# 

# Базовый курс по компьютерным сетям

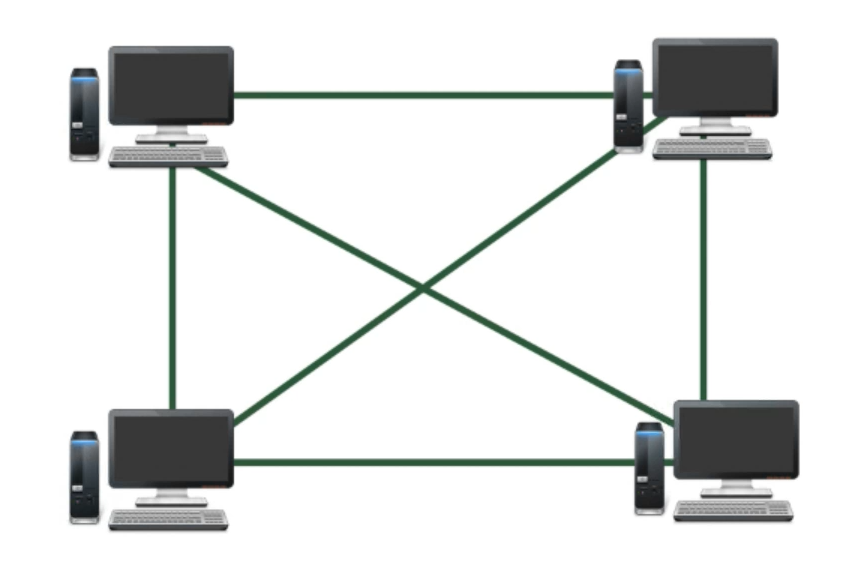
# Простая сеть на коммутаторе

**Как было раньше (топология общей шины):**



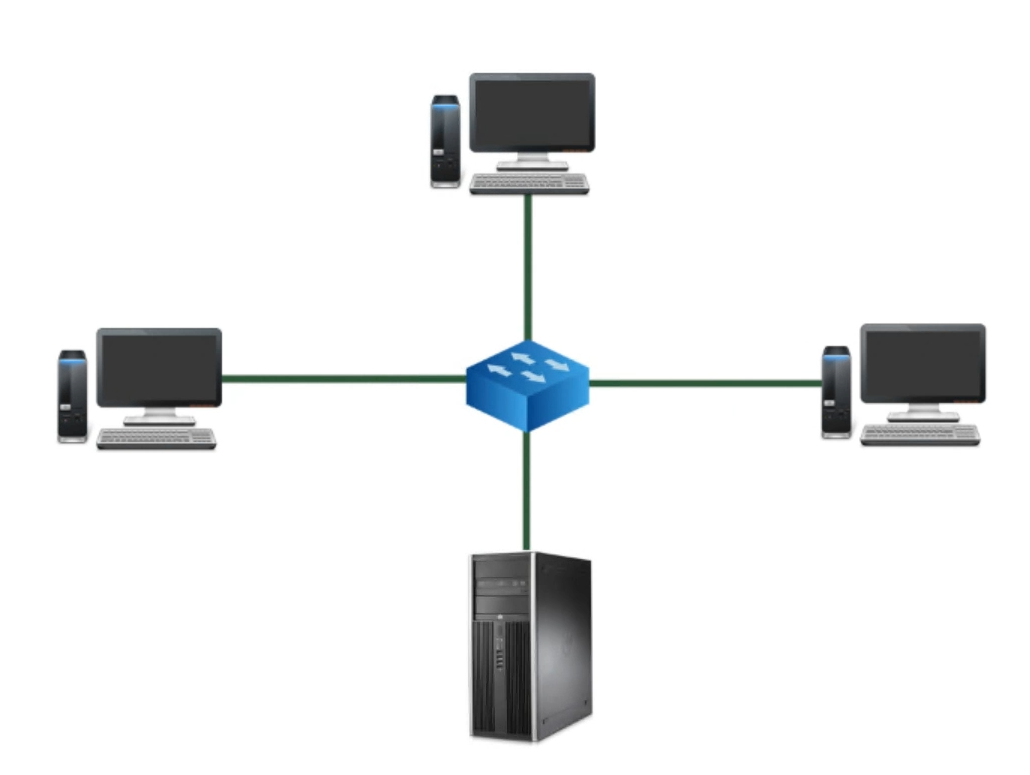
Пока 1 из пк отправляет трафик, другие не могут отправлять трафик в сеть, могут конечно, но тогда возникнет коллизия (ошибка).

**Топология каждый на каждого (Full-Mesh):**



Нужна куча сетевых карт, не вариант.

**Топология звезда:**



Данные попадают на коммутатор (свич) и ложится в его буфер, после чего ждут освобождения канала, как только канал освобождается, данные поступают.

Данные будем называть кадрами, кадр - единица информации которая передается по сети в пределах одной канальной среды. Кадр передается на MAC адрес, который мы узнаем с помощью ARP таблиц и IP адреса. MAC зашит в сетевое оборудование устройства.

**Структура МАС адреса**



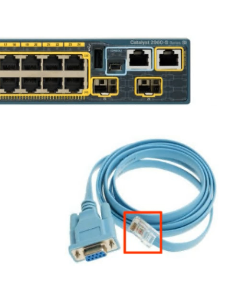
* Unicast - конкретному ПК
* Broadcast - всем в сети
* Multicast - группе ПК

Комутатор не умеет работать с IP, он работает с MAC адресами. Коммутатор запоминает за каким портом (на специальном очень быстром чипе), какое устройство находится (по МАС) чтобы помещать в буфер этого порта кадры. Это называется “МАС таблица коммутатора” (CAM таблица в CISCO-теории).

Если у нас комутатор 1 связан с коммутатором 2, то за портом коммутатора 1 может закрепляться группа МАС адресов, которые связаны с коммутатором 2.

Размер таблицы от 1 тысячи до 8, в зависимости от стоимости. Если всё переполнено, рассылает на все порты (Unicast Flooding).

Управление свичём происходит через консольный порт. В CPT он эмулируется во вкладке “CLI” - это Cisco iOS :)



Напишите знак ? чтобы увидеть все команды.

Когда вы видите “>” вы в простом режиме.

Войдите в привилегированный режим, введя en (enable), вы попадете в “#”.

Чтобы войти в конфигурацию свича, наберите conf ter.

Нажмите на tab чтобы команда дописалась автоматически по сокращенному названию, как с “en”.

