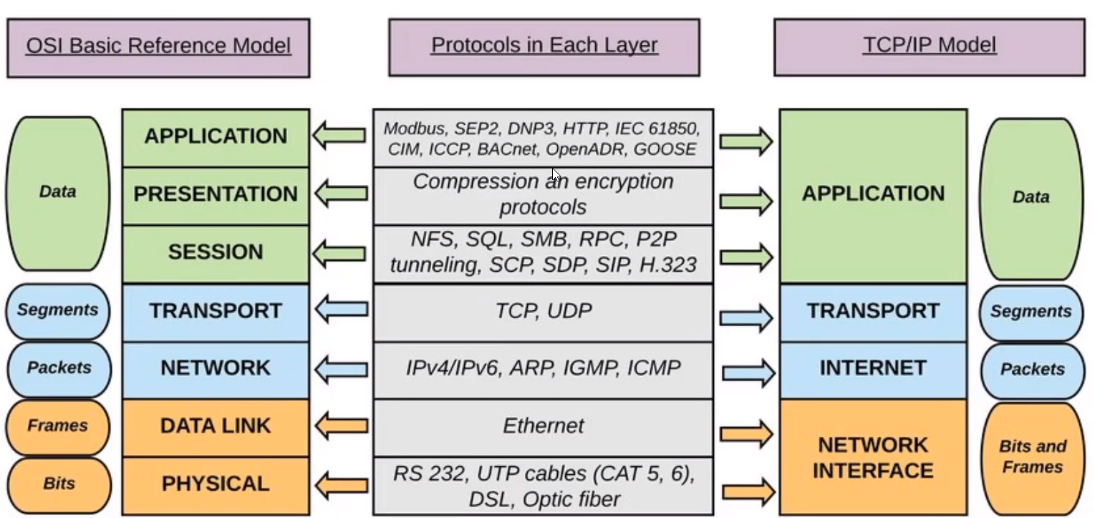
**TCP/IP** — сетевая модель передачи данных. Наборы правил, решающих задачу по передаче данных, составляют стек протоколов передачи данных, на которых базируется Интернет. Название TCP/IP происходит из двух важнейших протоколов семейства — Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были первыми разработаны и описаны в данном стандарте. Также изредка упоминается как модель DOD (Department of Defense) в связи с историческим происхождением от сети ARPANET из 1970-х годов (под управлением DARPA, Министерства обороны США).

Каждый уровень может взаимодействовать только с соседним уровнем

****

**Application** (Прикладной) – на этом уровне работает большинство сетевых приложений. Эти программы имеют свои собственные протоколы обмена информацией

* HTTP/HTTPs (**TCP порт 80/443**) – передача данных в Web
* FTP (**TCP порт 20 для передачи, 21 для управляющих команд**) – передача файлов по сети
* SSH (**TCP порт 22**) - позволяет производить удалённое управление операционной системой
* DNS (**UDP порт 53**)
* DHCP (**UDP порт 67 для передачи, 68 для приема**) - Протокол, позволяющий компьютеру автоматически получать настройки сети от DHCP сервера
* SMTP/IMAP/POP3 (**TCP порт 25/143/110**) – для отправки и получения почты

**Transport** (Транспортный) – протоколы решают проблему негарантированной доставки сообщений («дошло ли сообщение до адресата?»), а также гарантировать правильную последовательность прихода данных. В стеке TCP/IP транспортные протоколы определяют, для какого именно приложения предназначены эти данные.

**И TCP, и UDP используют для определения протокола верхнего уровня число, называемое портом. (порты есть только у протоколов Application уровня)**

* TCP (IP идентификатор 6) — «гарантированный» транспортный механизм с предварительным установлением соединения, предоставляющий приложению надёжный поток данных, дающий уверенность в безошибочности получаемых данных, перезапрашивающий данные в случае потери и устраняющий дублирование данных
* UDP (IP идентификатор 17) протокол передачи датаграмм без установления соединения. Также его называют протоколом «ненадёжной» передачи, в смысле невозможности удостовериться в доставке сообщения адресату, а также возможного перемешивания пакетов

