Некоторые базовые понятия

- **Хост (Host)** или **Узел (Node) –** компьютер или устройство-участник сети
- **Трафик (Traffic)** - поток данных, передаваемых по сети
- **Локальный хост (Local host) –** текущий хост, создающий (originating) трафик
- **Удалённый хост (Remote host)** – хост, который принимает трафик
- **Клиент (Client)** – компонент системы, запрашивающий у **сервера** ресурсы, услуги, вычислительные мощности
- **Сервер (Server)** – компонент системы, предоставляющий доступ к ресурсам или услугам по запросу **клиента**
- **Интерфейс (Interface)** – сущность, способствующая взаимодействию между собой разнородных сущностей (пользовательский интерфейс – между человеком и компьютером, сетевой интерфейс – между компьютером и сетевым кабелем и пр.)
- **Протокол (Protocol) –** формализованное соглашение о правилах взаимодействия между объектами сети, позволяющее получить успешный результат от взаимодействия. К примеру,
- **Стек протоколов (Protocol Stack) –** иерархически организованный набор протоколов, достаточный для организации взаимодействие узлов в сети. Протоколы в стеке выполняют конкретную задачу – преобразование и подготовку данных, адресацию, передачу и приём и т. д. - и полагаются в своей работе друг на друга (а точнее, на протоколы соседних уровней)
- **API (Application Programming Interface) -** описание способов, которыми одна компьютерная программа может взаимодействовать с другим программным кодом – к примеру, соглашения об именовании функций и их аргументов, типов и структур используемых данных

Исторические технологии LAN и WAN

- Исторически глобальные сети появились раньше локальных - первые сети (конец 60х-начало 70х гг.) строились между большими мейнфреймами, а малых персональных компьютеров тогда ещё не существовало
- Глобальные сети строились через арендованные у телефонных компаний линии (leased lines) соединения «точка-точка» (point-to-point). Технологии и протоколы доступа к WAN в то время были технологиями пакетной передачи через телефонные сети (протоколы X.25, Frame Relay, PPP, T1/E1 leased lines)
- Основной виток развития технологий ЛВС начался в начале 80х гг. Основной потребностью потребителей ЛВС в то время был совместный доступ к файлам и принтерам
- Существовало несколько несовместимых между собой конкурирующих технологий ЛВС (Novell NetWare, Banyan VINES, LAN Manager, AppleTalk etc.), реализовывавших весь стек протоколов и приложений от сетевого уровня до уровня приложений
- TCP/IP в 80е гг. существовал только внутри сетей ARPANET и NSFNET

Локальные и глобальные сети

По географическому признаку можно разделить на:

- Глобальные сети (Wide Area Network, WAN) - сети, охватывающие значительную часть Земли или всю Землю, и открытые неограниченному числу людей. Примеры - Internet, международные телефонные сети
- Локальные (Local Area Network, LAN) - сети, покрывающие небольшую территорию, и, как правило, принадлежащие одному владельцу. Пример – сети малых предприятий или домашние сети за роутером

Также встречаются такие термины:

- Городские/региональные (Metropolitan/Regional Area Network, MAN/RAN) - сети, как правило, построенные на технологиях ЛВС, но разросшиеся до масштабов города или региона. Примеры - сети городских интернет-провайдеров
- Enterprise Networks - Закрытые локальные сети предприятий. Как и MAN, могут иметь более сложную, чем у LAN, структуру, и использовать для передачи данных оборудование более дорогого класса ради обеспечения производительности, безопасности и отказоустойчивости. Географически могут быть как небольшими, так и простираться через страны и континенты в случае крупных корпораций.

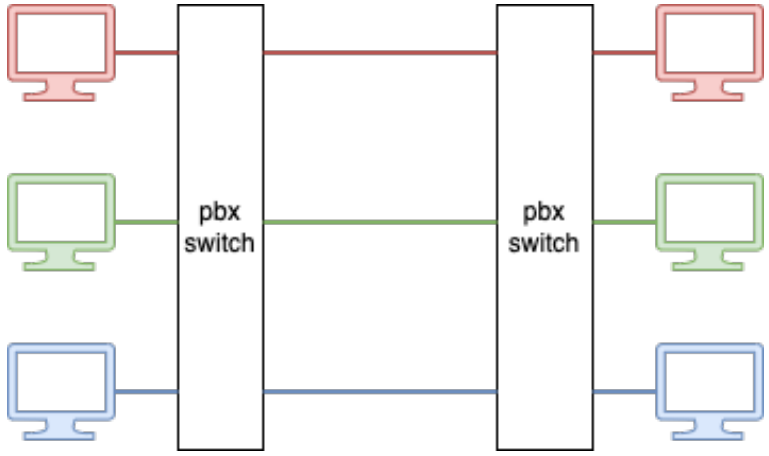
Концептуальные различия ЛВС и ГВС

- Конкретные локальные сети находятся как правило под единым административным контролем, глобальные сети же (не только Интернет, но и классические телефонные или телевизионные) же как правило есть конгломерация многих локальных сетей, потому не принадлежат никому конкретному в отдельности (хотя и могут иметь объединяющие участников административные структуры – такие, как ICANN, ITU или 3GPP Consortium)
- За счёт этого в глобальных сетях, как правило, возможно **получить**, но невозможно **гарантировать** какие-либо параметры связи между двумя случайными точками сети (ширина канала, потери пакетов, latency итд.). Внутри локальных сетей же возможно иметь гарантированное качество связи

Как частный вывод - исторически технологии построения локальных сетей были высокоскоростными, но рассчитанными на малую дальность.

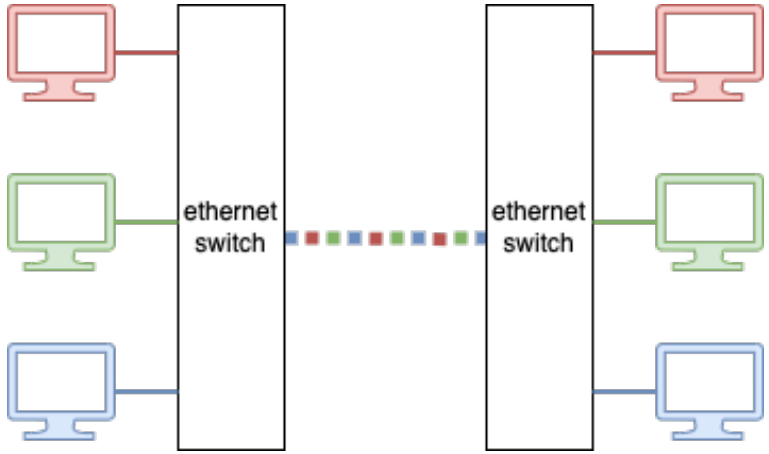
Технологии же абонентского доступа к глобальной сети («последняя миля») зачастую были низкоскоростными и завязанными на существовавшие уже телефонные сети. И до сих пор насущной проблемой часто является проблема «последней мили» - обеспечения качественного подключения конечного абонента или устройства к опорной мультигигабитной сети

Принципы коммутации в сетях



Коммутация каналов (circuit switching) – установление между конечными узлами фиксированных каналов передачи данных, зарезервированных полностью между двумя узлами на время передачи.