*Команда для свича в CTP “****# show mac-address-table****” - показать таблицу*

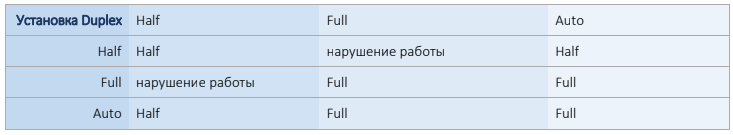
**Резюме. Коммутатор:**

1. Соединяет ПК между собой;
2. Предотвращает появление коллизий;
3. Направляет трафик только в нужный порт (через таблицу MAC), снижая нагрузку на сеть.

## Режимы работы

**Полудуплекс (Half-duplex)** - данные или принимаются или отправляются в один промежуток времени.

**Полный дуплекс (Full-duplex)** - данные и принимаются и отправляются. Т.е. в витой паре по одним жилам они передаются, по другим принимаются. В 100мб проводе 2 на прием 2 на отправку, в гигабитном 4 прием, 4 отправка.

Свич и ПК договариваются о типе соединения и скорости, смотря кто что поддерживает, это называется **автосогласование**.

**Внимание!** Если на вашем устройстве не включено автосогласование, то свич выставит самые слабые параметры работы (half и 10 мбит/c). Т.к. он считает что ваше оборудование очень старое. Согласуйте данный интерфейс вручную!

При **Full-duplex** - коллизий быть не может.

**Half-duplex** может понадобится если у нас плохой кабель или на него идут наводки (в общем сеть глючит (теряются пакеты, большой пинг)). Если при переводе на **Half-duplex** всё становится ок, значит дело в кабеле (может не так обжат) или в наводках на него.

**Внимание!** При перезагрузке свича, ваши настройки не сохранятся.

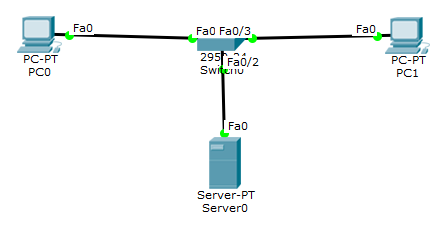
# 2. Простая сеть на маршрутизаторах

Существует протокол [ARP](https://ru.wikipedia.org/wiki/ARP) с помощью которого создаётся соответствие между IP и MAC адресом.

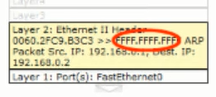
Как это работает:

1. Отправитель спрашивает в сети, есть кто с IP, например 192.168.0.1
2. Если такой абонент есть, он отвечает, это я и отправляет свой MAC
3. Отправитель записывает его MAC и IP в свою таблицу ARP

*Чтобы увидеть ARP таблицу на вашем ПК, отправьте команду arp -A*



1. Отправим ping с PC0 на PC1
2. Сформируется два пакета ICMP (ваш ping) и ARP сообщение;
3. Просмотрев ARP вы увидите



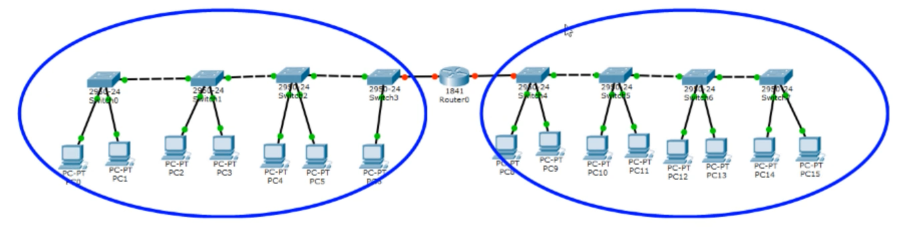
1. Вы увидите тело запроса, мы видим что он широковещательный (FFFF.FFFF.FFFF). Т.е. Его получают все участники сети, но отвечает только тот, у которого указанный нами IP.
2. После того как ARP таблица на вашем ПК будет заполнена, отправляются основные данные, в нашем случае это ICMP пакет (команда ping).

**Сложная сеть на свичах**

Всё это - одна большая **канальная (широковещательная) среда**, это значит что кадр от PC0 пройдёт по свичам до PC15.

Если мы отправим ping от PC0 до PC15 и посмотритм как передаются **ARP сообщения**. Все компьютеры в сети получают **ARP сообщение**, это называется **широковещательный шторм.**

Хорошо это или плохо? А если таких PC тысяча? Нагрузка на сеть будет большой, чтобы этого не происходило надо использовать **роутеры**.

Модифицировали сеть, установив роутер, мы получаем две канальные среды, теперь PC0 не сможет достать широковещательным запросом PC15.

Роутер - перекладывает пакеты из одной канальной среды в другую, это называется **маршрутизация**.

**Отличия коммутатора от маршрутизатора:**

У коммутатор все настройки включены по умолчанию, у маршрутизатора всё выключено, нет MAC таблицы, он **не работает с кадрами**, **работает с пакетами**, поэтому на **каждый интерфейс роутера** должен быть **назначен IP адрес**.

Если мы отправим команду ping c PC0 на PC15. Операция не выполнится, т.к. не указан **основной шлюз** (устройство которое перекладывает из одной канальной среды в другую).

Нужно задачть Gateway на PC0 и PC15, им будут IP интерфейсы роутера. Теперь при отправке команды ping. **Первый пакет потеряется**, потому что время было потрачено на ARP, выяснилось соотношение IP и MAC.

В канальной среде для адресации используется MAC адрес. Когда мы хотим отправить сообщение в другую канальную среду, происходит следующее: PC0 видит что ему нужно отправить сообщение на IP из другой подсети, значит это другая канальная среда и ему нужно отправлять сообщение не по MAC, а через **Gateway (основной шлюз)**, который уже пошлет пакет на PC15.

**Алгоритм работы:**

1. Отправляем ping с PC0 на PC15. **PC0** сразу **понимает** что **PC15 из другой подсети**, и надо отправить сообщение **на Gateway**, для этого он отправляет **ARP-запрос**, чтобы **узнать какой MAC** у основного шлюза 192.168.0.1.
2. Роутер ему ответит, прислав MAC.
3. Далее PC0 отправит пакет на коммутатор , в котором прописан MAC роутера но IP у PC15.
4. Роутер увидит что данные надо отправить на IP из другой подсети, но он не знает MAC адреса PC15, по этому первый пакет откинется (потеряется).
5. Далее роутер отправляет ARP-запрос, с вопросом у кого IP 172.20.20.25 и какой у вас MAC.
6. PC15 ответит, роутер получит MAC, формирует ARP-таблицу и в следующий раз роутер уже отправит сообщение.