**Internet** (сетевой) – передает данные из одной сети в другую

* IPv4/IPv6
* ICMP - используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают
* IGMP - протокол управления групповой (multicast) передачей данных в сетях, основанных на протоколе IP. IGMP используется маршрутизаторами и IP-узлами для организации сетевых устройств в группы. Используется только в сетях IPv4, так как в IPv6 групповая передача пакетов реализована через протокол Multicast Listener Discovery.
* OSPF/RIP (**TCP порт 89/UDP порт 520**)
* ARP – определить MAC адрес по IP адресу

**Network Access** (Канальный) - описывает способ кодирования данных для передачи пакета данных на физическом уровне (то есть специальные последовательности бит, определяющих начало и конец пакета данных, а также обеспечивающие помехоустойчивость), так же этот уровень соединяет машины между собой в локальной сети

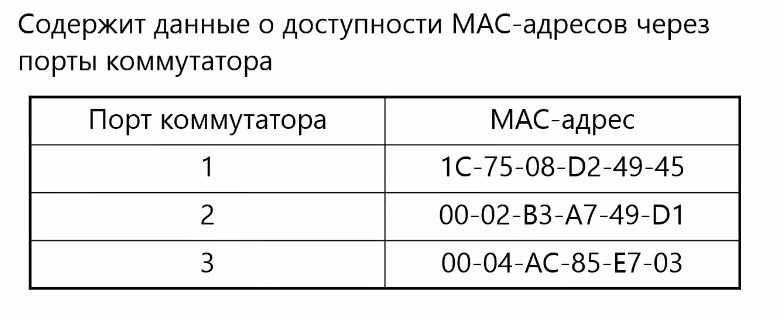
* Ethernet – связь по кабелю
* STP (Spanning Tree Protocol) – устраняет петли в топологии сети
* IEEE 802.11 – беспроводная связь

**Канальный уровень** – на этом уровне происходит трансформация информации в электрические сигналы и передача их по проводу (Ethernet) или по воздуху (WiFi). MAC адрес находится на этом уровне. Для того что бы компьютеры могли между собой общаться, достаточно только одного MAC адреса, но это должга быть локальная сеть. **Данные которые передаются - кадры**

Именно на канальном уровне работает сетевое оборудование, как Switch – коммутатор. Он служит для того, что бы передавать данные внутри сети между хостами

Когда данные попадает на него из физического уровня, switch:

1. преобразует их в кадры, считывает мак адрес получателя и отправителя
2. отправляет их согласно таблице коммутации. Если switch не знает хоста с указанным мак адресом получателя, то
   1. кадры отправляются на все хосты в сети – switch flooding
   2. Нужный хост отвечает, а не нужный отбрасывает кадр
   3. Заполняет таблицу коммутации (какой мак адрес к какому порту на коммутаторе подключен). Использует FF:FF:FF:FF:FF:FF – широковещательный MAC адрес

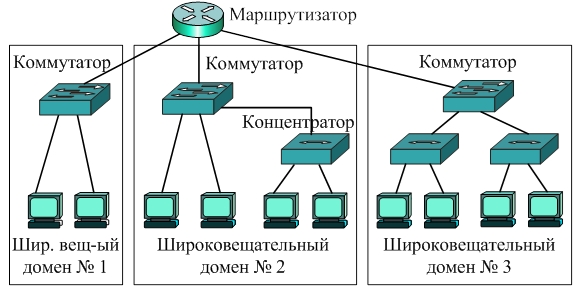


***MAC-адрес****(****мак адрес****)* - это уникальный идентификатор сетевого интерфейса (обычно сетевой карты) для реализации коммуникации устройств в сети на канальном уровне. Если есть виртуальная машина, или установлено несколько сетевых карт, то будет несколько MAC адресов.

Так, это уникальный номер, который хранится в доступной только для чтения памяти, назначенный сетевой карте ее производителем.