Пример: классические телефонные сети



Коммутация пакетов (packet switching) – последовательная передача данных частями небольшого размера (пакетами), что позволяет разделять одни и те же физические каналы между многими узлами сети

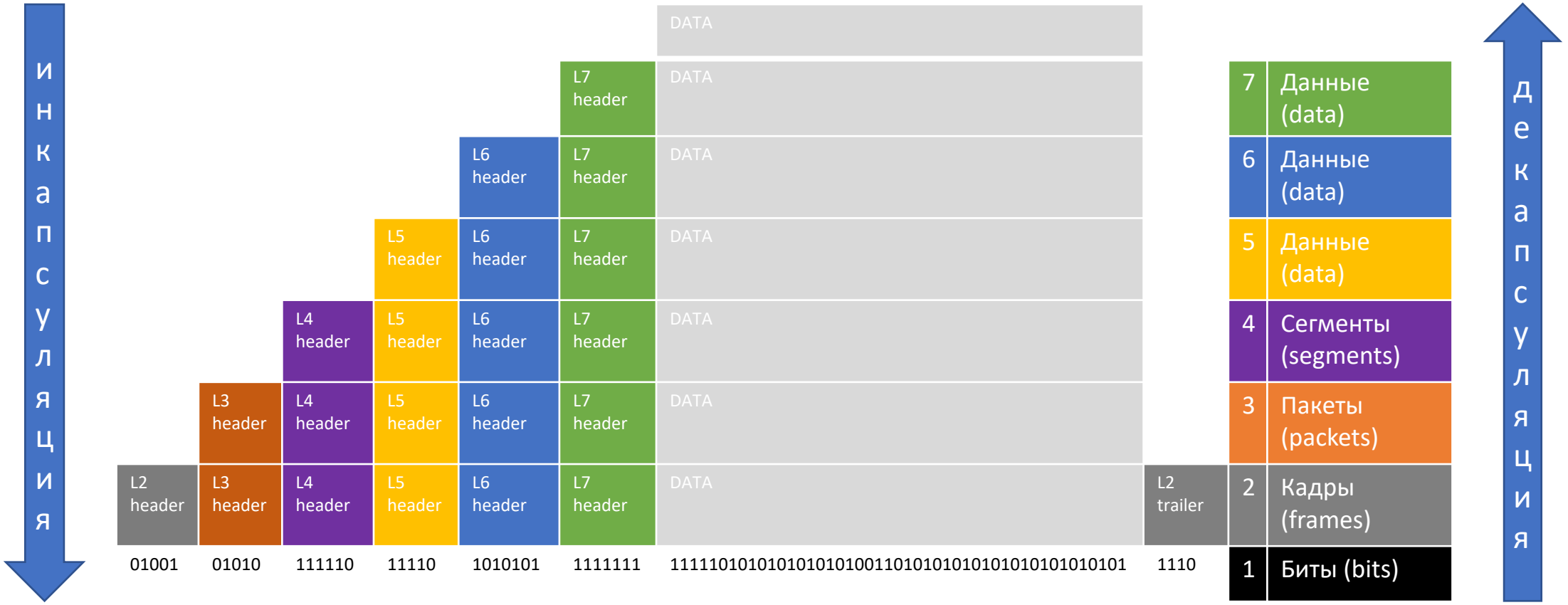
Пример: Ethernet, HDSPA+/LTE

Сетевая модель OSI

7	**Уровень приложения (Application layer)**
6	Уровень представления (Presentation layer)
5	Сеансовый уровень (Session layer)
4	Транспортный уровень (Transport layer)
3	Сетевой уровень (Network layer)
2	Канальный уровень (Data Link layer)
1	Физический уровень (Physical layer)

- Создана в конце 70-х годов Международной Организацией по Стандартам (International Standards Organization, ISO)
- Описывает абстрактную модель, принципы работы и иерархию сетевых технологий
- Представляет 7 уровней с чётко разграниченными функциями (физический, канальный, сетевой, транспортный, сеансовый, представления, приложения)
- В процессе передачи данные проходят «сверху вниз» по уровням сетевой модели с добавлением служебной информации протоколов каждого уровня (инкапсуляция)
- В процессе приёма данные проходят «снизу вверх» с удалением служебной информации протоколов своего уровня (декапсуляция)
- Сетевая модель OSI была ранней попыткой стандартизации и обобщения различных технологий передачи данных и предоставляла также свой стек протоколов, но не выдержала конкуренции со стеком протоколов TCP/IP, но тем не менее является эталонной концепцией для обучения и документирования

Инкапсуляция и декапсуляция



Стек протоколов и сетевая модель TCP/IP

4	**Уровень приложений (HTTP, SMTP, SSH)**
3	Транспортный уровень (TCP, UDP)
2	Сетевой уровень (IP, IPv6)
1	Канальный уровень (DHCP, ARP)

- Изначально создан в начале 70х по инициативе Управления перспективных исследовательских проектов Министерства обороны США (DARPA), и на основе него была построена сеть ARPANET, ставшая прообразом Интернета
- Ныне управляется IETF (Internet Engineering Task Force), имеющую более открытую и свободную политику, нежели аналогичные организации стандартизации в электросвязи (IEEE, CCITT/ITU). Технические документы IETF открыты для всех, и имеют название Requests for Comments (RFC)
- Имеет более упрощённое разделение по уровням, чем сетевая модель OSI, но покрывает все предоставляемые ею функции
- Не определяет физический уровень и фрейминг канального уровня, полагается на инкапсуляцию IP-пакетов в PDU соответствующих технологий, но использует некоторые протоколы, работающие на L2 (как правило, привязаны к соответствующим технологиям канального уровня)

Современная модель

L7	Уровень приложений	HTTP, SMTP, SSH
L4	Транспортный уровень (TCP, UDP)	TCP, UDP, SCTP
L3	Сетевой уровень (IP, IPv6)	IP, IPv6
L2	Канальный уровень	Ethernet
L1	Физический уровень	802.11* (wi-fi), DWDM, 1GBase-T Ethernet

Сетевая модель OSI была ранней попыткой стандартизации и обобщения различных технологий передачи данных

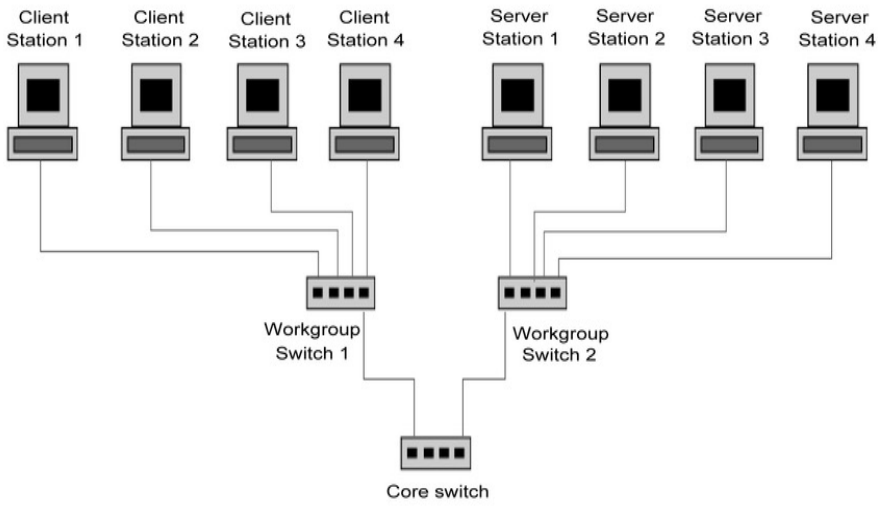
ISO предоставляла также свой стек протоколов, но он появился слишком поздно, был весьма громоздким и сложным и не выдержал конкуренции со уже существовавшим и более открытым стеком протоколов TCP/IP

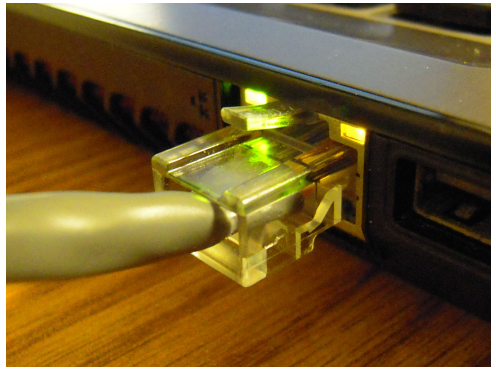
Тем не менее модель OSI является эталонной концепцией для обучения и документирования

А что сейчас?

