда USER имя-пользователя – идентифицирует пользователя, пытающегося получить доступ к файлам.

После получения и исполнения команды, FTP-сервер отправляет FTP-клиенту ответ, содержащий специальный код из трех цифр, и отделенную пробелом поясняющую фразу, которая разъясняет причину именно такого ответа.

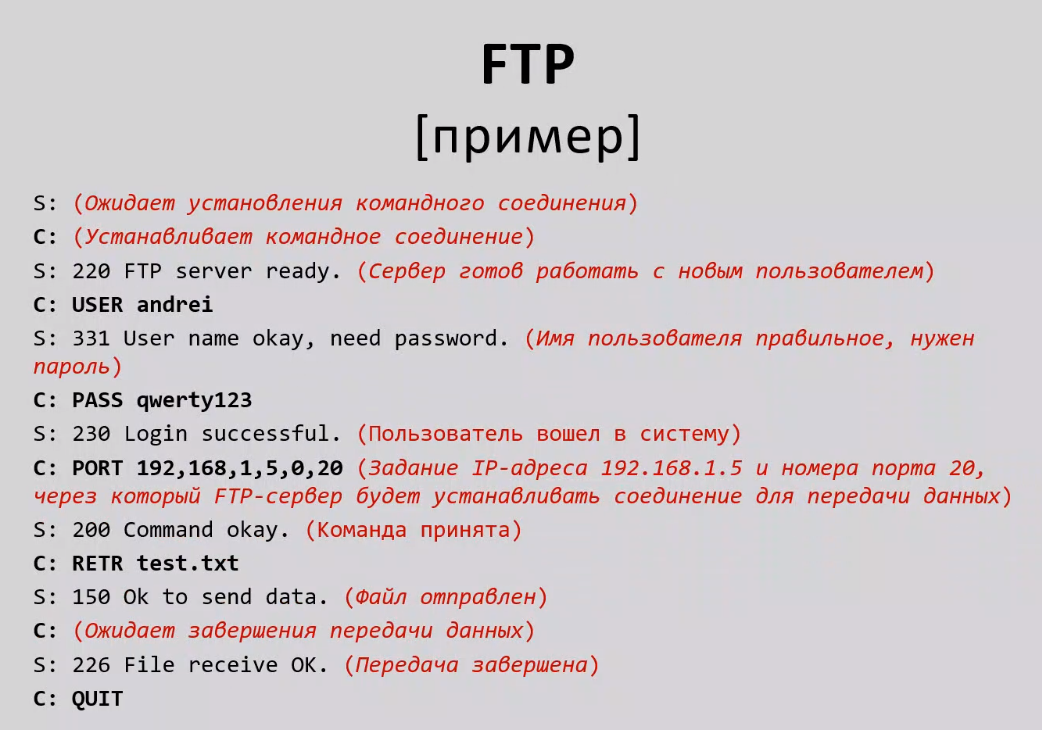
Первая цифра в коде ответа FTP-сервера трактуется следующим образом:

* 1 – команда принята к выполнению, но ещё не завершена;
* 2 – выполнение команды успешно завершено;
* 3 – команда принята и ожидается дополнительная команда;
* 4 – в данный момент команда не может быть выполнена;
* 5 – команда принципиального не может быть выполнена.

Вторая цифра в коде ответа:

* 0 – синтаксическая ошибка, либо ничего не означает;
* 1 – информационное сообщение;
* 2 – сообщение относится либо к управляющему соединению, либо к соединению данных;
* 3 – сообщение об идентификации пользователя и его правах;
* 4 – не определено значение;
* 5 – сообщение о состоянии файловой системы.

Третья цифра в коде ответа окончательно идентифицирует ответ.



**Протокол HTTP**

5/12/2020

Протокол HTTP – прикладной протокол стека протоколов TCP/IP, предназначенный для передачи данных.  
Протокол HTTP может также использоваться в качестве транспортного для других прикладных протоколов, например, протокол SOAP или RPC-XML. Текущей версией протокола HTTP является версия 1.1 (принято записывать HTTP/1.1). Определено в RFC/2616

Передача данных по HTTP осуществляется посредством протокола TCP. По умолчанию HTTP-сервер использует порт с номером 80. Можно поставить любой порт.

**HTTP-сообщение**

Каждое HTTP-сообщение состоит из трех частей, которые передаются в следующем порядке.

1. Стартовая строка (определяет тип HTTP-сообщения);
2. Заголовки (характеризуют содержимое HTTP-сообщения, параметры передачи и прочие сведения);
3. Тело сообщения (содержит данные HTTP-сообщения).

Заголовки и Тело сообщения могут отсутствовать, но Стартовая строка является обязательной. Заголовки должны отделяться от тела сообщения **хотя бы одной пустой строкой**.

Стартовая строка запроса HTTP-клиента и ответа HTTP-сервера различны. Но в обоих случаях стартовая строка заканчивается символами возврата каретки или перевода строки.

Стартовая строка запроса HTTP-клиента указывает метод запроса, ресурс и версию протокола.

2 - класс кодов, информирующих о случаях успешного принятия и обработки запроса HTTP-клиента.

3 – класс кодов что для успешного выполнения необходимо делать другой запрос.

4 – класс кодов, указывающих на ошибки в запросах HTTP-клиентах.

5 – класс кодов, указывающих на ошибке выполнения операций по вине HTTP-сервере.

Параметр метод – указывает какая операция должна выполняться HTTP-сервером. Метод URI – указывает на ресурс, над которым должна проводиться операция.

Методы:

Delete – для указания ресурса который надо удалить. Сервера только проверяют этот метод. Не используется.

Get – для получения содержимого указанного ресурса.

HEAD – для получения метаданных или проверки наличия указанного ресурса.

Options – для определения возможностей HTTP-сервера

POST – для передачи данных указанному ресурсу. Предполагает, что будет производиться обработка запроса.

Put – используется на загрузки содержимого запроса на ресурс.

Стартовая строка ответа: HTTP/версия код- состояния пояснение

Классы состояний RFC/22116?:

1 – коды информирующие о процессе передачи.