Длинна MAC-адреса равна 48 битам, что обеспечивает 248 (или 281,474,976,710,656) общее число всех возможных адресов. Как правило, MAC-адрес записывается как шесть групп двойных **шестнадцатеричных чисел (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)**, разделенных символами "-" или ":". Например, MAC-адрес может выглядеть так - "00:11:22:33:44:55", или так - "67-78-89-AB-CD-EF". Производитель

Широковещательный запрос (switch flooding) FF:FF:FF:FF:FF:FF – широковещательный домен не проходит через роутер.



Чтобы узнать MAC-адреса сетевых устройств на своем компьютере, достаточно просто напечатать "ifconfig" в командной строке Linux, "ipconfig /all" в командной строке Windows или "networksetup -listallhardwareports" в Mac OS. Он будет отображен, как значение полей "HWaddr", "Physical address" или "Ethernet address" соответственно.

В общем, MAC-адрес назначается на постоянной основе устройству и не может быть изменен. Но в некоторых случаях существует возможность изменения MAC-адреса на программном уровне. Это известно, как MAC-спуфинг. Он может применяться для:

* При наличии в одной сети компьютеров с одинаковыми MAC адресами
* В случае привязки на старый MAC адрес информации, программ, оборудования
* При ограничении провайдером количестве подключений, если имеется несколько компьютеров
* При испытании инструментов управления сетью

**При отправке сообщения клиент указывает MAC адрес получателя, а так же свой MAC адрес**

**Сетевой уровень –** основная задачаэто маршрутизация трафика из одной сети в другую, используя таблицу маршрутизации. **Данные которые передаются - пакеты**

Именно на сетевом уровне работает сетевое оборудование, как Router – роутер. Он служит для передачи информации между сетями. Разбивать 1 сеть на много разных меньших нужно для того, что бы компьютеры из 1 сети не имели доступ к компьютерам 2 сети. А так же для уменьшения нагрузки на сеть



Хост что бы отправить информацию на другой хост:

1. Хост отправителя определяет с помощью маски подсети является ли хост получателя в этой же сети, или в другой
2. Если подсеть одна, то пакет отправляется на хост получателя
3. Если подсеть другая, то хост отправителя отправляет пакет на шлюз по умолчанию

Когда данные попадает на роутер из физического уровня, Router:

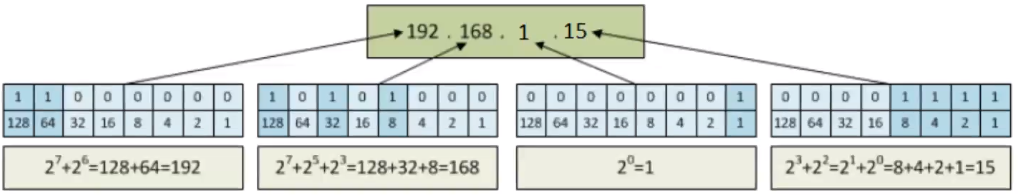
1. преобразует их пакеты
2. смотрит IP адрес получателя
3. смотрит в таблицу маршрутизации
4. отправляет пакет далее по сети

**IP адрес** - Уникальный числовой идентификатор устройства в компьютерной сети, работающий по протоколу TCP/IP

**IP** протокол — это протокол так называемого сетевого уровня. Задача этого уровня — доставка ip-пакетов от компьютера отправителя к компьютеру получателю используя таблицу маршрутизации. По-мимо собственно данных, пакеты этого уровня имеют ip-адрес отправителя и ip-адрес получателя. Номера портов на сетевом уровне не используются. Какому порту, т.е. приложению адресован этот пакет, был ли этот пакет доставлен или был потерян, на этом уровне неизвестно — это не его задача, это задача транспортного уровня.