- Благодаря развитию Интернета в 1990х гг. стек протоколов TCP/IP также стал стандартом в локальных сетях и вытеснил конкурирующие технологии (AppleTalk, Novell IPX/SPX, NetBIOS)
- Ethernet за счёт быстрого наращивания скоростей передачи данных (100мбит/с в 1995г., 1Гбит/с в 1998г., 10Гбит/с в 2002г. и далее) стал победившей технологией построения локальных сетей и вытеснил такие популярные ранее технологии ЛВС, как Token Ring и FDDI
- Развитие оптоволоконной связи и появление стандартов Ethernet, построенных на оптоволокне, позволило строить на основе Ethernet большие географически распределённые сети. Так Ethernet. постепенно замещает классические WAN-технологии (T1/E1, SDH/Sonet, xDSL etc.)
- Также Ethernet постепенно проникает в более узкоспециализированные ниши (как, например, Storage Area Networks, классически построенные на FibreChannel) и позволяет быть единым транспортом для других технологий и использовать одно и то же оборудование и/или стандарты (гиперконвергентные сети)
- Появление в конце 1990х-начале 2000х годов серии стандартов 802.11 (wi-fi) беспроводных локальных сетей
- Развитие пакетной передачи данных (3G/4G) в сетях мобильной телефонной связи принесло высокоскоростной доступ в Интернет с мобильных устройств в зоне покрытия радиосвязью без необходимости иметь отдельное физическое подключение

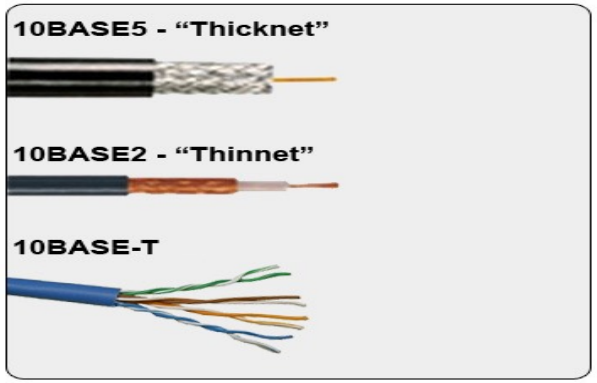
Сеть Ethernet

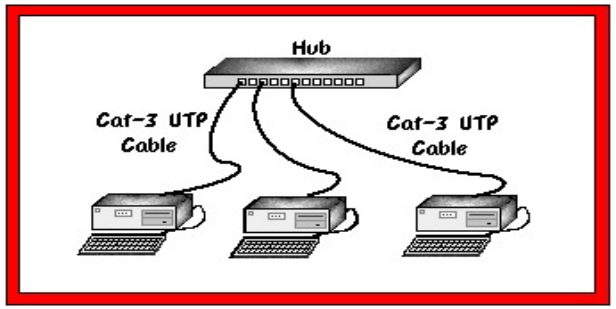


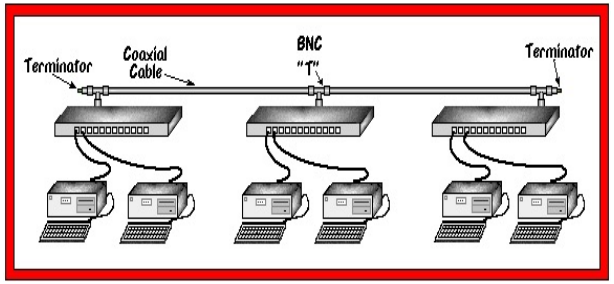


- Технология организации многоточечной локальной сети, покрывающая уровни 1 и 2 модели OSI
- Позволяет широковещательную (broadcast) передачу. Сегмент сети, на который распространяется широковещательная передача называется broadcast domain
- Адресация внутри broadcast-домена осуществляется с помощью MAC-адресов (Media Access Control, контроль доступа к среде) – шестибайтовых значений, уникальных для физического сетевого интерфейса, обычно представляемых в шестнадцатиричном виде (пример 01:0a:de:ad:be:ef).
- Сеть Ethernet имеет древовидную структуру, не позволяющую иметь запасных путей к другому сегменту. Закольцовывание соединений приводит к широковещательному шторму – аварийной ситуации, препятствующей работе сети. Для обеспечения резервирования сети существуют отдельные протоколы (Spanning Tree, ERPS, etc.)
- Ethernet предоставляет контроль целостности фрейма за счёт CRC-суммы в трейлере фрейма, но не производит восстановления фрейма. Повреждённый фрейм уничтожается. Целостность данных обеспечивается вышележащими протоколами

Классический Ethernet (10Base2, 10Base-T)

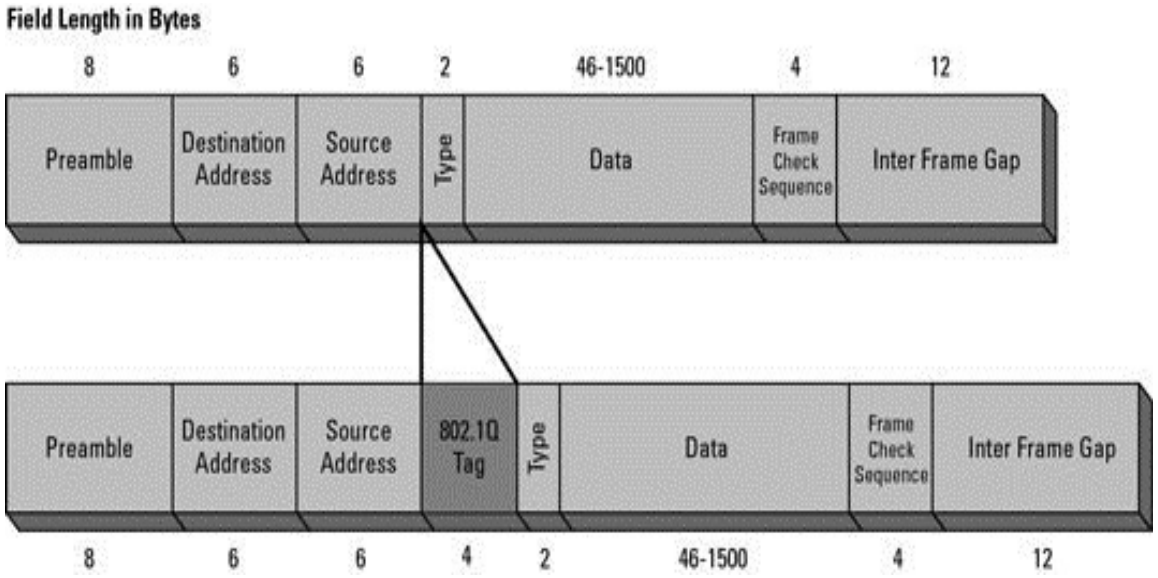






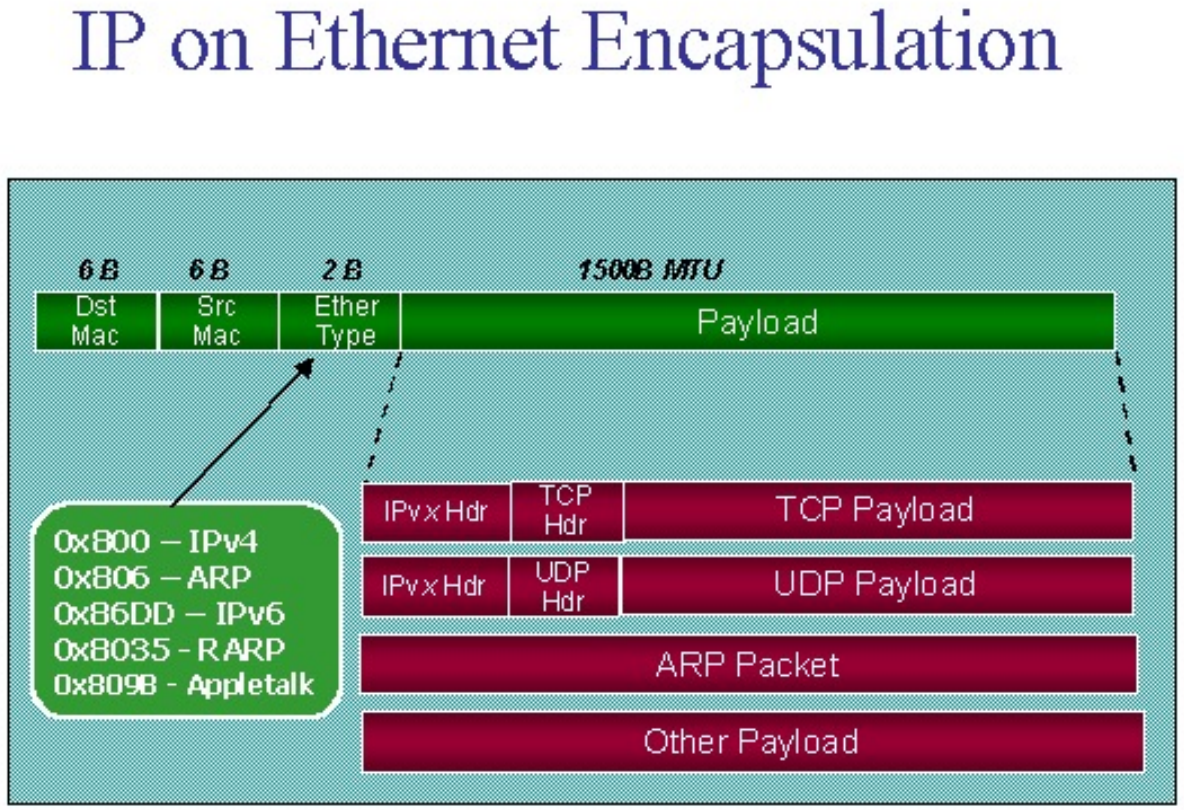
- Предоставлял скорость линии в 10Мбит/с в полудуплексном режиме (в один момент времени происходит либо приём электрического сигнала, либо его передача)
- Топология сети – либо «шина», либо «звезда»
- В сети с топологией «шина» использовался общая шина из коаксиального кабеля, к которой все устройства подключались через Т-образные BNC-коннекторы (стандарты 10Base-2, 10Base-5). Максимальная длина сегмента сети – 180м для «тонкого коаксиала» и до 500м для «толстого коаксиала»
- В сети с топологией «звезда» устройства подключались кабелем типа «витая пара» через центральное устройство – **концентратор**. Концентраторы работают на уровне L1, ретранслируя сигнал с одного порта на все остальные, таким образом, также создавая общую электрическую шину
- Все устройства сети получают один и тот же электрический сигнал. Как следствие – в сегменте может вести передачу только одно устройство в момент времени. В противном случае произойдёт искажение сигнала – **коллизия**
- Для обеспечения работы сети используется **принцип Carrier Sense Multiple Access with Collision Detection (CSMA/CD) — множественный доступ с прослушиванием несущей и обнаружением коллизий**. При обнаружении сигнала в линии во время передачи устройство останавливает передачу и возобновляет её спустя случайное время
- Все устройства в сети получают один и тот же сигнал. Для определения кто именно получатель пакета устройства смотрят на указанный в заголовке MAC-адрес получателя. Если он совпадает с собственным – устройство принимает и обрабатывает кадр. В противном случае оно его игнорирует
- …но можно перевести сетевую карту в «неразборчивый» (promiscuous) режим, и получать весь поступающий сигнал.
- Сегмент сети, в котором распространяется единый на всех электрический сигнал, называется **доменом коллизий (collision domain)**
- Из-за этого производительность классической Ethernet-сети была сравнительно низкой, поскольку любой трафик в сети создавал сигнал в линии во всём сегменте и препятствовал передаче. Ряд конкурирующих технологий ЛВС зачастую были производительнее при меньших физических скоростях (4mbit/s у Token Ring, 2.5mbit/s у ARCNET)
- Для повышения производительности сети применяли **мосты и коммутаторы (многопортовые мосты)** – устройства, работающие на уровне L2 и разделявшие сегменты сети. Они передавали электрический сигнал в сегмент только тогда, когда он предназначался этому сегменту или же широковещательный трафик. Мосты и коммутаторы являлись дорогими устройствами в то время.