2- коды информирующие о случаях успешного принятия запроса от клиента

3 – для успешного выполнения нужно еще что-то (перенаправление )

4 – 404 ошибка)

5 – ошибки выполнения операций по вине сервера 503 – внутренняя ошибка сервера

GET/index.html http/1.1

Ответа: http/1.1 200 ОК

**Заголовки HTTP**

Заголовок HTTP представляет собой текстовую строку, содержащую разделенную двоеточием пару параметров значения. Строка завершается символами возврата каретки и перевода строки. (CR , LF – именно для протокола)

Все заголовки разделяются на 4 основные группы:

1. Основные заголовки (должны включаться в каждое сообщение HTTP-клиента и сервера);
2. Заголовки запроса (используются только в запросах HTTP-клиентов);
3. Заголовки ответа (используется только в ответах HTTP-серверов);
4. Заголовки сущности (сопровождают каждую сущность HTTP-сообщения).

Этот список не полон, а полный список можно найти в документе .

Заголовки начинаются с XHTTP-Requested – только для тех кто принимает.

(примеры, рисунок)

Сессия:

Хотим получить содержимое:

Современные браузеры с Mozilla/5.0

Date – когда обновлялась страница крайний раз.

Chunked – страница загружается частями.

**Протокол SMTP**

Протокол SMTP - прикладной протокол стека протоколов TCP/IP, предназначенный для **отправки** электронной почты. Отправка (только отправка) электронной почты по протоколу SMTP осуществляется посредством протокола TCP, по умолчанию используется порт с номером 28.

Документ: RFC 5321.3

Команды SMTP – представляют собой текстовые строки, содержащие ключевое слово, за которым может следовать аргумент. Строка завершается символами для возврата каретки CR (13) и переводом строки LF (10).

1. Команда DATA (инициирует передачу почтовых данных; после этой команды SMTP-сервер трактует дальнейшие строки, как почтовые данные от отправителя; почтовые данные завершаются строкой, содержащей только точку – 3 байта, потому что еще CR и LF);
2. Команда EHLO (параметр имя-клиента) (используется для представления SMTP-клиента серверу SMTP);
3. Команда HELO (параметр имя-клиента) (используется для представления SMTP-клиента серверу SMTP; данная команда является устаревшей и должно использоваться для поддержки только старых клиентов SMTP);
4. Команда HELP (получает сведения о доступных командах на SMTP-сервере);
5. Команда MAIL FROM: <имя отправителя> (инициирует передачу и идентифицирует отправителя электронного письма); То , что мы видим при заполнении в почте - это связано с Mailформатом. Описание письма. А эти команды только для идентификации.
6. Команда NOOP (проверяет связь с SMTP сервером); В TCP если здесь долго не происходит обмен сообщениями, то сервер может решить, что соединения больше не существует.
7. Команда QUIT (завершает соединение);
8. Команда RCPT TO: <имя отправителя> (идентифицирует получателя электронного письма). Сервер отправляет код из 3 цифр, и поясняющее сообщение. RFC 821, 1869.

После получения команды, SMTP-сервер отправляет в ответ специальный код из 8-ми цифр и отдельную пробелом поясняющую фразу.

(рисунок)

**Протокол POP3**

Протокол POP3 (Post office protocol)– прикладной протокол стека протоколов TCP/IP, предназначенный для получения электронной почты с почтового сервера.

RFC 1939.

При использовании протокола POP3 все электронные письма загружаются клиентов и затем удаляются сервером. Все дальнейшие действия с сервером будут производиться только на клиенте. Другими словами, протокол POP3 не предоставляет возможности по манипуляции с электронной почты на сервере.

Для этого используется более продвинутый протокол – IMAP (Internet Message Access Protocol). RFC 3501.

Практически все современные клиенты и серверы электронной почты поддерживают оба протокола.

Получение электронной почты по протоколу POP3 осуществляет с использованием протокола TCP (по умолчанию POP3-сервер использует порт с номером 110).

Команды POP3 – представляют собой текстовые строки, содержащие ключевое слово, за которым может следовать один или несколько аргументов. Строка завершается символами для перевода каретки и возврата строки.

1. APOP имя digest (идентифицирует почтовый ящик, письма с которого необходимо получить и указывает пароль к почтовому ящику (пароль – digest); пароль хэшируется с помощью алгоритма MD5);
2. DELE номер-сообщения (помечает заданное почтовое сообщение, как удаленное; все последующие обращения к этому сообщению будут вызывать ошибку; POP3-сервер не удаляет почтовые сообщения, пока не завершено соединение); помечает как удаленное, после закрытие удалит.
3. LIST [номер-сообщения] (если аргумент указан, то POP3-сервер возвращает строку, содержащую информацию о заданном сообщении; если же аргумент не указан, то POP3-сервер возвращает для каждого сообщения в почтовом ящике, строку с информацией о нем; каждая такая строка состоит из номера сообщения, после которого через пробел следует точный размер сообщения в байтах; список строк завершается строкой, содержащей только точку; сообщения, помеченные, как удаленные, не учитываются);нумерация с учетом удаленных.
4. NOOP (проверяет связь с POP3-сервером);
5. PASS пароль (указывает пароль к почтовому ящику);
6. QUIT (завершает соединение);
7. RETR номер-сообщения (получает содержимое с заданного почтового сообщения; нельзя обращаться к удаленным сообщениям);
8. STAT (возвращает число сообщений в почтовом ящике и размер сообщений в байтах (октеттах – конкретно указывают, что используются 8 бит));
9. TOP номер-сообщения количество-строк (получает указанное количество строк из содержимого, заданного в почтовом сообщении; нельзя обращаться к удаленным сообщениям); Например выводим список писем, а точнее его заголовки, если нет необходимости загружать все письмо.
10. USER имя (идентифицирует почтовый ящик, письма которого необходимо получить).

После получения команд, POP3-сервер отправляет в ответ положительный или отрицательный индикатор состояния и отделенную пробелом информационную строку.

(рисунок)

Указывается точный размер. Чтобы его можно было получить. Иначе будет читать пока не завершиться.

**Протокол TELNET**

Протокол TELNET – прикладной протокол стека протоколов TCP/IP, позволяющий подключиться к удаленному компьютеру и передавать определенные команды для доступа к его командам в терминальном режиме.

Протокол TELNET на транспортном уровне использует протокол TCP (по умолчанию сервер TELNET использует порт с номером 23).

RFC 854.

*RDP – для удаленного доступа к рабочему столу.*

Протокол: команды пишем у себя, передаются с протоколом TELNET, обрабатываются на сервера, возвращаются и выводится результат у нас.

(рисунок)

**Сетевой виртуальный терминал**

Концепция сетевого виртуального терминала позволяет обеспечить доступ к удаленному компьютеру с любого терминального устройства. Отвечает за перевод приходящих по сети кодов, в исходящие коды. Не обеспечивает безопасности.