**Резюме**

* Роутеры используются чтобы ограничить размер широковещательного домена (канальной среды), и снизить загрузку сети;
* Назначением роутера является перемещение IP пакетов из одной канальной среды в другую;
* Если узел отправителя не может достать до узла получателя по канальному MAC адресу, то узел отправителя использует услуги основного шлюза (роутера).

# 3. IP адресация

Internet Protocol (IP) адреса бывают **частные** и **публичные**.

**3 диапазона частных адресов:**

* 192.168.0.0/16 (то есть со 192.168.0.0 по 192.168.255.255 включительно);
* 172.16.0.0/12 (то есть с 172.16.0.0 по 172.31.255.255 включительно);
* 10.0.0.0/8 (то есть всё что начинается на 10).

*Диапазоны могут быть другими, но эти чаще всего используются по дефолту.*

Они работают только в пределах нашей организации (нашей локальной сети), они не маршрутизируется в Интернет. Можем назначить любой IP адрес, какой хотим.

**Примеры публичных адресов:**

* 178.217.16.246
* 217.69.139.200
* 78.110.50.125
* 81.192.76.172

Маршрутизируется в интернете, назначаются публичным веб серверам, организациям. Выдача IP адресов контролируется организациями.

**Как получить выделенный (статический) публичный IP адрес:**

Вы -> Интернет провайдер -> LIR -> RIR -> IANA -> ICANN

LIR - локальный интернет регистратор, выдает пачку адресов провайдеру.

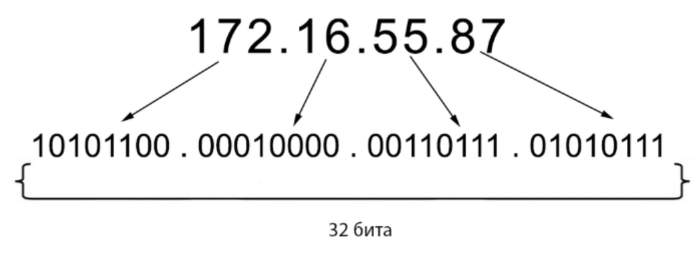
RIR - региональный интернет регистратор

IANA - международная некоммерческая организация (контролирует IP и Доменные имена). Выдаёт пачками IP региональным провайдерам.

ICANN - контролирует работу IANA.

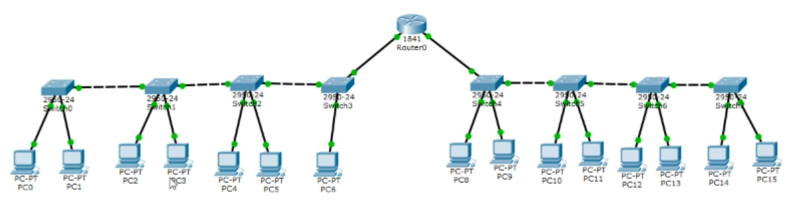
**Кейс.** Ваша организация переехала в другой город, там другой провайдер и другие IP. Вам придётся менять IP на всех серверах, иногда это очень проблемно, когда много сервисов завязаны на статических адресах. Для решения этой проблемы, вы можете пойти сразу к LIR и получить **провайдера независимый IP адрес (Provider Independent).**

**Структура IP-адреса v4**



* 172.16.55.87 - то как видим IP мы (десятичная система исчисления),
* 1010110.00010000.0011.0111.01010111 - то как видит его ПК (двоичная система).
* IP состоит из 4х октетов (числа до точки), в данном случае это 172, 16, 55, 87.
* Каждый октет = 8 бит или 1 байт.

**Пример**



Отправим ping с PC0 на PC1.

*При отправке PC0 нужно будет понять, где находится PC1, в той же канальной сети или в другой, использовать услуги шлюза или нет. Для того чтобы это определить используется маска подсети.* ***Она нам даёт ответ какая часть IP относится к адресу хоста, а какая к адресу сети.***

*Маска говорит нам о том, какая часть адресного пространства (ip) используется для адресации сетей, а какая для адресации хостов.*

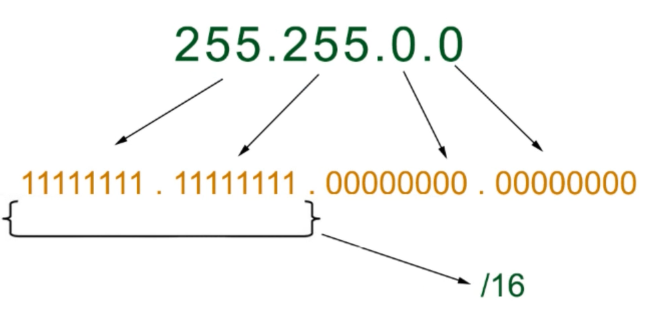
**Переменная длина маски подсети**

**Variable Length Subnet Mask (VLSM)**

****

* В первом случае, к адресу сети относится 3 октета по 8 = 24. (маска /24 или 255.255.255.0), **адрес сети 172.16.55.0**
* Во втором случае, 2 октета по 8 = 16. (маска /16 или 255.255.0.0), **адрес сети 172.16.0.0**

**Маска подсети**



Чтобы PC0 понять в одной и сети с ним находится PC1 или нет, по маске определяем сколько октетов сравнивать, и производим сравнение октетов “адреса сети” в IP адресе.

**Широковещательный пакет**

Отправляется всем участникам сети.

**Пример №1, у нас есть ПК с 172.16.55.87 (255.255.255.0 /24)**

172.16.55.0 - адрес сети;

172.16.55.1 - адрес первого хоста;

172.16.55.254 - адрес последнего хоста;

**172.16.55.255 - широковещательный IP адрес (адрес широковещательной рассылки);**

172.16.56.0 - адрес следующей подсети;

Итог: 254 хостов в сети

**Пример №2, у нас есть ПК с 172.16.55.87 (255.255.0.0 /16)**

172.16.0.0 - адрес сети;

172.16.0.1 - адрес первого хоста;

172.16.255.254 - адрес последнего хоста;

**172.16.255.255 - широковещательный IP адрес**

172.17.0.0 - адрес следующей подсети;

Итог: 65534 хостов в сети

**Делим адреса на сетевые и хостовые внутри октета**

**Пример №3, у нас есть ПК с 172.16.55.87 (255.255.255.192 /26)**

Разложим **255.255.255.192** в двоичный вид, получим:

11111111.11111111.11111111.11000000

**Внимание! Мы не можем поставить любое число вместо 192, его необходимо выбрать таким образом, чтобы в двоичном виде это выглядело как фиксированная последовательности единиц и нулей (пример 11110000 или 11111000), не должно быть чего то вида 11001100.**

Как получить адрес сети:

У нас маска заканчивается на 192, это 11000000 в двоичном виде. У нас заняты первые два бита, найдём адрес подставляя нули, значит нашим адресом сети будет 01000000, в десятичном виде это 64. Широковещательный адрес находим подставляя единицы, 01111111 это 127.