**IPv4**адрес выглядит как 4 числа через точку. Используется двоичная система: 128 64 32 16 8 4 2 1

10011000 = 128 + 0 + 0 + 16 + 8 + 0 + 0 + 0 = 152



**Маршрутизация:** OSPF (89 порт) ; RIP (UDP 520 порт)

**ARP** – позволяет определить MAC адрес по IP адресу – это широковещательный протокол, который отправляет пакеты всем пользователям в сети и по этим данным строит таблицу маршрутизации

1. Смотрим по маске что хост в нашей сети

2. Отправляем broadcast запрос с посланием у кого нужный IP адрес

3. у кого нужный IP адрес отправляет ответ с MAC адресом

ARP запрос не уходит дальше чем локальная сеть, так как его запросы не проходят через маршрутизатор

Адреса IP, используемые в локальных сетях, относят к частным.  
Адреса [Intranet](https://ru.wikipedia.org/wiki/Intranet):

* 10.0.0.0/8 = 11111111.00000000.00000000.00000000. Значит первая часть будет неизменна – 10. А последующие 255.255.255 (10.0.0.0 – 10.255.255.255)
* 172.16.0.0/12 = 11111111.11110000.00000000.00000000. Значит первая будет неизменна – 172. Вторая часть – остаток 15. Остальные части остаток 255. Значит (172.16.0.0 – 172.31.255.255)
* 192.168.0.0/16

Адреса для внутреннего использования:

* 127.0.0.0/8 — используется для коммуникаций внутри хоста (см. [localhost](https://ru.wikipedia.org/wiki/Localhost)).
* блок с 169.254.1.0 по 169.254.254.255 (подсеть 169.254.0.0/16 за исключением подсетей 169.254.0.0/24 и 169.254.255.0/24) — используется для автоматической настройки сетевого интерфейса в случае отсутствия DHCP (см. [link-local](https://ru.wikipedia.org/wiki/Link-local_address)).

**При отправке сообщения клиент указывает IP адрес получателя, а так же свой IP адрес**

В протоколе IP поддерживается несколько типов адресов, некоторые из которых имеют эквиваленты на канальном уровне.

* Направленные (unicast) — адреса, которые обозначают отдельный сетевой интер­фейс.
* Групповые (multicast) — адреса, идентифицирующие группу узлов.
* Широковещательные (broadcast) — адреса, обозначающие все узлы локальной сети.
* Альтернативные (anycast) — адреса, обозначающие любой из группы узлов.

**Маска подсети**

**Маска подсети** - Битовая маска для определения по IP-адресу адреса подсети и адреса узла (хоста, компьютера, устройства) этой подсети**.** Если компьютер-получатель принадлежит этой же сети, что и компьютер-отправитель, то пакет передается ему напрямую, в противном случае пакет отправляется на шлюз по умолчанию, который далее, по известным ему маршрутам, передает пакет в другую сеть. Маска может быть представлена в виде **255.255.255.255** или **/32**

Маска подсети предназначена для определения — принадлежит ли компьютер-получатель к этой же локальной сети или нет. Если компьютер-получатель принадлежит этой же сети, что и компьютер-отправитель, то пакет передается ему напрямую, в противном случае пакет отправляется на шлюз по умолчанию, который далее, по известным ему маршрутам, передает пакет в другую сеть.

В терминологии [сетей](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) [TCP/IP](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2_TCP/IP) **маской подсети** или **маской сети** называется [битовая маска](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%B0%D1%81%D0%BA%D0%B0), определяющая, какая часть [IP-адреса](http://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0 находится в сети 12.34.56.0[/24](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%BA%D0%B0_%D0%BF%D0%BE%D0%B4%D1%81%D0%B5%D1%82%D0%B8#.D0.9C.D0.B0.D1.81.D0.BA.D0.B8_.D0.BF.D1.80.D0.B8_.D0.B1.D0.B5.D1.81.D0.BA.D0.BB.D0.B0.D1.81.D1.81.D0.BE.D0.B2.D0.BE.D0.B9_.D0.BC.D0.B0.D1.80.D1.88.D1.80.D1.83.D1.82.D0.B8.D0.B7.D0.B0.D1.86.D0.B8.D0.B8_.28CIDR.29) с длиной префикса 24 бита. В случае адресации [IPv6](http://ru.wikipedia.org/wiki/IPv6) адрес 2001:0DB8:1:0:6C1F:A78A:3CB5:1ADD с длиной префикса 32 бита (/32) находится в сети 2001:0DB8::/32.

Другой вариант определения — это определение подсети IP-адресов. Например, с помощью маски подсети можно сказать, что один диапазон IP-адресов будет в одной подсети, а другой диапазон соответственно в другой подсети.