По умолчанию подразумевается, что сетевой локальный терминал использует символы с кодами от 32 до 126, а также 33 управляющих символа от 0 до 31 и 127 (например, символ табуляции, разврата коретки и.т.д., 9 – HT – горизонтальная табуляция).

Последовательность CR и LF должна обрабатываться, как окончание строки.

Последовательность CR и 0 обрабатывается, как возврат каретки.

Использовать CR по другому не требуется.

Команды TELNET – в отличии от других прикладных протоколов, команды передаются не как текстовая строка, а в виде последовательности байтов.

1. SE (завершает согласование параметров) 240 ;
2. NOP (нет операций) 241 – для связи с сервером, проверка;
3. DM (метка данных) 242 ;
4. BRK (прерывание) 243 ;
5. IP (прерывание процесса) 244;
6. AO (отмена ввода) 245;
7. AYT (проверка связи) 246 ;
8. EC (удаление предыдущего символа) 247 удалится уже отправленный на сервер символ?;
9. EL (удаление строки) 248 ;
10. GA (ожидание передачи данных)249;
11. SB (начало согласования параметров)250 ;
12. WILL (указывает на желание выполнять или подтверждает, что сейчас выполняется указанный параметр) 251;
13. WONT (указывает на отказ начать или продолжить исполнять указанный параметр) 252;
14. DO (требование того, чтобы другая сторона исполнила или подтвердила исполнение указанного параметра) 253;
15. DONT (требование того, чтобы другая сторона остановила исполнение или же подтвердила, что указанный параметр не исполняется) 254;
16. IAC (указывает на то, что следующий байт нужно интерпретировать, как команду) 255.

Каждая команда TELNET начинается с IAC -> <код команды>. 255 243 – 243 распознается как команда

**Согласование параметров**

Сетевой виртуальный терминал предоставляет минимальный необходимый набор параметров, который позволяет взаимодействовать даже с самым примитивный терминальным устройством. Однако, терминальные устройства могут обладать большими возможностями. А согласование параметров позволяет использовать эти возможности.

IAC-> WILL/WONT/DO/DONT -> <имя параметра>

Параметры могут быть согласованы в любой момент времени с помощью команд WILL/WONT/DO/DONT. Команды параметра и их смысл описываются в отдельных документах RFC. 3 байт код параметра.

При согласовании для каждой команды согласования есть различные возможные ответы.

(рисунок)

При согласовании параметров могут быть возвращены дополнительные данные, в этом случае нужно использовать команды SB (начало согласования) и SE(конец согласования) .

**Администрирование информационных сетей**

Для обеспечения надежной, эффективной и безопасной работы сложных информационных сетей, как правило, используется система управления сетями.

Система управления сетями (NMS), которая собирает информацию по состоянию каждого сетевого устройства в сети и предоставляет эту информацию сетевым администраторам.

В небольших сетях, напротив, более целесообразно использование небольших программ конфигурирования и управления, которые предоставляются производителями сетевых устройств. Однако, при расширении сети, рано или поздно, придется отказаться от таких программ и заменить их системой управления сетями.

**Задачи систем управления сетями**

Существуют рекомендации ITU-T.X.700 и стандарт ISO 7498-4, согласно которым задачи систем управления сетями можно разделить на группы:

1. управление конфигурации сети (задача этой группы заключается в конфигурировании параметров, как отдельных элементов сети, так и всей сети в целом; для элементов сети могут изменяться таблицы коммутации и маршрутизации, задаваться параметры сетевых интерфейсов и тому подобное; для сети, в целом, управление конфигурацией обычно начинается с построения топологической карты сети и изменением связи между элементами сети);
2. обработка ошибок (в эту группу входят выявления, определения и устранения последствий сбоев и отказов в работе сети);
3. анализ производительности и надежности (задачи этой группы связаны с оценкой накопленной статистической информацией, таких параметров, как пропуская способность каналов связи, интенсивность трафика и тому подобное);
4. управление безопасностью (задача этой группы включает в себя контроль доступа к ресурсам сети и обеспечение сохранности данных при передачи их по сети; базовыми элементами управления безопасностью являются процедуры аутентификации пользователя, назначение и проверка прав доступа к ресурсам сети);
5. учет работы в сети (регистрация времени использования в различных ресурсах сети).

**Структура систем управления сетями**

Система управления сетями состоит из трех компонентов.

менеджер – программный модуль, осуществляющий удаленным управлением сетевыми устройствами;

агент – программный модуль, предоставляющий менеджерам возможность управлять сетевыми устройствами;

Management Information Base - база данных агента, в которой хранятся объекты, содержащие информацию о текущем состоянии управляемого сетевого устройства.

**Менеджеры и агенты**

Менеджер устанавливается на отдельном компьютере, который также выполняет роль консоли управления для сетевого администратора. Менеджер может взаимодействовать сразу с несколькими агентами, обрабатывая получаемые от них данные и передавая им управляющие воздействия.

Агенты, как правило, встраиваются в управляемое сетевое устройство (маршрутизатор, принтер и др.), но могут также устанавливаться на отдельном компьютере, который связан с управляемым сетевым устройством каким-либо интерфейсом.

В больших информационных сетях система управления обычно строится на базе нескольких менеджеров. На это есть ряд причин:

* единственный менеджер (программный модуль, управляемый, иногда называют диспетчером) не обеспечит необходимую производительность, так как он будет вынужден обрабатывать весь поток сообщений от агента, что при большом количестве управляемых сетевых устройств потребует очень высокопроизводительной платформы для работы менеджера и перегрузит сеть служебной информацией (наличие нескольких менеджеров позволяет разделить между ними нагрузки);
* при отказе единственного менеджера будет потеряно управление сетью;
* в большой сети целесообразно располагать в каждом географическом пункте отдельный сетевой администратор, управляющий своей частью сети, а это удобно реализовывать с помощью отдельных менеджеров.

При наличии нескольких менеджеров, чаще всего используется два подхода к их соединению:

1. одноразовая структура (в случае одноразовых сетей, каждый менеджер управляет своей частью сети на основе информации, получаемой от своих агентов; координация работ менеджером достигается за счет обмена информацией между каждым менеджером; такой подход построения системы управления сетью сегодня является неэффективным и устаревшим); СХЕМА
2. иерархическая структура (в случае иерархических связей каждый менеджер нижнего уровня выполняет также функции агента для менеджера верхнего уровня; такой менеджер-агент работает с базой управляющей информацией в своей части сети, в которой собирается именно та информация, которая нужна именно менеджеру верхнего уровня для управления сетью, в целом; такой подход построения системы управления сетью позволяет сократить объемы информации, циркулирующей между менеджерами, что приводит к гораздо более эффективному управлению). НЕ ЗАБУДЬ СХЕМА

**База управляющей информации MIB**

Вся информация, необходимая для управления сетевым устройством, концентрируется в базе управляющей информацией – **SMI – структура управляющей информации**. База MIB имеет иерархическую структуру, каждый объект которой является определенной характеристикой управляемого устройства.