Формат кадра Ethernet

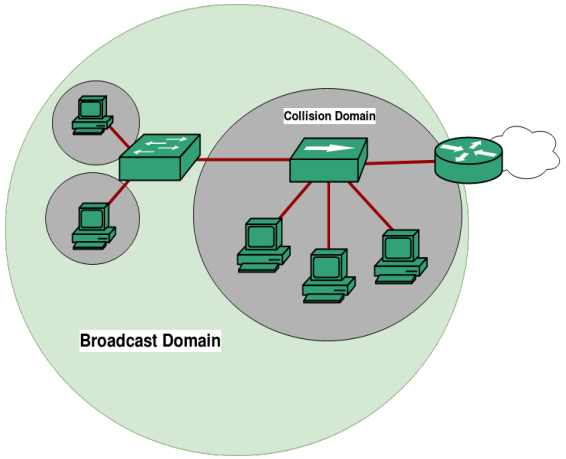


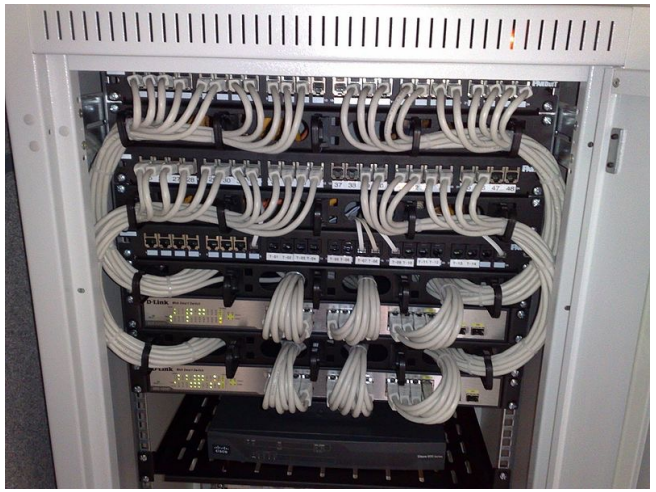
- Preamble – не является значимой частью заголовка. Постоянный битовый паттерн, для электрической синхронизации с приёмником
- Destination Address (6 байт) – MAC-адрес принимающего устройства. В случае широковещательной передачи имеет значение FF:FF:FF:FF:FF:FF
- Source Address (6 байт) – MAC-адрес отправителя
- EtherType (2 байта) – тип инкапсулированных внутри данных
- 802.1Q Tag (4 байта) – опционально. Метка Virtual LAN позволяет создавать broadcast-домены (виртуальные локальные сети) безотносительно физической топологии сети.
- Data (46-1500 байт) - данные. Имеют произвольную длину от минимальной длины до **MTU**
- FCS (4 байта) – Контрольная сумма пакета. Служит для проверки целостности пакета и обнаружения факта его повреждения при передаче
- Inter-Frame Gap - не является частью заголовка. Просто пауза между передачей двух соседних кадров.

Передача IP-пакета через Ethernet



Современный Ethernet





- Начиная со стандарта Fast Ethernet (100мбит/с) используется только витая пара. Сеть имеет только топологию «звезда». Также появились стандарты для передачи через оптоволокно, что увеличило дальность физических линков
- Развитие электронных компонент значительно снизил стоимость коммутаторов, которые полностью вытеснили концентраторы и мосты.
- **Коммутатор** – интеллектуальное устройство, работающее на уровне L2, и имеющее несколько портов для подключения устройств или же других сегментов сети. Коммутатор имеет внутри таблицу MAC-адресов и соответствий их портам, и отправляет трафик только в порты назначения. В случае если порт назначения для MAC-адреса неизвестен или же в случае широковещательного трафика коммутатор рассылает пакеты на все порты
- Таким образом, исчез общий **домен коллизий**, но была сохранена возможность многоадресной и широковещательной передачи (сохранилось понятие **броадкаст-домена**). На физическом уровне сеть перестала быть сетью с общей шиной и стала сетью «точка-точка» - электрический сигнал распространялся только между коммутатором и конечным устройством, и не влиял на остальные станции
- Это также позволило использовать **полнодуплексный** режим – возможность одновременно передавать и получать пакеты. В более современных версиях (начиная с Gigabit Ethernet) полудуплексного режима нет вообще
- Эти изменения радикально увеличили производительность сетей на базе Ethernet, что привело к вымиранию конкурирующих технологий проводных локальных сетей