172.16.255.64 - адрес сети;

172.16.255.65 - адрес первого хоста;

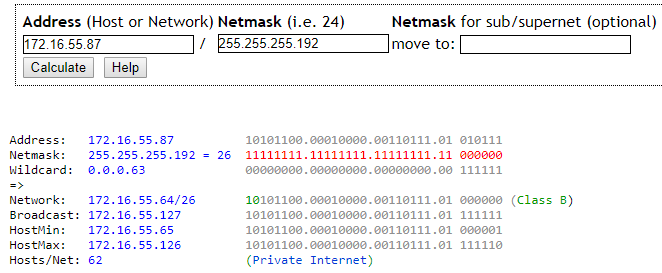
172.16.255.126 - адрес последнего хоста;

172.16.255.127 - широковещательный IP адрес;

172.16.255.128 - адрес следующей подсети;

Итог: 62 хоста в сети.

Для расчетов используйте онлайн калькулятор, например [jodies.de/ipcalc](http://jodies.de/ipcalc)



Wildcard - перевернутая маска, пойдём на шаге про ACL.

**Сокращение IP адресов**

К адресу 192.168.0.1 можно обращаться как к 192.168.1, а к 172.0.0.1, как 172.1. Однако многие программы могут это не поддерживать и люди могут быть не знакомы с эим.

**Классовые сети**

Использовались до создания масок подсети (**Variable Length Subnet Mask**):

****

Необходимо было фиксировать первые биты, как на примере. Сейчас от этого отказались и придумали маски подсети.

**Пример:**

Класс A, ip начинались с 1 до 126. Количество хостов составляло 16.777.214

Класс В, ip начинались со 128 до 191. Количество хостов составляло 65.535

Класс С, ip начинались со 192 до 223. Количество хостов составляло 254

**Сейчас классовые сети не используются. Только маска подсети.**

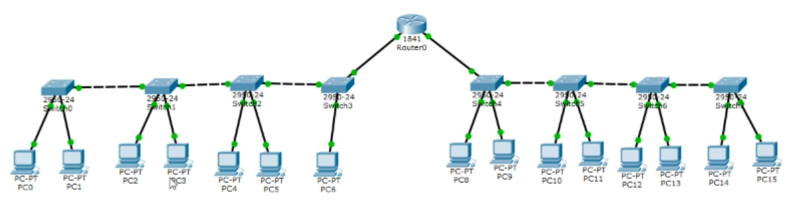
**Зарезервированные IP адреса**

*Данные адреса невозможно использовать, ограничения наложены на уровне ПО.*

* **Адреса 127.0.0.0 - 127.255.255.255 (loopback)** *- адреса обратной петли (используются для диагностики)*
* **Адрес 0.0.0.0 и любые другие, начинающиеся на 0, например 0.х.х.х *-*** *используются для отправки пакетов, в случае если IP еще не назначен (Пример - включили роутер, и ему надо обратиться к другому узлу (например к DHCP), он не знает к кому и в поле источник пишет 0.0.0.0).*
* **Адреса 169.254.0.0 -169.254.255.255 - Automatic Private IP Addressing (APIPA)** - *штука которая придумывает нам IP адрес, если DHCP не сработал, и ручками мы ничего не прописали. Она автоматически генерирует IP адреса для вас.*
* **Широковещательный адрес 255.255.255.255** *- когда узел рассылки не знает какой у него IP и какая сеть.*
* **Мультикаст-адреса от 224.0.0.0 до 247.255.255.255** - адреса для отправки сообщений всем роутерам, свичам, DHCP или просто группе ПК.
* **Адреса, начинающиеся в самом начале от 248 до 255, например 248-255.х.х.х** - зарезервированы для тестирования
* **Адреса, заканчивающиеся на 0, например х.х.х.0 или х.х.0.0** - адреса сети
* **Адреса, заканчивающиеся на 255, например х.х.х.255 или х.х.255.255** - широковещательные адреса.

**Proxy ARP**

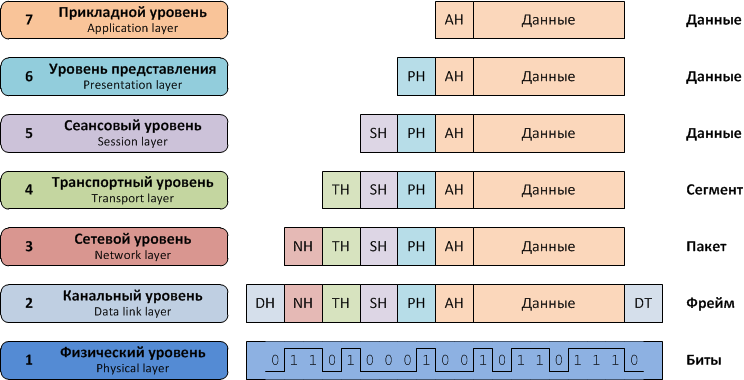
Если по какой то причине был отправлен пакет с неправильной маской сети, то данное сообщение Роутер обработать не сможет. Для того чтобы устранить эту проблему существует настройка **Proxy ARP**.

1. Отправим пинг с неправильной маской, с PC0 на PC15, с маской 255.255.0.0.
2. Мы увидим как ARP пакеты пойдут по канальной сети (до роутера), PC0 не получит ответа, т.к. мы ему сказали что у нас адрес сетьи 192.168.0.0, а не 192.168.1.0 и он подумал что IP компьютера PC15 находится в его канальной сети.
3. С помощью технологии **Proxy ARP** мы имитируем работу в одной канальной среде, для PC из разных сред.
4. Заходим на Роутер и включаем технологию **Proxy ARP**, для каждого интерфейса.
5. Повторим запрос с неправильной маской (255.255.0.0). Происходят опять ARP-запросы.
6. Роутер понимает что запрос не для него, но с включенным **Proxy ARP** он видит сеть назначения, и отвечает нашему PC0 своим MAC.
7. PC0 думает что всё впорядке, и отправляет ему сообщение. И у нас всё работает.
8. В итоге мы будем видеть в ARP-таблице на PC0, MAC адрес роутера вместе с IP PC15, тоже самое в ARP-таблице на PC15.