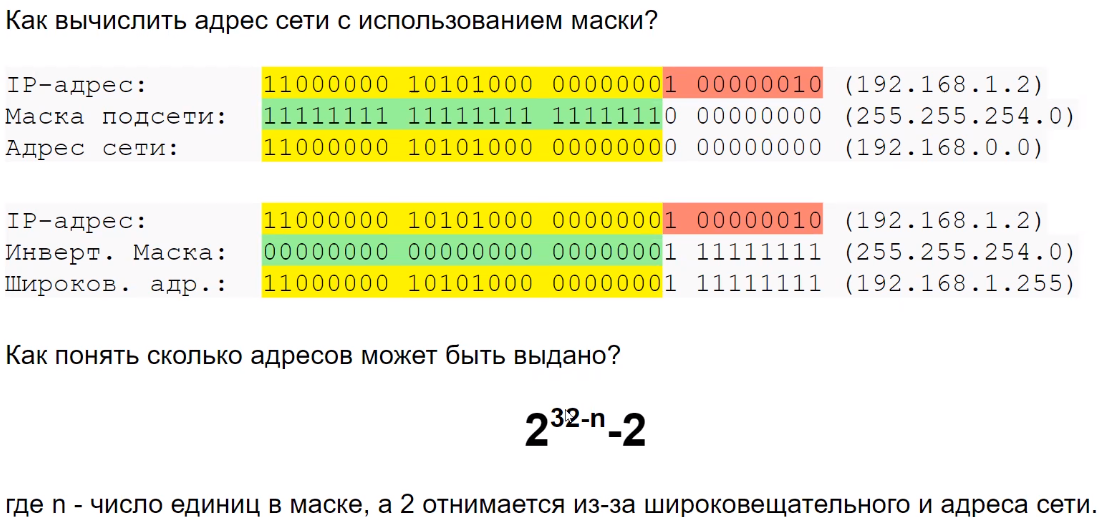
Для вычесления адреса сети мы суммируем IP адрес а так же маску подсети. 1 + 1 = 1. В любом другом случае пишем 0 в Адрес сети

Для вычесления широковещательного адреса мы инвертируем маску, а после суммируем. 1 + 1 = 1 ; 0 + 1 = 1 ; 1 + 0 = 1. В случае 0 + 0 = 0.

Желтым – адрес сети.

Красным – адрес хоста

Зеленым – маска подсети



Разбиение одной большой сети на несколько маленьких подсетей позволяет упростить [маршрутизацию](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F). Например, пусть таблица маршрутизации некоторого [маршрутизатора](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80) содержит следующую запись:

|  |  |  |
| --- | --- | --- |
| **Сеть назначения** | **Маска** | **Адрес шлюза** |
| 192.168.1.0 | 255.255.255.0 | 10.20.30.1 |

Пусть теперь маршрутизатор получает пакет данных с адресом назначения 192.168.1.2. Обрабатывая построчно таблицу маршрутизации, он обнаруживает, что при наложении маски 255.255.255.0 на адрес 192.168.1.2 получается адрес сети 192.168.1.0. В таблице маршрутизации этой сети соответствует [шлюз](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%88%D0%BB%D1%8E%D0%B7) 10.20.30.1, которому и отправляется пакет.

Маски при бесклассовой маршрутизации (CIDR)

Маски подсети являются основой метода [бесклассовой маршрутизации](http://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%81%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%B0%D1%86%D0%B8%D1%8F) (CIDR). При этом подходе маску подсети записывают вместе с IP-адресом в формате «*IP-адрес/количество единичных бит в маске*». Число после знака дроби означает количество единичных разрядов в маске подсети.

Рассмотрим пример записи диапазона IP-адресов в виде 10.96.0.0/11. В этом случае маска подсети будет иметь двоичный вид 11111111 11100000 00000000 00000000, или то же самое в десятичном виде: 255.224.0.0. 11 разрядов IP-адреса отводятся под *адрес сети*, а остальные 32 − 11 = 21 разряд полного адреса (~~11111111 111~~00000 00000000 00000000) — под локальный адрес в этой сети. Итого, 10.96.0.0/11 означает диапазон адресов от 10.96.0.0 до 10.127.255.255.