Краткая технологическая история Интернета

- Разработки начались в конце 60х - начале 70х гг. по заказу Министерства Обороны США (Department of Defence) внутри Агентства Министерства обороны США по перспективным исследованиям (DARPA) для экспериментальной сети ARPANET
- Целью разработки было создание компьютерной сети, устойчивой к выходу из строя или уничтожению в ходе военных действий отдельных её сегментов и обеспечивавшей надёжную и бесперебойную передачу данных по низкокачественным каналам связи, а также пригодной для использования с различными технологиями (протяжённые телефонные линии или спутниковая связь)
- 1 января 1983 года ARPANET полностью перешла на использование TCP/IP и стала первой в мире сетью, использующей принципы **маршрутизации** пакетов – пересылки пакетов до конечного узла через другие узлы или сети, имеющие **маршруты** до конечного адресата. В начале-середине 80гг. также появились протоколы **динамической маршрутизации**, позволявшие в реальном времени обмениваться маршрутами между узлами сети с выбором лучших маршрутов и отсечением вышедших из строя
- В 1984-1987г. Был разработан протокол DNS (Domain Name System), позволявший использовать вместо IP-адресов **доменные имена**, являющиеся более человеко-читаемыми
- Успех и бурный рост ARPANET заинтересовал научные круги, и на базе разработок ARPANET была в 1984г. создана сеть NSFNET. Из ARPANET впоследствии сети военного назначения были выделены в отдельную сеть MILNET, а ARPANET стал конкурирующей с NSFNET научной сетью. За NSFNet вскоре закрепилось разговорное название **«Интернет».**
- С конца 80х гг. шёл процесс бурной коммерциализации Интернета. ARPANET была закрыта в 1990г., а её мощности были переданы NSFNET, до середины 90х гг. бывшей опорной сетью для Интернета, но с 1995 года маршрутизацией в Интернете занимаются коммерческие **провайдеры**, что сделало опорные каналы связи Интернета децентрализованными. NSFNet всё еще существует как академическая университетская сеть
- Разработка 1989-1993г. вокруг идеи **гипертекста** (системы из текстовых страниц и документов, имеющих перекрёстные ссылки на другие документы в Сети) технологий **HTTP** (HyperText Transfer Protocol, протокол передачи гипертекста), **HTML** (HyperText Markup Language, язык разметки гипертекста), **URL** (Uniform Resource Locator, единообразный указатель ресурса) привело к появлению **Всемирной Паутины (World Wide Web, WWW) –** распределённой системы, предоставлявшей на основе гипертекста доступ к ресурсам на различных подключённых к Сети компьютерах. Всемирная Паутина стала вскоре самым востребованным сервисом в Интернете, обогнав традиционные сервисы передачи файлов и электронной почты
- Таким образом, слово **Интернет** следует понимать как **«межсетевой», «между сетями»** - сеть, связывающую между собой узлы через другие, промежуточные сети с динамическим (псевдослучайным) выбором маршрутов, что позволяет при наличии достаточного количества физических каналов быть устойчивой к их повреждению

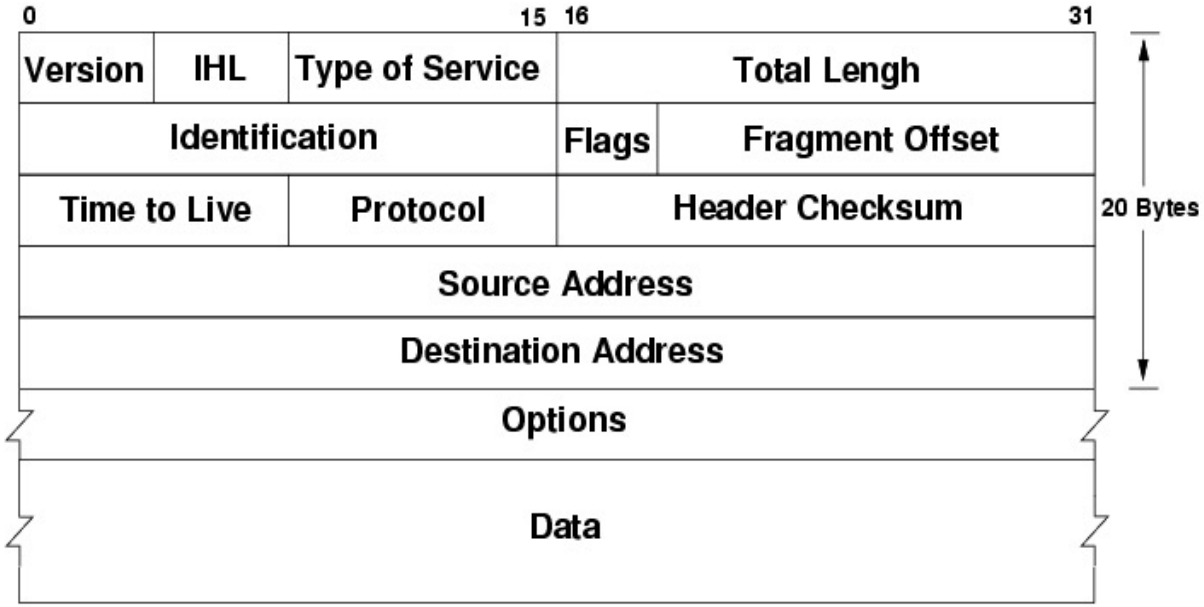
Базовые протоколы стека TCP/IP

- **IP (Internet Protocol, межсетевой протокол)** – протокол уровня 3 модели OSI (сетевого), ответственный за адресацию узлов сети и маршрутизацию пакетов до них. Не обеспечивает надёжность доставки. Ключевыми сущностями для протокола IP являются **IP-адрес**, **маска подсети** и **маршрут**. Ввиду исчерпания IP-адресов получает развитие в виде протокола **IPv6**
- **TCP (Transmission Control Protocol**, протокол контроля передачи) – протокол 4 уровня модели OSI (транспортного), ответственный за контроль передачи данных, сегментацию данных при отправке и сборку пакетов в правильном порядке при получении. Обеспечивает надёжную доставку пакетов за счёт установления предварительного логического соединения методом **«трёх рукопожатий (3-way handshake)**, периодического подтверждения доставки пакетов и переотправки потерянных. Ключевой сущностью для протокола TCP является **порт** – 16-битное целое число (от 1 до 65535), позволяющее идентифицировать конкретное приложение на узле, отправляющее трафик (порт отправителя), либо принимающее на удалённом узле (порт получателя)
- **UDP (User Datagram Protocol, протокол пользовательских дейтаграмм)** – протокол транспортного уровня (как и TCP). Простейший транспортный протокол, не использующий концепцию виртуального соединения, не контролирующий передачу и не гарантирующий корректной доставки. Предоставляет необязательный механизм контрольной суммы для проверки целостности пакета. Также как и TCP, имеет такую же концепцию **порта**, но порты TCP и UDP между собой не пересекаются.
- **DNS (Domain Name System, система доменных имён) -** протокол получения информации о соответствии **доменных имён** IP-адресам. Нужен для получения IP-адреса по доменному имени хоста для последующей отправки пакета по адресу получателя. Основой DNS является представление об иерархической структуре имени и *зонах*. Каждый сервер, отвечающий за имя, может *передать* ответственность за дальнейшую часть домена другому, что позволяет возложить ответственность за актуальность информации на узлы, отвечающие только за «свою» часть доменного имени
- **DHCP (Dynamic Host Configuration Protocol, протокол динамической конфигурации хоста) –** широковещательный протокол, позволяющий хосту получить настройки IP (IP-адрес, маска подсети, основной шлюз, DNS-сервера итп.) в автоматическом режиме без необходимости ручной настройки
- **ARP (Address Resolution Protocol, протокол разрешения адреса) –** широковещательный протокол установления соответствия IP-адресов MAC-адресам в Ethernet-сети, служащий для определения к какой именно станции в многоадресной сети отправлять пакет с заданным IP-адресом. Используется в многоадресных широковещательных сетях вроде Ethernet, поскольку IP-адрес есть программный адрес, а за одним проводом могут быть несколько узлов. Не используется в соединениях типа «точка-точка», поскольку данные соединения предполагают только два узла
- **ICMP (Internet Control Message Protocol — протокол межсетевых управляющих сообщений)** – протокол, служащий для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна или хост не отвечает. На основе протокола ICMP основана утилита **Ping**, позволяющая с помощью посылки ICMP Echo Request сообщений определять доступность удалённого узла, процент потерь пакетов по пути и время, за которое пакет достиг узла и пришёл ответ **(задержка, latency)**