# 4. Модель OSI

Мы хотим передать информацию, чтобы это сделать, компьютер превращает информацию в последовательность бит (последовательность 0 и 1). Чтобы это сделать, мы воспользуемся моделью OSI (правила преобразования данных для передачи по сети).

**Сетевая модель OSI** (open systems interconnection basic reference model — Базовая Эталонная Модель Взаимодействия Открытых Систем)



**Физический (L1 уровень)** - решает вопрос, как передать отдельные биты, определяет характеристики сигнала. Какого вида и какой формы должен быть сигнал (пример. Если это Ethernet кабель - какая модуляция будет, какое напряжение, ток. У wi-fi какие волны будут, частота, модуляция).

**Устройства на этом уровне:** Сетевая карта, модем, wi-fi антенна.

**Протоколы:** -

**Канальный уровень** **(L2)**  - как передать осмысленные последовательности из бит. Для передачи в одной канальной среде, появляются **MAC адреса**.

**Устройства на этом уровне:** Сетевой коммутатор (свитч), сетевая карта (работает с MAC).

**Протоколы:** Ethernet (осуществляет доставку данных в одной канальной среде)

**Сетевой уровень (L3)** - передача данных между канальными средами. Появляется понятие **IP адреса**.

**Протоколы:** IP

**Транспортный уровень (L4)** - определяет для какого приложения предназначается отдельный кусок информации. Определяет какую последовательность бит какому приложению передать. Появляется понятие **порта**.

**Протоколы:** UDP (без контроля доставки), TCP (с контролем доставки).

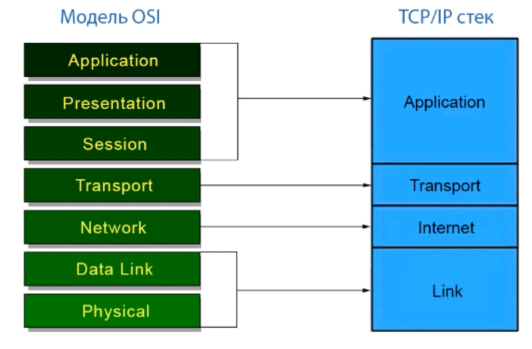
**Сеансовый уровень (L5)** - поддержка связи между приложениями;

**Уровень представления (L6)** - преобразует данные из одной формы в другую (кодирование и декодирование);

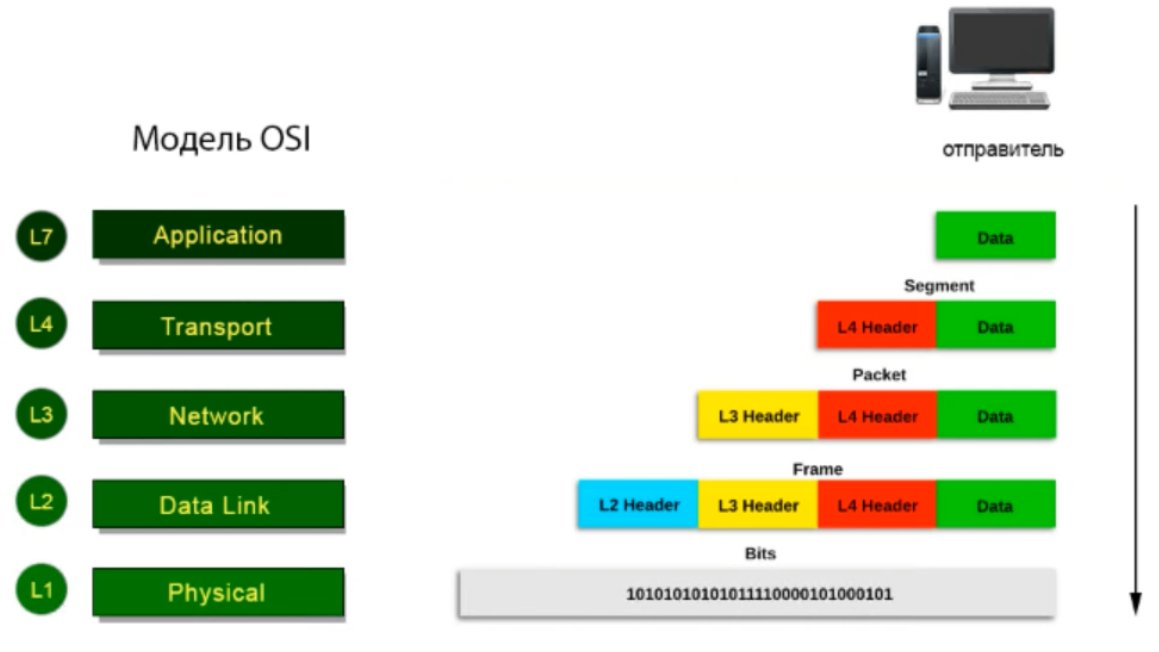
**Прикладной уровень (L7)** - для взаимодействия приложений со всеми другими уровнями.

*Благодаря многоуровневой системе, при написании приложения, программисту не надо думать как его программа будет общаться с 1 по 4 уровень.*

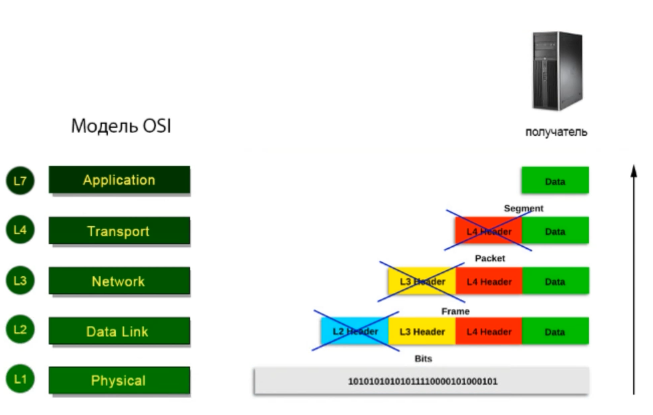
Модель **TCP/IP** - всё тоже самое только менее детально, более просто представление процессов, не отличается от OSI.