Имеется некая единая общая структура MIB, а также правила её дальнейшего формирования.

При описании объектов MIB используется формальный язык ASN.1. Каждый объект MIB имеет краткое и полное имя, а также уникальный идентификатор, который однозначно определяет объект в иерархии MIB.

При описании объекта MIB также учитываются следующие поля:

* синтаксис [Syntax] – указывается тип данных и возможное значение объекта;
* Доступ [Access] – указывается уровень доступа к объектам (только чтение, чтение и запись, только запись, недоступно);
* Состояние [Status] – где указывается статус объекта (обязательный, необязательный, устаревший);
* Дескриптор [Description].

На сегодняшний день существует несколько версий базы MIB, но на практике используется только MIB-I – RFC 1156, MIB-II.

В MIB-I определено 114 стандартных объектов, которые можно разделить на 8 групп:

1. System (содержит общую информацию об устройстве, например, название и версия оборудования, операционной системы, сетевого программного обеспечения и тому подобное);
2. Interfaces (параметры интерфейса сетевых устройств, например, их количество, тип, максимальный размер пакета и тому подобное);
3. Address Translation Table (содержит описание соответствия между сетевыми и физическими адреса, например, по протоколу ARP);
4. IP (данные относящиеся к протоколу IP);
5. ICMP (данные относящиеся к протоколу ICMP);
6. TCP (данные относящиеся к протоколу TCP);
7. UDP (данные относящееся к протоколу UDP);
8. EGP (данные относящиеся к внешним шлюзовым протоколам).

Из перечня групп, очевидно, что MIB-I разрабатывалась с ориентацией на работу с сетевыми устройствами, поддерживающим стек протоколов TCP/IP.

В MIB-II набор стандартных объектов был расширен до 185, а число групп до 10.

**Протокол SNMP**

Протокол SNMP – прикладной протокол стека протоколов TCP/IP, предназначенный для управления сетевыми устройствами и получения информации о них. Протокол SNMP стал почти общепринятым стандартом управления сетями и поддерживает много производителей сетевого оборудования. И для медийных устройств.

Протокол SNMP на транспортном уровне использует протокол UDP, который обеспечивает быстрый обмен данными между менеджером и агентом SNMP. По умолчанию SNMP агенты для получения запросов от SNMP менеджеров используют порт с номером 161. Кроме того, SNMP-менеджеры используют порт с номером 162 для получения от SNMP-агентов специальных уведомляющих сообщениях.

**Версии SNMP:**

1. SNMPv1 – RFC 1157 (в SNMP1 используется модель безопасности на основе сообществ (CSM), которая

предусматривает аутентификацию менеджера и агента SNMP **на основе имени сообщества**, где имя сообщества – это своеобразный пароль, передаваемые в SNMP-сообщениях **в открытом виде**; SNMP1 подверглась критике на свою модель безопасности, которая не способна бороться ни с одной из угроз безопасности, однако, несмотря на проблемы с безопасностью, версия SNMP довольно часто применяется в современных информационных сетях, так как является довольно простой);

1. SNMPv2p (в SNMP2p улучшила версию SNMP1 в сфере быстродействия и безопасности; данная версия протокола SNMP позволяет менеджерам получить от агента за один запрос информацию сразу о нескольких объектах;

модель SNMP2p поддерживает модель безопасности **на основе сторон (PSM)**, которая подразумевает, что менеджер и агент SNMP входят в виртуальное окружение исполнение, называемое стороной, для которого набор допустимых операций задан администратором сети;

в модели безопасности на основе сторон также используются протоколы аутентификации и шифрования;

из-за того, что модель безопасности на основе сторон многим показалась слишком сложной и запутанной, версия SNMP2p на сегодняшний день практически не используется);

1. SNMPv2c (SNMP2с отличается от SNMP2p только тем, что было решено отказаться от модели безопасности на основе сторон и вернуться к модели безопасности на основе сообществ; таким образом, проблемы с безопасностью остались нерешенными);Возможности предыдущей: перечисление объектов.
2. SNMPv2u (версия SNMP2u использует модель безопасности на основе имени пользователя (USM), которая подразумевает, что менеджер и агент действуют **от имени некоторого пользователя**; для аутентификации пользователей, каждый менеджер и агент имеют **общий секретный ключ** аутентификации, определяемый на этапе конфигурации системы управления сетью;   
   кроме того, в модели безопасности на основе имени пользователя подразумевается шифрование содержимого SNMP-сообщения и определение их своевременности с учетом показаний часов SNMP-агентов;

то есть, модель безопасности на основе имени пользователя во многом похожа на модель безопасности на основе сторон, но она значительно упрощает процесс идентификации пользователя);

1. SNMPv3 (версия SNMP3 поддерживает модель безопасности на **основе имени пользователя (USM),** и была более удачно доработана по отношению к предыдущим версиям).

Версии которые используются.

**Команды SNMP**

Менеджеры и агенты SNMP взаимодействуют между собой, используя специальный набор команд.

* GetRequest (используется менеджером для получения от агента значения какого-либо объекта базы MIB по его идентификатору);
* GetNextRequest (используется менеджером для получения от агента для получения следующего объекта в базе MIB без указания его идентификатора при последовательном просмотре объекта);
* GetResponse (использует агентом для передачи ответа менеджеру);
* SetRequest (используется менеджером для изменения значения какого-либо объекта в базе MIB; с помощью этой команды происходит управление сетевым устройством);
* Trap (используется агентом для отправки уведомляющего сообщения менеджеру о возникновении какого-либо события);
* GetBulkRequest (используется менеджером для получения от агента значений нескольких объектов базы MIB за один запрос; данная команда была введена в версии SNMPv2p);
* InformRequest (используется SNMP-менеджерами для обмена информацией из базы MIB между собой);
* SNMPv3-Trap (является улучшенной версией команды Trap).

**Структура модуля данных протокола**

Каждой из пяти команд соответствует свой тип PDU.