Базовые термины TCP/IP

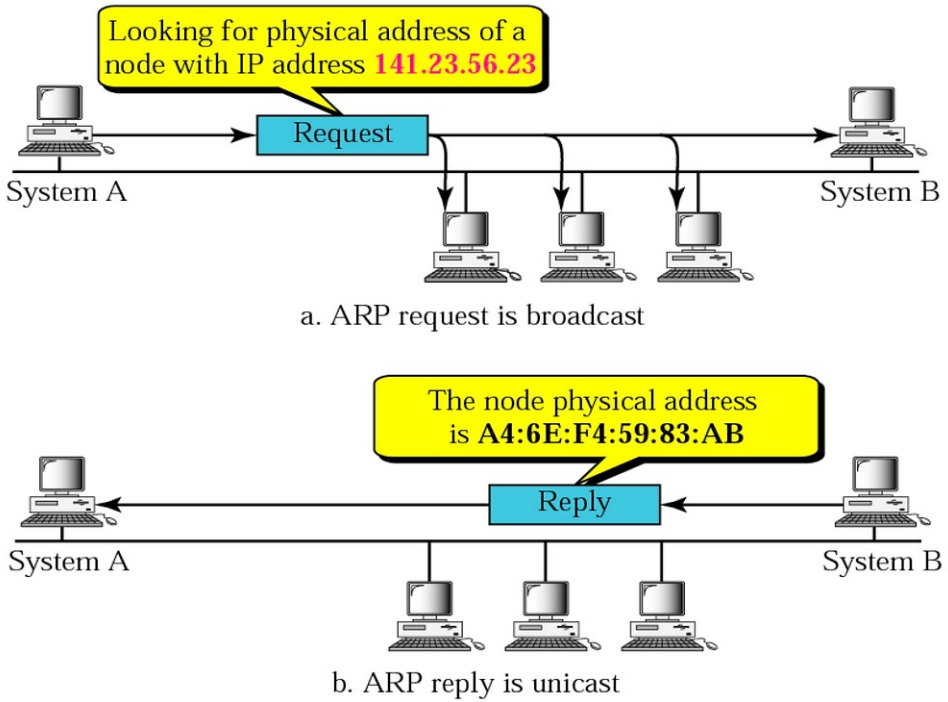
- **IP-адрес (IP address)** – уникальный идентификатор устройства в IP-сети. Представляет собой 4-байтовое число (32-бит), но для человеко-читаемости обычно записывается с разбивкой по октетам в двоично-десятичном представлении, где каждое число (от 0 до 255) соответствует одному байту в адресе (к примеру, 192.168.0.1)
- **Сеть, суперсеть (network, supernet)** - помимо «сети» как группы устройств под одним управлением, способных коммуницировать, также- сравнительно большой диапазон IP-адресов, выделенный (или же полученный от регистратора) для конкретной физической сети. Например, выбранный **приватный диапазон** 10.0.0.0/8 или полученный от регистратора диапазон **внешних адресов** 192.0.2.0/24
- **Подсеть (subnet)** – помимо меньшего физического сегмента большой **сети**, также - сеть (диапазон адресов) меньшего размера, созданная путем деления более крупной сети на равные непересекающиеся части. Размер подсети определяется **маской подсети (subnet mask)**
- **Маска подсети (subnet mask) – 32-битное число,** служащее **битовой маской (bitmask)** для разделения сетевой части (адреса подсети) и части хоста IP-адреса. Представляет собой **непрерывную** последовательность от 0 до 32 двоичных единиц, после которых остаток разрядов представляют двоичные нули. Смешение их недопустимо. Устройства в одной подсети имеют одинаковый адрес подсети и передают данные **на канальном уровне модели OSI**. Устройства в разных подсетях коммуницируют **через маршрутизацию.** Может быть записана, как и IP-адрес, в двоично-десятичной форме (к примеру, 255.255.0.0) или же в виде **префикса в CIDR-нотации -** числом от 0 до 32, обозначающего длину маски в битах (к примеру, подсеть 192.0.2.0/24, где /24 – это маска, равная 255.255.255.0)
- **Маршрут (route) –** запись в **таблице маршрутизации (routing table)** о следующем устройстве в сети **(next hop) ,** которому следует направить пакеты для пересылки в конечную сеть. Next hop’ом может быть либо адресом другой машины **в данной подсети**, либо в определённых случаях сетевой интерфейс
- **Маршрутизатор, «роутер» (router) -** устройство, выполняющее **пересылку (forwarding)** пакетов данных между разными IP-сетями. Как правило представляет собой либо отдельный компьютер с несколькими сетевыми интерфейсами со специальным ПО для маршрутизации (software router), либо специализированное устройство (hardware router) со встраиваемым ПО для тех же целей
- **Шлюз (gateway) –** исторически устройство, позволяющее коммуницировать между собой сетям, построенным на основе разных протоколов и технологий (как например TCP/IP и Novell IPX/SPX). В настоящее время ввиду отсутствия конкурентов у TCP/IP почти всегда является синонимом маршрутизатора - хотя бывают отдельные специфичные случаи, когда уместна классическое определение (например, TDM/IP gateways)
- **Маршрут по умолчанию, шлюз по умолчанию (default route, default gateway) –** маршрут до подсетей, не имеющих в таблице маршрутизации специфичного маршрута. На большинстве компьютеров является единственным маршрутом в таблице маршрутизации, кроме **локального маршрута** (до своего IP-адреса) и **прямого маршрута** (до своей собственной подсети в конкретный сетевой интерфейс)
- **Дейтаграмма** - пакет данных, отправляемый без предварительной установки соединения. Отправляется получателю без гарантии доставки вообще, без гарантии доставки в правильном порядке, но может содержать контрольную сумму

Протокол IP



- **Version** – версия протокола IP. Всегда 4
- **IHL** – длина IP заголовка в 4-байтных словах (dword). Обычно 5 (5x4=20 байт), но при наличии IP Options может быть больше
- **Type Of Service** – значение, используемое для классификации и приоритизации трафика
- **Total Length** – общая длина пакета включая заголовок
- **Identification** – задаваемый отправителем идентификатор пакета. Используется для сборки пакета из фрагментов
- **Flags** – битовые флаги, относящиеся к фрагментации - *DF – Don't Fragment*, "не фрагментировать" и *MF – More Fragments*, "есть ещё фрагменты"
- **Fragment Offset** - Смещение фрагмента. Значение, определяющее позицию фрагмента в потоке данных. Используется для сборки пакета из фрагментов
- **TTL - Time To Live** - число узлов, которые может пройти этот пакет. Уменьшается на единицу при прохождении каждого узла, при TTL=0 пакет уничтожается с отправкой TTL Exceeded отправителю. Используется для предотвращения бесконечных петель в IP-сети
- **Protocol** - идентификатор протокола следующего уровня
- **Source Address** – IP-адрес отправителя
- **Destination Address** – IP-адрес получателя
- **Options** – Опции протокола. Почти не используются

Address Resolution Protocol (ARP)



- Протокол канального уровня. Служит для получения канального адреса устройства (MAC-адреса) в многоадресной физической сети, для которого известен IP-адрес
- Работает в пределах одной подсети

Принцип работы

- Перед отправкой IP-пакета хост-отправитель отправляет широковещательный запрос **ARP Who-Has** с IP-адресом получателя
- Устройство, владеющее данным IP-адресом, отвечает unicast-сообщением **ARP Reply** конкретному хосту
- Отправитель кеширует ответ в arp-таблице. Дальнейшая отправка пакетов будет происходить без повторного разрешения MAC-адреса согласно содержимому таблицы
- Записи таблицы имеют ограниченное время жизни

Пример адресов и масок подсети (classless)

```
➜  ~ ipcalc 192.185.91.4/18
Address:   192.185.91.4          11000000.10111001.01 011011.00000100
Netmask:   255.255.192.0 = 18    11111111.11111111.11 000000.00000000
Wildcard:  0.0.63.255            00000000.00000000.00 111111.11111111
=>
Network:   192.185.64.0/18       11000000.10111001.01 000000.00000000
HostMin:   192.185.64.1          11000000.10111001.01 000000.00000001
HostMax:   192.185.127.254       11000000.10111001.01 111111.11111110
Broadcast: 192.185.127.255       11000000.10111001.01 111111.11111111
Hosts/Net: 16382                  Class C
```

Пример адресов и масок подсети (classful)

```
➜  ~ ipcalc 192.168.24.1/24
Address:   192.168.24.1          11000000.10101000.00011000. 00000001
Netmask:   255.255.255.0 = 24    11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255             00000000.00000000.00000000. 11111111
=>
Network:   192.168.24.0/24       11000000.10101000.00011000. 00000000
HostMin:   192.168.24.1          11000000.10101000.00011000. 00000001
HostMax:   192.168.24.254        11000000.10101000.00011000. 11111110
Broadcast: 192.168.24.255        11000000.10101000.00011000. 11111111
Hosts/Net: 254                    Class C, Private Internet
```

Таблица маршрутизации

Таблица маршрутизации – таблица, хранящаяся в памяти на маршрутизаторе, которая описывает соответствие между адресами назначения и интерфейсами, через которые следует отправить пакет данных до следующего маршрутизатора