**Процесс передачи.** Каждый уровень инкапсулирует, заворачивает информацию прошлого уровня в свою оболочку (заголовки). Уровни 6-5 нас не интересуют.



Процесс получения, мы декапсулируем, чтобы получить данные.



**Структура заголовка протокола Ethernet**

Из чего состоит **L2 Header**

**(определяет как данные будут передаваться внутри канальной среды):**



* **MAC получателя** стоит в начале, для того чтобы умные Свитчи с технологией [Cut-Through](https://en.wikipedia.org/wiki/Cut-through_switching), которая принимает решение до получения всего кадра, на какой порт предложить кадр.
* **Тип (длина)** - указывается длина данных, которые содержатся в след. поле. Без неё было бы не понятно, когда заканчиваются данные и начинается чек сумма.
* **SNAP/LLC** - указывает для какого вышестоящего протокола предназначаются данные (обычно для IP, раньше был [IPX](https://ru.wikipedia.org/wiki/IPX));
* **+ данные** - ваша информация, которую вы передаете, до 1500 байт - это размер [MTU](https://ru.wikipedia.org/wiki/Maximum_transmission_unit) (т.к. чек сумма работает неэффективно после 1500 байт).

Но его можно увеличить в настройках сетевой карты, повысив скорость работы сети но понизив эффективность работы проверки целостности. Это можно делать, если вы уверены что у вас сеть работает стабильно. Эти увеличенные кадры называются [jumbo frame](https://ru.wikipedia.org/wiki/Jumbo-%D0%BA%D0%B0%D0%B4%D1%80) (jumbo-кадр). Но и принимающая сторона должна уметь работать с jumbo-кадрами. Кроме того это повышает нагрузку на роутер.

* **FCS** - чек сумма, для проверки не побился ли кадр при передаче (плохо обжат провод, неверно работает оборудование, были эл. маг наводки, в итоге 0 превратился в 1, а 1 в 0 и всё, сбой). Как это работает: при отправке данных считается чек сумма (математ операция), при получении выполняется та же операция, если сходится то всё хорошо. Если кадр побился, то он будет отправлен по новой.
* **Преамбула** - использовалась в топологии с общей шиной, когда одновременно было невозможно передавать кадры одновременно, она предотвращала коллизии, сообщая все участникам, кто сейчас будет передавать данные.

**Структура IP заголовка (L3 Header)**

для передачи между канальными средами

****

*Появляется IP адресация!*

**Поля:**

1. **Версия** - какая версия IP протокола v4 или v6
2. **IHL** - размер заголовка (многие поля плавающие в L3)
3. **Тип обслуживания** - для механизма очередей, в какой очерёдности посылать пакет (при большой нагрузке можем сказать насколько важный пакет для передачи)
4. **Длина пакета** - какого размера весь пакет, включая пользовательские данные.
5. **Время жизни (TTL)** - для того чтобы предотвратить петли маршрутизации. Мы задаём определённое количество жизней (255 обычно) проходя через каждый роутер отнимается “одна жизнь”. В итоге через 255 роутеров, пакет удалится. Это необходимо, чтобы не было блуждающих в сети пакетов.
6. **Протокол** - для какого вышестоящего протокола пакет (для UDP или TCP);
7. **Контрольная сумма заголовка** - считается чек сумма от полей (**не проверяется целостность данных!!**), только целостность заголовка.
8. **Опции** - редко используется, дописывается для специфических устройств (например - система контроля дверных замков);
9. **Смещение** - добивает 00 к полю Опции.
10. **Данные** - наши данные.

**Идентификатор, Флаги и Смещение фрагмента** - используются когда мы не можем передать данные одним пакетом, и разбиваем его на несколько пакетов.

**Например, разбили наши данные на 4 части,** то:

* **Идентификатор** - пишется единый для всех 4х ID.
* **Флаги** - указано что пакет фрагментированный а не единый. В последнем фрагменте также стоит флаг что он последний.
* **Смещение фрагмента** - нумерация пакета, какой пакет по счёту.

**Структура UDP заголовка (L4 Header)**



Осуществляет разделение трафика приложений.

Мы хотим поиграть в игру [Delta Force](https://ru.wikipedia.org/wiki/Delta_Force_(%D0%B8%D0%B3%D1%80%D0%B0)) по сети через UDP, для этого:

1. Отправляем запрос, генерируя случайный **порт источника** (например 2000),
2. **Порт назначения указываем** = 35.
3. Сервер игры принимает на 35 порту данные,
4. Меняет местами порт источника и порт назначения и отвечает на 2000 порт.

**Контрольная сумма** - если не сходится, пакет отбрасывается (удаляется). Такое допустимо при отправке аудио или видео сигнала. Но недопустимо при передаче например электронной почты.

*UDP просто отправляет данные, без установки сессии, ему не важно данные дойдут или нет.*

**Структура TCP заголовка**



**Функции:**

1. Выполняет функцию **контроля доставки**. Если контрольная сумма не сходится, то данные запрашиваются еще раз.
2. **Механизм сборки пакета в правильной последовательности** (при разбивке данных на разные пакеты). Наши пакеты могут прийти не в правильной последовательности, TCP всё равно соберёт всё верно.
3. **Установление сессии** путем трёхэтапного рукопожатия.
4. Пори источника - случайный.
5. Порт назначения - обычно конкретный, исходя из [сервисов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BF%D0%BE%D1%80%D1%82%D0%BE%D0%B2_TCP_%D0%B8_UDP).
6. Порядковый номер - рассмотрим позже
7. Номер подтверждения - рассмотрим позже
8. **Длина заголовка** - определяет сколько бит заголовок, сколько бит данные.
9. Флаги - для установления сессии
10. Размер окна - число, говорящее на какое число пакетов отправлять подтверждение о получении пакетов. Размер этого числа зависит от качества работы сети, постепенно увеличивается, если потери встречаются очень редко.
11. Контрольная сумма - математ. операция от всех полей и данных для контроля целостности.
12. Указатель важности - уровень важности данного пакета.
13. Опции - у TCP есть масса дополнений и расширений, взаимодействие с которыми происходит через это поле.