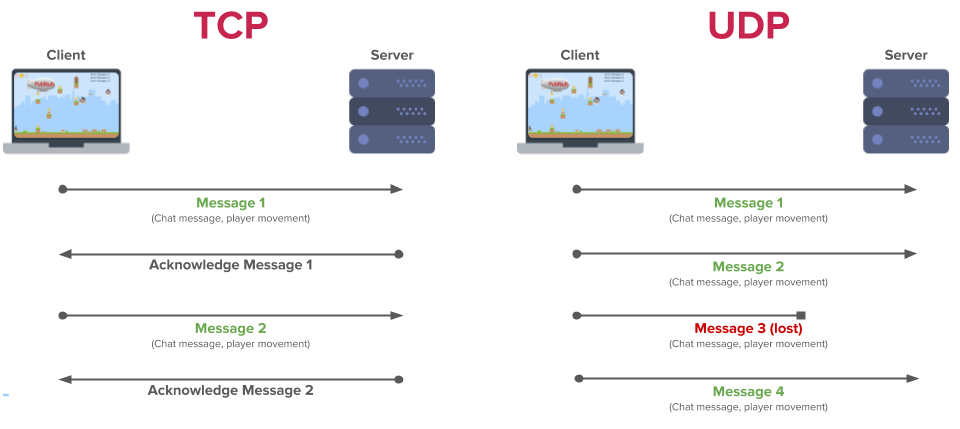
**Транспортный уровень –** занимается передачей данных и адресацией к конкретному приложению.

**Данные которые передаются – сегменты (TCP)/датаграммы (UDP)**

**TCP и UDP** — это протоколы так называемого транспортного уровня. Транспортный уровень находится над сетевым. На этом уровне к пакету добавляется порт отправителя и порт получателя.

**TCP**— это протокол с установлением соединения и с гарантированной доставкой пакетов. Сначала производится обмен специальными пакетами для установления соединения, происходит что-то вроде рукопожатия (-Привет. -Привет. -Поболтаем? -Давай.). Далее по этому соединению туда и обратно посылаются пакеты (идет беседа), причем с проверкой, дошел ли пакет до получателя. Если пакет не дошел, то он посылается повторно («повтори, не расслышал»). Сообщения, Отправка файлов, Оплата, Online Games (положение пули, человека), загрузка файлов, текстовых сообщений, ...

**UDP**— это протокол без установления соединения и с негарантированной доставкой пакетов. (Типа: крикнул что-нибудь, а услышат тебя или нет — неважно). Skype, Online Games (задний план, статичные объекты), видео, аудио, ...



**Порт** – служит что бы машина кому мы передаем данные поняла, какому приложению относятся полученные данные. Порт по сути говорит какой протокол верхнего уровня использовать

**При отправке сообщения клиент указывает PORT получателя (статические порты могут использовать только получатели (сервер)), а так же свой PORT (динамический - 41xxx - 65535). Порт получателя – это порт протокола Application уровня**

Еще нужен для того, что бы идентифицировать ту программу, которой нужно послать запрос, и какая будет посылать. Например у нас на ПК могут быть Telegram, Torrent, открыт браузер. Все они используют HTTP протокол, а так же имеют один IP адрес (нашего компа). По этому, для идентификации программы, нужен порт

**Прикладной уровень.** **Данные которые передаются - data**

**САЙТЫ**: http (80 порт/TCP), https (443 порт/TCP)

**ПОЧТА**:

SMTP (simple message transfer protocol) - отправка почты  
IMAP - делает копии полученного письма  
POP3 - письмо переносится с сервера на компьютер

**ЗАГРУЗКА ФАЙЛОВ С СЕРВЕРА**: ftp (20, 21)

**УДАЛЕННЫЙ ДОСТУП С ШИФРОВАНИЕМ**: SSH (22 порт/TCP)

Например HTTP и FTP — используют надежный протокол TCP, а ***DNS***-сервер работает через ненадежный протокол UDP. Прикладной уровень работает с данными, которые обрабатываются клиентом