Пример:

```
~ netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags           Netif Expire
default            192.168.0.1        UGScg             en0
127                127.0.0.1          UCS               lo0
127.0.0.1          127.0.0.1          UH                lo0
169.254            link#14            UCS               en0      !
169.254.183.207    10:c3:7b:ac:52:78  UHLSW             en0      !
192.168.0          link#14            UCS               en0      !
192.168.0.1/32     link#14            UCS               en0      !
192.168.0.1        e8:48:b8:3a:79:26  UHLWIir           en0   1171
192.168.0.101      54:3a:d6:1b:42:9c  UHLWI             en0      !
192.168.0.102/32   link#14            UCS               en0      !
192.168.0.255      ff:ff:ff:ff:ff:ff  UHLWbI            en0      !
224.0.0/4          link#14            UmCS              en0      !
224.0.0.251        1:0:5e:0:0:fb      UHmLWI            en0
239.255.255.250    1:0:5e:7f:ff:fa    UHmLWI            en0
255.255.255.255/32 link#14            UCS               en0      !
```

Типы Network Address Translation

- **Статический NAT** — постоянное отображение исходного IP-адреса в конечный IP-адрес «один-в-один». Позволяет устройству внутри сети быть доступным снаружи сети по некому статическому адресу
- **Динамический NAT** — отображение «один-в-один» исходного IP-адреса в случайный конечный адрес из большого общего пула адресов. Если адреса в пуле заканчиваются, то коммуникация становится невозможной
- **Перегруженный NAT (NAPT, NAT Overload, PAT, маскарадинг)** — форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя на стороне NAT-шлюза различные порты отправителя. При перегрузке каждый компьютер в частной сети транслируется в один или несколько «внешних» адресов на стороне шлюза, но с различным номером порта отправителя. Самый популярный тип NAT. *Не позволяет инициировать соединения снаружи к устройствам за NAT, но позволяет экономить «внешние» IP-адреса*

Приватные IP-адреса и подсети

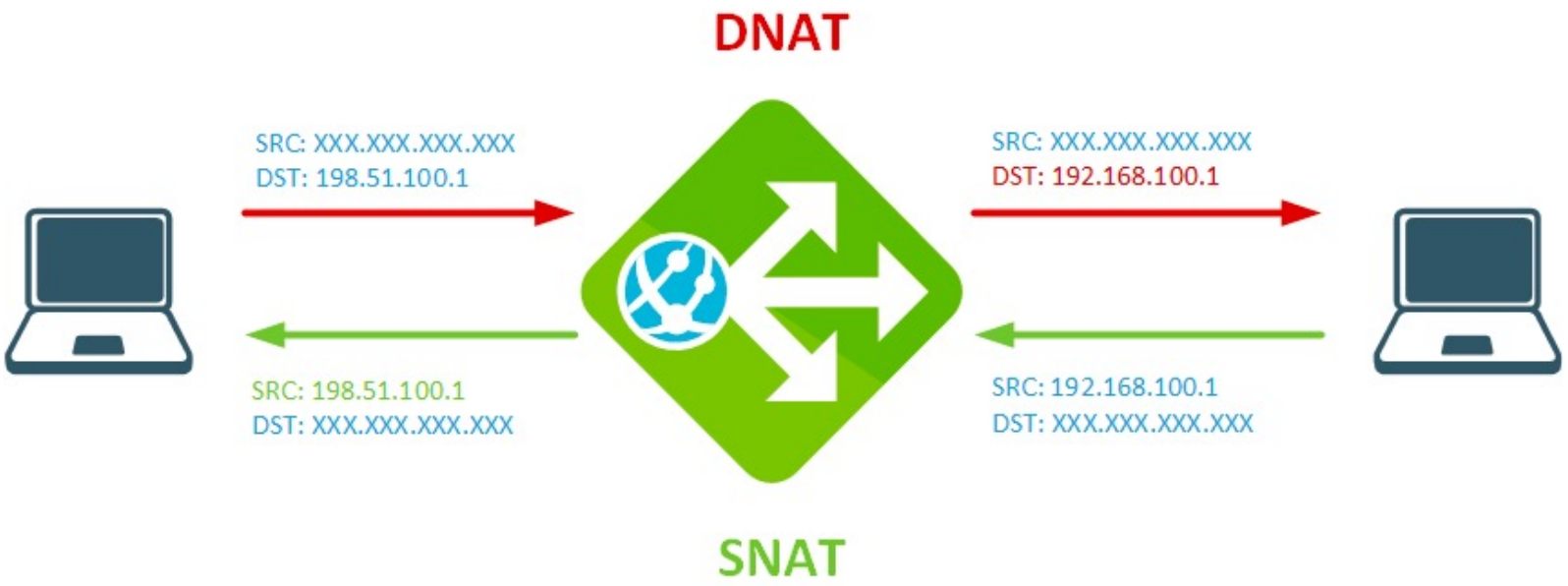
10.0.0.0/8 (от 10.0.0.0 до 10.255.255.255)
172.16.0.0/12 (от 172.16.0.0 до 172.31.255.255)
192.168.0.0/16 (от 192.168.0.0 до 192.168.255.255)

- Определены RFC1918 для экономного использования ip-адресов
- Могут быть использованы только внутри локальной сети
- Могут быть маршрутизируемы, но маршруты до подсетей из этих не должны появляться в глобальной сети
- Требуется применение NAT (Network Address Translation) для возможности коммуникации с хостами в глобальной сети

Зарезервированные адреса и подсети

- **0.0.0.0** – мета-адрес, означающий ”этот хост в этой сети”. Не может быть адресом отправителя или получателя, но может служить маршрутом (префикс 0.0.0.0/0 - синоним маршрута по умолчанию) , а также значит «все адреса на данной машине» (настройка listen tcp 0.0.0.0:8080 значит «слушать tcp-порт 8080 на всех интерфейсах»)
- **100.64.0.0/10** – приватная суперсеть для провайдерского NAT (Carrier Grade NAT). Также, как и сети из RFC1918, эти адреса недоступны в глобальной сети и требуют NAT во внешние адреса для возможности коммуникации
- **127.0.0.1, 127.0.0.0/8** – Loopback-адреса. Сеть, предназначенная для интерфейсов «обратной петли» - трафик наф эти адреса предназначен самому **локальному компьютеру (localhost)** и не выходит за его пределы. Служит для возможности сетевой коммуникации внутри локальной машины. Не маршрутизируема
- **169.254.0.0/16** – сеть, зарезервированная для **автоматической приватной адресации (APIPA, Automatic Private IP Addressing).** Может назначаться автоматически при недоступности DHCP-сервера и позволяет функционировать в пределах broadcast-домена. Не маршрутизируема
- **192.0.2.0/24** – сеть, зарезервированная для использования в примерах и документации
- **224.0.0.0/4** – суперсеть класса D – для многоадресной (multicast) передачи. Могут быть *только адресами получателя*
- **240.0.0.0/4** – сеть класса E. Не должны быть использованы, потому как «зарезервированы для будущего использования» (спойлер – «будущее» наступило слишком поздно, эти адреса невозможно корректно использовать, они бездарно и трагически похоронены…)
- **255.255.255.255** – широковещательный адрес для доставки пакетов всем хостам одного broadcast-домена

NAT Overload (Port Address Translation, PAT)



- **Source NAT** – трансляция IP-адреса/порта отправителя при прохождении пакета из внутренней сети (nat inside) во внешнюю (nat outside)
- **Destination NAT** - трансляция IP-адреса/порта получателя при прохождении пакета из внешней сети (nat outside) во внутреннюю (nat inside)
- **Port Forwarding (проброс портов)** – возможность трансляции номеров портов назначения при прохождении пакетов через NAT. Например, для проброса трафика, поступающего на определённый порт внешнего интерфейса роутера на какой-либо порт какой-либо из машин внутри локальной сети. Настройкой проброса портов можно сделать доступными извне ресурсы внутри приватной сети