1 поле: Тип PDU – может принимать значения:

* AO (в 16-ричной системе счисления) для GetRequest;
* А1 (в 16-ричной системе счисления) для GetNextRequest;
* А2 (в 16-ричной системе счисления) для GetResponse;
* А3 (в 16-ричной системе счисления) для SetRequest.
* A4(в 16-ричной системе счисления) для Trap

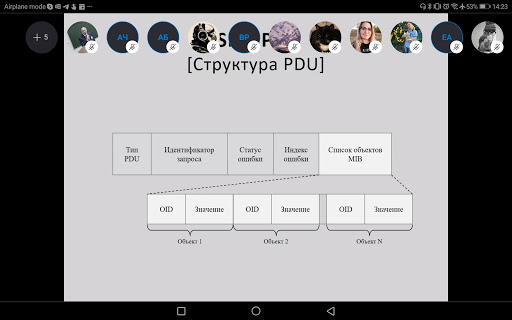
2 поле: идентификатор запроса – используется для согласования запросов и ответов;

3 поле: статус ошибки – указывает на наличие ошибки (в запросах должно быть равно 0, в ответах значения отличные от 0, означают следующие значения:

* 1 – слишком большой размер модуля данных протокола;
* 2 – не существует такого имени;
* 3 – неверное значение;
* 4 – только чтение;
* 5 – ошибка общего характера).

4 поле: индекс ошибки – уточняет тип возникшей ошибки, в случае, если в поле статус ошибки установлено значение, указывающее на наличие ошибки.

5-8 поле – это список: идентификатор объекта 1, значение объекта 1, идентификатор объекта 2, значение объекта 2.



**Формат SNMP-сообщений**

Рассмотрим формат SNMP-сообщений только для версии SNMP1, так как он является наиболее простым: каждое сообщение SNMP состоит из трех частей: версия протокола [0:SNMPv1], имя сообщества [default: public] и область данных.

В версии протокола указывается используемая версия SNMP (для SNMP1 это значение 0). Менеджер и агент должны использовать одну и ту же версию SNMP, в противном случае SNMP-сообщения должны передаваться с помощью прокси-агентов SNMP, которые обеспечивают взаимодействие агентов и менеджером, поддерживающих разные версии SNMP.

Имя сообщества задается сетевым администратором и представляет собой идентификатор группы устройств, управляемых определенным менеджером.

По умолчанию используется значение public. Агенты могут входит в состав только одной группы, но менеджеры могут быть включены в состав сразу нескольких групп, что позволяет им взаимодействовать с множеством объектов.

Область данных состоит из одного или многих модуля данных и протоколов PDU, в котором содержится идентификатор и значения определенных объектов базы MIB.

**Для команды Trap**

Он идет от менеджера

1 поле: тип PDU;

2 поле: тип объекта – указывает на каких объектах сформировал управляющее сообщение;

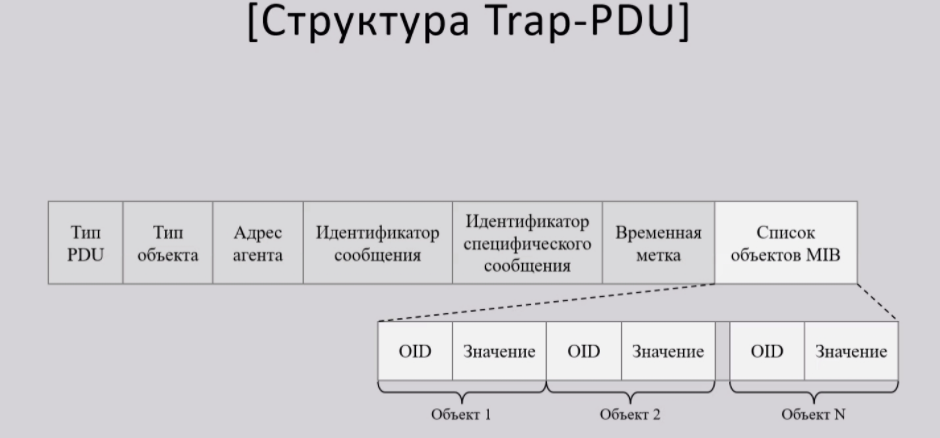
3 поле: адрес агента – содержит сетевой адрес агента, который является источником уведомляющего сообщения;

4 поле: идентификатор сообщения – указывает на причину, вызвавшего необходимость в отправке уведомляющего сообщения (если это значение равно 0, то происходит установление начального состояния объекта; если 1, то восстановление начального состояния объекта; если 2, то нарушение работы линии связи; если 3, то восстановление работы линии связи; если 4, то отказ в аутентификации запроса от менеджера; если 5, то сосед по внешнему шлюзовому протоколу не функционирует; если 6, то это специфическое сообщение);

5 поле: идентификатор специфического сообщения – указывает на уведомляющее сообщения, сформированное производителем сетевого устройства;

6 поле: временная метка – указывает на время формирования уведомляющего сообщения;

7-10 поле – это список: идентификатор объекта 1, значение объекта 1, идентификатор объекта 2, значение объекта 2.



**ASN.1 и BER**

SNMP- сообщения в отличии от многих других не имеют заголовок с фиксированными полями. Вместо этого, каждое поле сообщения претворяется его типов и размером, согласно формальному языку (ASN.1) и основным правилом кодирования данных (BER).

Протокол SNMP не совсем соответствует стандарту ASN.1, есть отличия в типах данных:

* INTEGER [02];
* STRING [04]
* NULL [05];
* OID [06];
* SEQUENCE [30] - последовательность;
* IPADDRESS [40];
* COUNTER [41];
* GAUGE [42] - датчик;
* TIMETICKS [43] – отмена времени.

SysDescr





**Безопасность в информационных сетях**

Большинство угроз безопасности в информационных сетях связаны с уязвимостью протоколов, в частности протоколов стека протоколов TCP/IP. Эти протоколы разрабатывались в то время, когда вопросы безопасности не стояли на повестке дня, так как тогда круг пользователей информационной сети был ограничен, как правило, это были специалисты, заинтересованные в работе сетей, и никто не покушался на работоспособность сети.

Однако, в процессе развития информационных сетей, возникла проблема обеспечения их безопасности. Даже появился специальный термин – сетевая безопасность.

Сетевая безопасность – комплекс мер, направленных на защиту информационных сетей от атак.

Атака – представляет собой попытку злоумышленника получить доступ к конфиденциальным данным, повредить или уничтожить данные, или каким-либо образом подорвать безопасность компьютера или сети.