**Процесс установления TCP сессии:**

1. Генерируется случайный порт источника;
2. Порт назначения всегда определен, например 80 порт, если мы стучимся к веб-серверу;
3. В поле порядковый номер генерируется случайное число;
4. Устанавливается флаг SYN (флаг желания установки сессии);

**Данные отправляются, доходят до сервера;**

1. Сервер меняет местами порт источника и порт назначения;
2. Порядковый номер - генерируется любое число;
3. Номер подтверждения, прибавляет 1 к порядковому номеру, который пришел;
4. К флагу SYN добавляется флаг ACK

**Отправляет данные клиенту;**

1. Клиент меняет местами порт источника и порт назначения;
2. Номер подтверждения, прибавляет 1 к порядковому номеру, который пришел;
3. Оставляет только флаг ACK

**Данные отправляются, доходят до сервера;**

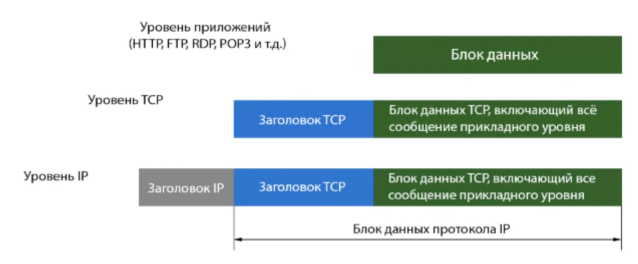
Установка сессии завершена! У нас произошло трехэтапное рукопожатие (3-way handshake).

Когда сессия установлена, поля Порядковый номер и Номер подтверждения используются по другому. В Порядковом номере указывается номер пакета, в номере подтверждения указываются номера пакетов которые уже были получены стороной отправления.

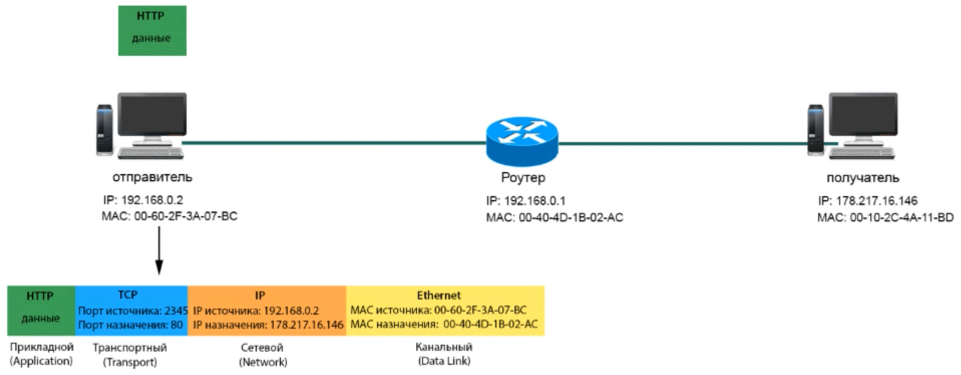
Подробно рассмотрим поле **данные**, для каждого из уровней.

1. На канальном уровне данными является: Заголовок IP + другие данные;
2. Данными для заголовка IP будет: Заголовок TCP с данными;
3. Данными для заголовка TCP будет: например протокол http с пользовательскими данными (например, с html кодом);

Всё это дело называется ”Инкапсуляция данных”



## Рассмотрим процесс передачи данных по сети через роутер



У нас есть отправитель, который хочет отправить какие то HTTP данные. начинает работать модель OSI которая

1. берет эти данные,
2. добавляет TCP заголовок,
3. добавляет IP заголовок,
4. добавляет Ethernet заголовок.

* **Транспортный уровень:** придумывает себе порт, порт получения известен.
* **На сетевом уровне:** задается IP получателя и отправителя (для передачи данных в разных канальных средах);
* **На канальном уровне:** задаются MAC получателя в нашей кнанальной среде (т.е. роутера) и MAC отправителя.

Роутер: отрезает Ethernet заголовок с MAC (он больше не нужен) и добавляет свой Ethernet заголовок, вбивая туда как MAC отправителя свой, и MAC получателя.

## Рассмотрим процесс передачи данных по сети через коммутатор

## 

Всё тоже самое, только канальный уровень не отрезается и MAC сразу задан у целевой машины.

# Виды сетей и среды передачи сигнала

## Виды сетей

* **PAN (Personal Area Network)** - максимум 8 участников, радиус действия до 30 метров (Пример: блютус).
* **LAN (Local Area Network)** - от десятка до сотен участников, радиус действия до 100 метров. Если сделать сеть более 100 метров (длина одного сегмента (например: ПК - свич), то будет происходить затухание сигнала.
* **CAN (Campus Area Network)** - объединяет в себе несколько LAN. Представляет из себя большую, многосегментную LAN. Многосегментная LAN, сгруппированные LAN но не с помощью интернета, а одним проводом, кампусы.
* **MAN (Metropolitan Area Network)** - сети масштаба города. Работают на расстоянии тысяч метров и объединяет тысячи участников. Пример - интернет провайдер или WIMAX точки доступа.
* **WAN (Wide Area Network)** - глобальная компьютерная сеть. Миллионы абонентов (Интернет).

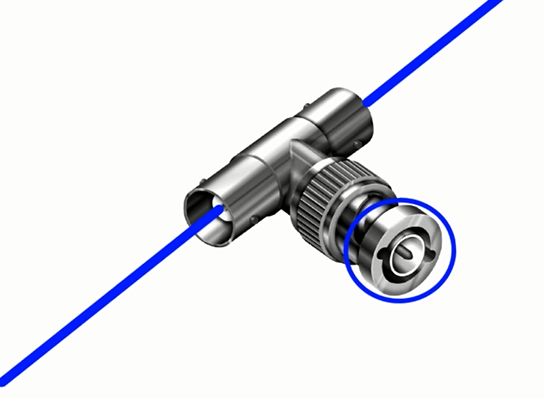
## Виды физических сред передачи данных

* **Коаксиальный кабель** (устаревший)
* **Медный кабель** (витая пара)
* **Оптический кабель** (одномодовый и многомодовый)
* **Радиоканалы** (Wi-Fi, WiMAX. MMDS, Bluetooth, спутники)

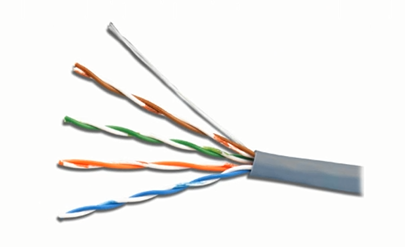
## Коаксиальный кабель



**КК** - устаревший, использовался до 2000 года, внутри медь. Постепенно начали переходить на медный кабель. Внутри электрические импульсы. Затухание через 100 метров. Работал по типу шины, подключался в сетевую карту:

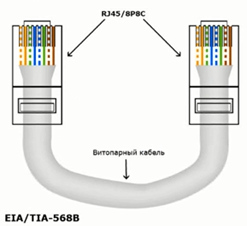


## Витая пара



**Витая пара** - сначала он содержал 2 витые пары, 4 жилы. Их скручивают для уменьшения наводок на данный кабель. Если проложить этот кабель около сидовой линии, он будет работать плохо, т.к. проходящий ток будет создавать магнитное поле и будет влиять на нашу слаботочную витую пару, превращая 1 в 0 и наоборот.