**Классификация сетевых атак**

Классификация, принятая в области сетевой безопасности, включает в себя несколько классов сетевых атак:

1. Сетевая разведка и анализ сетевого трафика (как правило, атака притворяется сетевой разметкой, то есть сбором информации о компьютере жертвы с целью обнаружения уязвимости его безопасности; простейшая сетевая разведка может проводиться с помощью команды ping и специальной программой для сканирования портов; например, команда ping помогает определить какие ip-адреса присвоены компьютеру и какие из них доступны;

а сканирование портов выполняется с целью определить – какие известные службы и приложения запущены на атакуемом компьютере; кроме того, злоумышленник может использовать специальные программы и устройства, называемые sniffers и предназначенные для перехвата и анализа данных; снифферы могут применяться как для обнаружения уязвимости безопасности, так и для кражи идентификаторов пользователей и их паролей);

1. Вредоносные программы (термин вредоносные программы подразумевает три типа различных программ:

* компьютерный вирус – это вредоносная программа, которая внедряется в другие программы для выполнения определенных вредоносных действий;
* троянская программа – это вредоносная программа, которая выглядит, как полезная программа, но, на самом деле, выполняет вредоносные действия;
* сетевой червь – это вредоносная программа, самостоятельно распространяющаяся через информационную сеть.

В контексте сетевой безопасности вредоносные программы выполняют следующие действия:

* распространяют другие вредоносные программы;
* похищают данные, представляющие ценности или тайну, в том числе, данные для аутентификации;
* получают несанкционированный доступ к ресурсам компьютера, в том числе, осуществлять прямое управление компьютером;
* могут уничтожать данные, выводя из строя или приводя к отказу обслуживание компьютеров или сетей;
* шпионят за пользователем и передают третьим лицам сведения такие, как адреса электронной почты/часто посещаемые сайты.

1. Отказ в обслуживании (DoS) (имеет своей целью заставить атакуемый компьютер не отвечать на поступающие запросы, за счет превышения допустимых пределов его функционирования; данных класс атак не подразумевает получения доступа к атакуемому компьютеру или получения от него каких-либо данных; часто для приведения подобных атак злоумышленник использует сразу несколько компьютеров, как правило, без ведома их пользователей; в этом случае, имеет место **распределенная атака отказа в обслуживании (DDoS)**; в качестве примеров атак данного класса можно рассмотреть две атаки:

* SYN – флуд (атака заключается в отправке большого количества SYN-сегментов, каждый из которых созданию нового TCP-соединения; в результате проведения данной атаки, атакуемый компьютер, выделяя ресурсы для новых TCP-соединений, переполняет свою очередь соединений, при этом злоумышленник игнорирует TCP-сегменты с флагами ACK и SYN, не высылая ответные TCP-сегменты; таким образом, на атакуемом компьютере образуется множество полуоткрытых … из-за этого атакуемый компьютер не может отвечать на запросы попытки установить TCP-соединение);
* Smurf (заключается в отправке на широковещательный или групповой адрес большого количества ICMP-пакетов, при этом злоумышленник указывает ip-адрес компьютера жертвы в качестве ICMP-пакетов ; в результате проведения данной атаки, атакуемый компьютер может оказаться временно неспособным работать в сети, а также возможно нарушение функционирования самой сети из-за возросшего сетевого трафика).

**Подмена**

Представляет собой процесс фальсификации передаваемых данных и применяется злоумышленником с целью маскировки под кого-то другого. В большинстве случаев, информационные сети уязвимы к данному классу атак.

**1. Подмена МАС-адреса**

Подмена МАС-адреса на сетевом адаптере может **заставить коммутатор отправлять кадры на порт, к которому подключен злоумышленник**, независимо от того, что эти кадры могут предназначаться другому получателю. Кроме того, используя подмену МАС-адреса можно ввести в заблуждение систему безопасности, осуществляющую контроль доступа, на основе некоторого списка МАС-адресов.

Такая система безопасности становится бесполезной в том случае, если в кадре исходный МАС-адрес будет заменен на МАС-адрес из списка разрешенных.

**2. Подмена IP-адреса**

Используя подмену IP-адреса отправителя в заголовке IP-пакета, злоумышленник может выдать себя за кого-то другого. Как правило, данная атака осуществляется для того, чтобы замести следы несанкционированного доступа, чтобы они привели к кому-то другому.

Также можно использовать подмену IP-адресов, как и в случае с подменой МАК-адресов для того, чтобы обойти список контроля доступа и обмануть систему безопасности.

**3. Подмена записей ARP**

Заключается в том, что злоумышленник отправляет в локальную сеть ARP-пакеты с целью ассоциировать МАС-адрес своего компьютера с IP-адресом компьютера, которому доверяет жертва. В результате проведения данной атаки, данные, предназначенные легитимному читателю, теперь будут направляться злоумышленнику. При этом злоумышленник может скрыть факт атаки, оправляя все принятые данные на IP-адрес реального получателя. И поскольку потери данных не происходит, практически невозможно определить присутствие злоумышленника.

**4. Отравление кэша DNS**

позволяет воздействовать на процесс определения IP-адресов по доменному имени. Данная атака осуществима двумя способами:

1. Злоумышленник устанавливает на компьютер жертвы вредоносную программу, которая будет модифицировать кэш DNS-клиента таки образом, чтобы жертва получала некорректные IP-адреса.
2. Злоумышленник атакует DNS-сервер и модифицирует его кэш. Таким образом, все, кто использует этот сервер, будут получать некорректные IP-адреса. В очень редких случаях злоумышленники могут получить доступ к корневому DNS-серверу, который хранит записи доменов верхнего уровня, и модифицировать записи на этом сервере. При этом другие DNS-серверы получат измененные данные автоматически, что может привести к отказам глобальных сетевых служб и сайтов.

**Повторное воспроизведение**

Атака повторного воспроизведения заключается в том, что злоумышленник перехватывает данные, передаваемые между двумя компьютерами, а затем воспроизводит эти данные для одного из них. В результате проведения данной атаки, атакуемый компьютер может обработать повторно воспроизведенные данные, как допустимые, что в свою очередь может привести к негативным последствиям. Наиболее уязвимыми к атакам повторного воспроизведения являются беспроводные сети, так как злоумышленник может относительно легко получить доступ к каналу связи, находясь недалеко от передатчика электромагнитных волн.

**Спам и фишинг**

Спам подразумевает отправку огромного количества электронных писем одному или нескольким пользователям, при этом электронные письма могут содержать как безобидную и рекламную информацию, так и различные вредоносные программы. В результате проведения данной атаки работа с электронной почтой становится практически невозможно. Кроме того, спам может использоваться с целью проведения атаки, называемой **фишин**г.

Основным отличием **фишинг**а от других атак является то, что в **роли объекта атаки выбирается не компьютер, а его пользователь**. Целью данной атаки является получение от пользователя личных данных, которые могут быть использованы для получения доступа к конфиденциальной информации.

Для проведения фишинг-атаки, злоумышленник организует рассылку спама от имени банков, операторов сотовой связи и т.п. Обычно в таких письмах содержится уведомление о том, что по какой-либо причине получателю срочно нужно передать или обновить личные данные, и прилагается ссылка на сайт, где он это может сделать. По этой ссылке пользователь попадает на поддельный сайт, который внешне неотличим от оригинального сайта, где у него «выуживаются» его личные данные. При этом ссылка может содержать адрес очень похожий на адрес оригинального сайта так, чтобы пользователь не мог заподозрить обмана.

Зачастую успех фишинга объясняется незнанием пользователями простого факта того, что легальные организации не рассылают писем с просьбами сообщить свои личные данные.

**Защита от сетевых атак**

Защита заключается в реализации разнообразных функций сетевой безопасности, среди которых можно выделить следующие:

1. **Аутентификация источника данных** – проверка и подтверждение подлинности источника полученных данных, то есть того, что полученные данные были созданы именно указанным источником.
2. **Аутентификация сторон** – это проверка и подтверждение для одной из сторон подлинности другой стороны, то есть того, что на другой стороне действительно тот, за кого он себя выдает.
3. **Конфиденциальность данных** – сокрытие передаваемых данных от тех, кто не является участником обмена данными.
4. **Целостность данных** – проверка того, что защищаемые данные не подвергались несанкционированному изменению, то есть не были искажены или подделаны.
5. **Разграничение доступа** – обеспечение невозможности несанкционированного доступа к ресурсам компьютера.

Для реализации функций сетевой безопасности могут применяться как программные, так и аппаратные средства.

**Протокол сетевой безопасности**

Обычные сетевые протоколы никак не обеспечивают безопасность своих данных, что делает их уязвимыми к различным сетевым атакам. Протокол, обеспечивающий поддержку хотя бы одной функции сетевой безопасности называется протоколом сетевой безопасности.

Среди протоколов сетевой безопасности выделяют **криптографические** протоколы, которые обеспечивают безопасность с помощью криптографических средств защиты.

**Протоколы SSL (secure sockets layer) и TLS(transport layer security)** являются криптографическими протоколами, которые обеспечивают аутентификацию сторон, целостность и конфиденциальность данных. В контексте эталонной модели OSI протоколы SSL и TLS работают на границе между транспортным и сеансовом уровне, что позволяет сторонам договариваться о совместных сеансовых ключах, которые используются для шифрования передаваемых данных.

В большинстве прикладных протоколов TCP/IP для защиты данных может использоваться как протокол SSL, так и TLS. Как правило, защищенным версиям таких протоколов соответствуют другие номера портов FTP (990), HTTPS (HTTP Secure 443), SMTP (465), POP3 (995), IMAP (993). RFC 2246, 4346, 5246, 6176, 7568.

**Протокол SSH** – это прикладной протокол стека протоколов TCP/IP, позволяющий удаленное управление компьютером в терминальном режиме. Протокол SSH схож с протоколом TELNET, но в отличие от него обе**спечивает аутентификацию сторон и безопасность данных**. Кроме того, протокол SSH может использоваться для установки защищенного соединения между двумя узлами. Передача данных по протоколу SSH осуществляется с использованием протокола TCP. По умолчанию SSH-сервер использует порт с номером 22. RFC 4251.

**Протоколы IPSec** – представляют **собой набор протоколов для защиты данных**, передаваемых по протоколу IP.

* Протокол AH – это **протокол**, обеспечивающий **целостность** данных, **аутентификацию** источника и **защиту от повторной передачи** IP-пакета.
* Протокол ESP – это криптографический протокол, обеспечивающий конфиденциальность данных. Кроме того данный протокол может выполнять функции протокола AH.
* ISAKMP – это протокол, обеспечивающий обмен сеансовыми ключами, предназначенных для шифрования данных.

RFC 2401 – 2412

//из этих перехваченных данных он ничего не сможет получить не имя ключей.

//не получится ввести в заблуждение

**Межсетевые экраны**

Это программное или аппаратное средство, осуществляющее контроль сетевых соединений.

Межсетевой экран анализирует входящие и исходящие данные и принимает решение, разрешить или блокировать соединение, по которому эти данные передаются.

Программный межсетевой экран (брандмауэр) представляет собой сетевое приложение, работающее либо на компьютере, либо на каком-то граничном сетевом устройстве, например, маршрутизатор.

Аппаратный межсетевой экран представляет собой специальное сетевое устройство.

Большинство современных межсетевых экранов анализируют данные по следующим критериям:

1. По адресу – входит ли адрес физический или сетевой получателя или источника данных в заданный список контроля адресов, или же не входит.
2. По доменному имени – входит ли доменное имя получателя или источника или получателя в заданный список контроля доменов, или же не входит.
3. По номеру порта – входит ли порт, через которые были переданы данные в заданный список контроля портов, или же не входит.
4. По протоколу – входит ли протокол данных в заданный список контроля протоколов, или же не входит.

Также межсетевой экран может блокировать все соединения, которые соответствуют описаниям известных сетевых атак.

**Прокси-серверы**

Это программное средство, выполняющее **функции посредника между клиентом и сервером**.

Первым делом клиент подключается к прокси-серверу и передает какие-либо данные к серверу.

Затем прокси-сервер подключается к указанному серверу и передает ему данные от клиента.

Подобно межсетевому экрану, прокси-сервер может контролировать обмен данными между клиентом и севером.

Кроме того, в некоторых случаях, передаваемые данные могут быть изменены прокси-сервером в определенных целях. Например, чтобы скрыть информацию о клиенте.

**Чаще всего серверы применяются для организации доступа из локальной сети в сеть интернет**.

//NAT не может контролировать передаваемые данные. А Прокси-сервер может.

//запрос на реальный IP-адрес только после подключения к Прокси-серверу.