Коннектор RJ-45 обжимается на жилы.



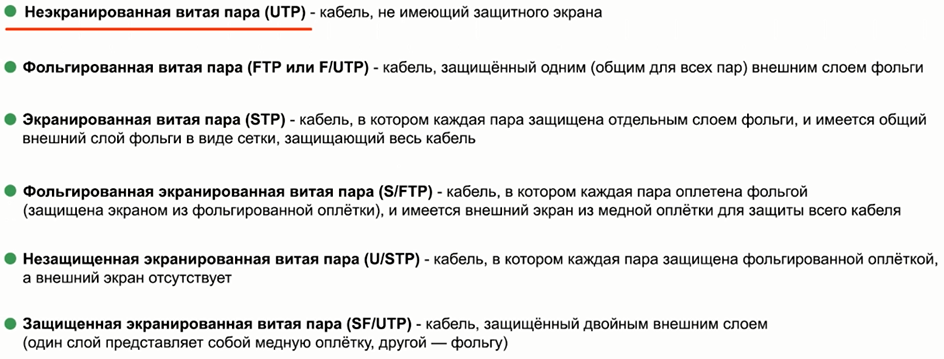
Есть два типа обжима кабеля, прямой и перекрестный. Сейчас обычно **используется прямой**. Перекрёстный использовался для соединения двух компьютеров между собой, это было в старых стандартах, современные сетевые карты могут само подстраиваться в нужный режим работы.



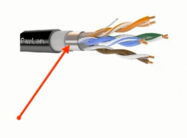
Кримпер - инструмент для обжимки кабеля.



## Виды кабелей типа “витая пара”

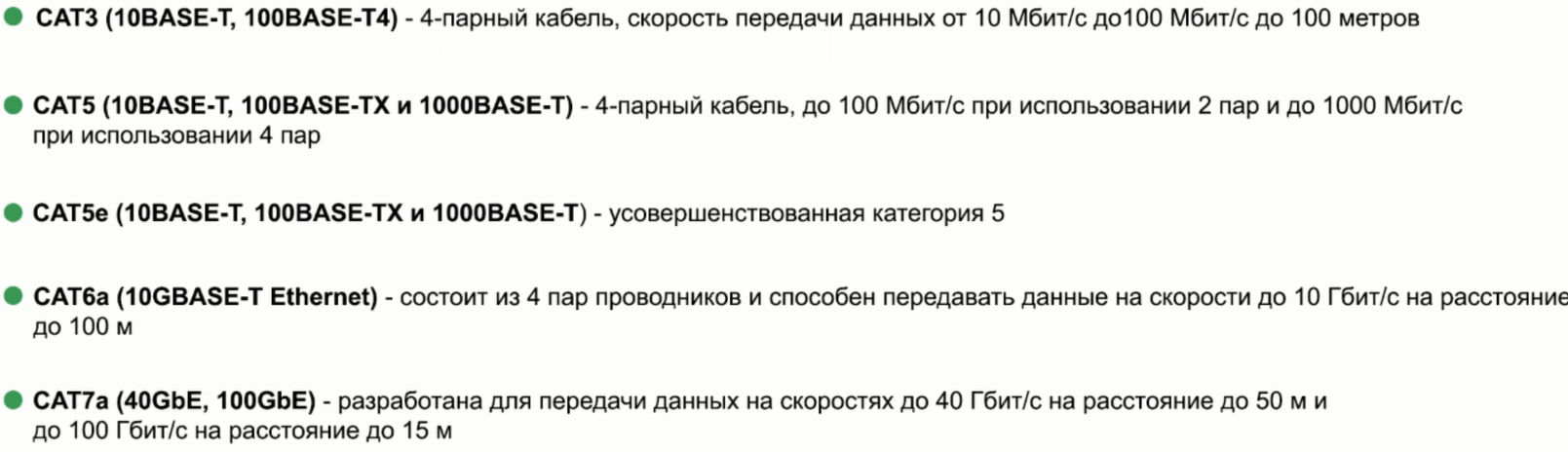
UTP - используют внутри здания

FTP - обычно используют на улице, когда надо объединить два здания.



И далее по уровню защищенности, исходя из задач. Разница только в степени защищенности кабеля от наводок и других воздействий.

## Категории кабелей “витая пара”



Каждая категория определяет какую частоту сигнала будет выдерживать кабель, чем выше частота сигнала - тем выше скорость передачи. На самом деле категорий больше. При этом изменяется и шаг завивки провода, все завивки делаются с на разном расстоянии.

## Медные провода

Медный провод с SFP разъемом для соединения свичей.



Кабеля для создания стекируемых свичей - работают на высоких скоростях, 34Гб и 140 Гб. Мы соединяем эти свичи в стек - чтобы они могли работать как одно единое логическое устройство.



## Оптический кабель



**Оптический кабель** - распространяются световые потоки, затухание через сотни километров, он не подвержен наводкам. Используются для передачи на большие расстояния, как правило провайдерами.

Коннекторы для оптической сети. Их нельзя просто обжать как витую пару, это довольно сложный процесс.



Витая пара бывает одномодовая и многомодовая. Для передачи в одну сторону или для передачи в две стороны. Многомодовая менее эффективна и работает на меньшее расстояние. Сингл более дорогая, но более качественная. Оптические кабеля очень хрупкие, если его пережать, перегнуть, прищемить, может повредится структура кабеля. Изгибать надо аккуратно.



Медиаконвертер - преобразует световые потоки в электрические сигналы которые передаются по витой паре.



## Протоколы канального уровня

Важно! Не стоит путать **протокол Ethernet** и **8P8C штекер!**



Это **не Ethernet**, а **8P8C штекер** и **UTP провод** (витая пара), еще его называют [**RJ-45**](https://ru.wikipedia.org/wiki/Registered_Jack)**,** что тоже не совсем верно, **RJ-45** это **стандарт** сетевого интерфейса.