**Виртуальные частные сети VPN**

Представляет собой изолированное соединение типа **точка-точк**а в общедоступной сети, например сети Интернет. При этом реальная инфраструктура общедоступной сети не имеет значения, поскольку данные передаются подобно тому, как если бы они передавались по выделенному каналу.

Существует два типа виртуальных частных сетей:

1. VPN-**соединение удаленного доступа** предоставляет компьютеру доступ к серверу локальной сети с помощью **инфраструктуры общедоступной сети**.
2. VPN-соединение типа **сеть-сеть** связывает **две локальные сети с помощью инфраструктуры общедоступной сети**. При этом обеспечивая безопасность связи. Для изолирования передаваемых данных VPN-соединение использует технологию туннелирование,

которая заключается в нестандартном (отличном от порядков эталонной модели OSI) способе инкапсуляции пакета одного протокола в пакеты другого. При этом протоколы более низкого уровня могут инкапсулироваться в пакеты

Туннелирование само по себе не защищает данные от несанкционированного доступа или искажение, но позволяет защитить инкапсулируемые пакеты. Для этого отправитель шифрует исходный пакет, а затем помещает зашифрованный пакет в другой внешний пакет. При этом внешний пакет имеет открытый заголовок, а исходный пакет шифруется вместе с заголовком, так как некоторые поля заголовков содержат информацию, которая может быть использована злоумышленником.

Для инкапсуляции по прибытии получатель извлекает из внешнего пакета исходный пакет, расшифровывает его использует его заголовок для дальнейшей передачи в локальной сети.

В туннелировании используются :

* Протокол-пассажир – протокол, предназначенный для транспортировки данных в локальных сетях
* Несущий протокол – предназначенный для транспортировки данных в общедоступной сети.
* Протокол-туннелирования – протокол, обеспечивающий инкапсуляцию пакета одного протокола в пакеты другого

Распространенным вариантом протокола-пассажира и несущего протокола является протокол IP.

В качестве туннелирования используются протоколы PPTP, L2tp/IPSec, SSTP

**Протокол PPP**

Это протокол канального уровня, предназначенный для отправки данных по соединению типа точка-точка. При этом протокол РРР может обеспечивать аутентификацию сторон, конфиденциальность и сжатие данных. РРР позволяет работать с различными сетевыми протоколами. В случах протоколов IP пакеты инкапуслируются в PPP кадров которые затем передаются по соединению типа точка-точка.

Для аутентификации сторон в протоколах PPP могут быть использованы следующие протоколы:

* PAP (password authentication protocol) – это протокол проверки подлинности, предусматривающий передачу имени пользователя и пароля без шифрования в открытом виде.
* CHAP – это протокол проверки подлинности, предусматривающий следующий алгоритм действий:   
  1) при установлении соединения одна сторона (сервер) передает сеансовый ключ, состоящий из случайных последовательных символов  
  2) другая сторона (клиент) с помощью алгоритма MD5 хэширует полученный сеансовый ключ и пароль, после чего передает их северу вместе с именем пользователя.  
  3) сервер, зная пароль указанного пользователя и высланный ранее сеансовый ключ, в свою очередь проделывает с ними те же действия, что и клиент. Если результаты совпадут, аутентификация считается успешной
* MS-CHAP – протокол проверки подлинности, являющийся расширение протокола CHAP и предусматривающий взаимную аутентификацию сторон, при которой обе стороны (клиент и сервер) должны доказать свою подлинность. Если обе стороны смогли доказать подлинность, аутентификация считается успешной.
* EAP-TLS – протокол проверки подлинности, использующий сертификаты открытых ключей и криптографический протокол TLS. Использование сертификатов открытых ключей предполагает наличие третьей стороны, которой доверяют обе стороны, и которая сможет подтвердить их подлинность. Если обе стороны смогли доказать подлинность, аутентификация считается успешной.

//Сертификат

// MPPE

**Протокол PPTP**

Тунельный протокол позволяющий создавать защищенное VPNсоединение. Протокол PPTP является расширением протокола PPP и использует протоколы проверки подлинности, шифрования и сжатия данных.

Протокол инкапсулирует PPP-кадры с помощью протокола GRE (Generic Routing ).(IP-заголовок, GRE-заголовок,PPP-заголовок,IP-пакет (отправитель и получатель / клиент и сервер) зашифрованный). Кадр PPP содержащий зашифрованный IP –пакет инкапсулируется вместе с заголовком GRE и IP. В IP-заголовке содержатся адреса отправителя и получателя, соответствующие VPN-клиенту и VPN-серверу.

**Протокол L2TP/IPSec**

L2TP-тунельный протокол позволяющий устанавливать защищенное VPN-соединение. Представляет собой комбинацию протоколов PPTP L2F. В отличии от PPTP не используется шифрование PPP-кадров. Для этого используется средство шифрования представленное IPSec.

**Инкапсуляция.**  С начала PPP-кадр содержащий зашифрованный IP-пакет инкапсулируется вместе с заголовком L2TP/UDP. L2TP далее шифруется. Затем полученному L2TP сообщению добавляются заголовки и замыкатель протокола ESP IPSec, замыкатель проверки подлинности IPSec, который обеспечивает целостность и аутентификацию источника данных а так же окончательный IPзаголовок.

**Протокол SSTP**

Туннельный протокол позволяющий устанавливать защищенное VPNсоединение. Протокол SSTP для передачи данных использует протокол HTTPS, который обеспечивает конфиденциальность данных средствами криптографического протокола SSL/TLS. Кроме того протокол HTTPS позволяет протоколу SSTP передавать данные через межсетевые экраны и прокси-серверы, которые могут блокировать данные протоколов PPTP L2TP/IPSec.

**Системы обнаружения и предотвращения вторжений**

**Система обнаружения вторжений-** комплекс программных и аппаратных средств, предназначенных для выявления сетевых атак. Система обнаружения вторжений делится на две категории

* Сетевые системы обнаружения вторжений-которые выявляют сетевые атаки анализируя данные своих агентов расположенных в разных узлах сети.
* Система обнаружения атак на уровне узла- которые выявляют сетевые атаки на компьютер анализируя данные которыми он обменивался.

**Система предотвращений вторжений**

Комплекс программных и аппаратных средств предназначенных не только для выявления сетевых атак, но и для защиты от них. Такие системы способны блокировать подозрительные программы, а также блокировать или разрывать сетевые соединения, по которым выполняется атака. Наибольшее распространение получила такая система-система предотвращения вторжений на уровне узла.

Кроме того различные элементы такой системы активно используется в антивирусных программах в борьбе с сетевыми атаками и вредоносными программами.