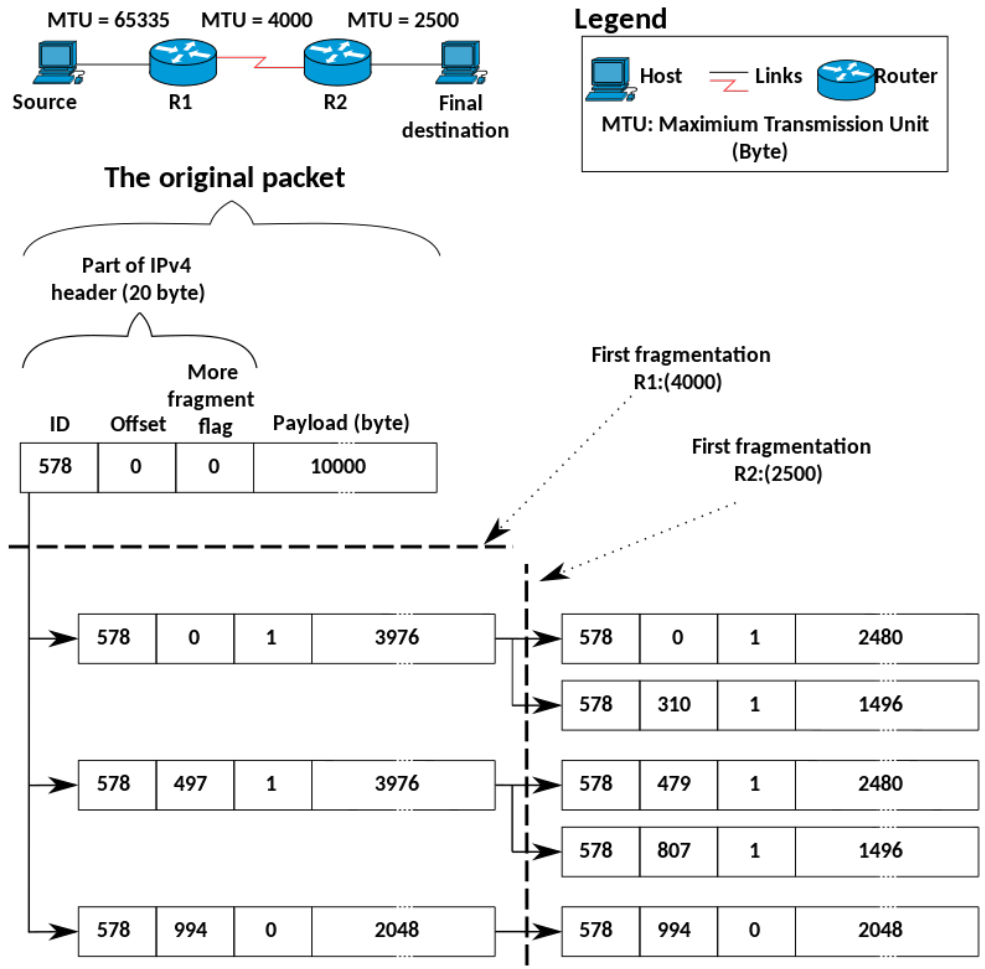
Классовая и бесклассовая адресация

- **Классовая адресация (Classful routing)** - архитектура сетевой адресации, которая делит адресное пространство протокола IPv4 на пять классов адресов: A, B, C, D и E. Принадлежность адреса к конкретному классу задаётся первыми битами адреса. Каждый класс определяет либо соответствующий размер сети, то есть количество возможных адресов хостов внутри данной сети. Использовалась в Интернете в период с 1981 по 1993 годы, до введения **CIDR**
- **CIDR - (*Classless InterDomain Routing,* бесклассовая междоменная маршрутизация)** — метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Использование этого метода позволяет экономно использовать ограниченный ресурс IP-адресов, поскольку возможно применение различных масок подсетей к различным подсетям
- Бесклассовая адресация основывается на **переменной длине маски подсети (*variable length subnet mask, VLSM*)**, при котором маска подсети может быть любой длины от 0 до 32 бит, в отличие от классовой адресации, при которой маска имела фиксированное значение в 8, 16 или 24 бит в зависимости от класса сети (A, B, C, D, E)

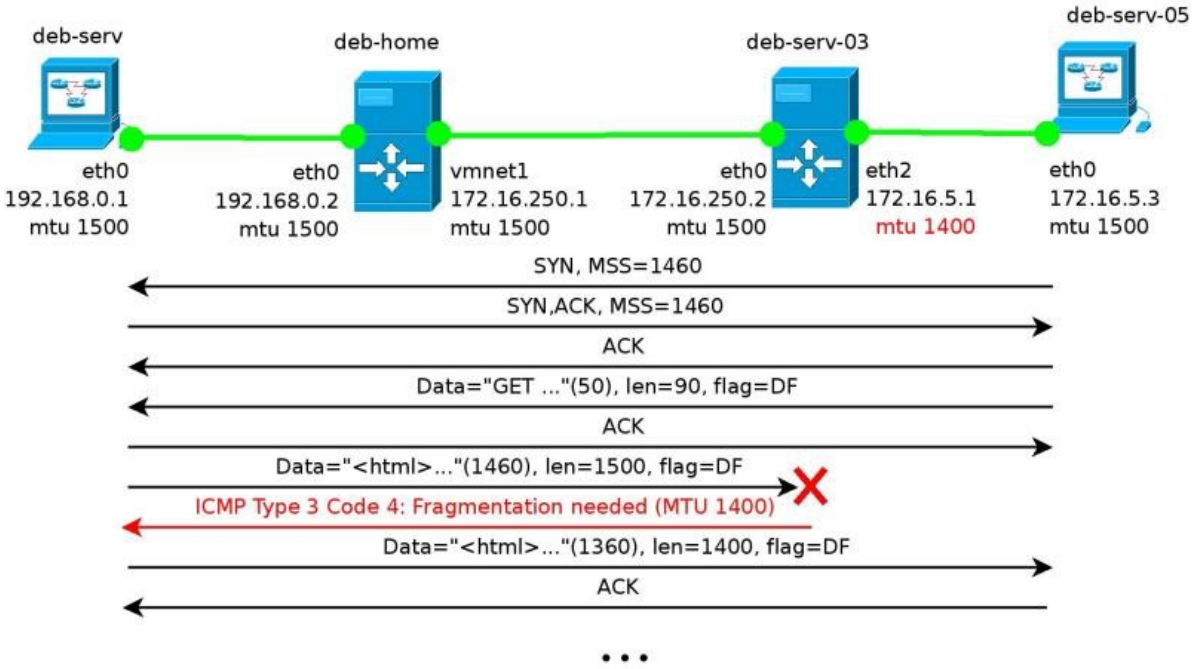
Размер пакета и фрагментация

- **MTU (Maximum Transmission Unit, максимальная единица передачи) –** максимальное количество байт (включая заголовки всех протоколов), которое физический интерфейс может передать за один раз. К примеру, для Ethernet это 1500 байт (1514 байт с заголовком самого Ethernet). Пакеты большего размера, чем MTU сетевого интерфейса, не могут быть переданы через данный интерфейс и отбрасываются. Может снижаться при использовании туннелей или какой-либо дополнительной инкапсуляции (VLAN, к примеру) на размер соответствующих заголовков
- **TCP MSS (Maximum Segment Size, максимальный размер сегмента) –** размер **полезной нагрузки (payload)** транспортного протокола, который можно упаковать в один пакет при заданном MTU. TCP MSS = Interface MTU – 40 байт (20 байт заголовок IP + 20 байт заголовок TCP). Не включает в себя заголовки вышестоящих уровней. При установлении TCP сессии каждая сторона отправляет своё значение MSS, и обе стороны обязаны использовать меньшее из двух значений. UDP не выполняет сегментацию и не имеет понятия MSS. Сегментацию и вычисление размера в случае UDP должно выполнять само приложение
- **IP фрагментация** — это процесс разбиения оригинального IP-пакета на множество фрагментов с добавлением IP-заголовка к каждому фрагменту для независимой доставки. Информация из IP-заголовка также используется для сборки воедино оригинального пакета на конечном или промежуточных узлах. Каждый фрагмент может быть повторно фрагментирован на транзитных узлах
- **DF bit (Do Not Fragment bit)** – флаг в заголовке IP, запрещающий фрагментацию пакета. Если транзитный узел не может передать далее пакет с флагом DF из-за размера, превышающего MTU, такой пакет будет отброшен, а отправителю будет отправлено ICMP сообщение Fragmentation Needed с указанием нужного размера MTU
- **MF bit (More Fragments bit)** – флаг в заголовке IP, сообщающий о наличии ещё фрагментов. Отсутствует у последнего фрагмента
- **Смещение фрагмента (Fragment Offset field) –** поле в заголовке IP, обозначающее позицию данных в текущем фрагменте относительно данных в исходном пакете. Используется при сборке оригинального пакета из фрагментов
- **PMTU Discovery (Path MTU Discovery)** – техника определения минимально допустимого MTU на всём пути следования пакета. При передаче данных в пакете выставляется DF bit, а при получении сообщения ICMP Fragmentation Needed значение MSS уменьшается на соответствующую величину. Несмотря на то, что UDP не имеет понятия MSS, приложения на основе UDP могут внутри себя реализовывать эту логику своими силами

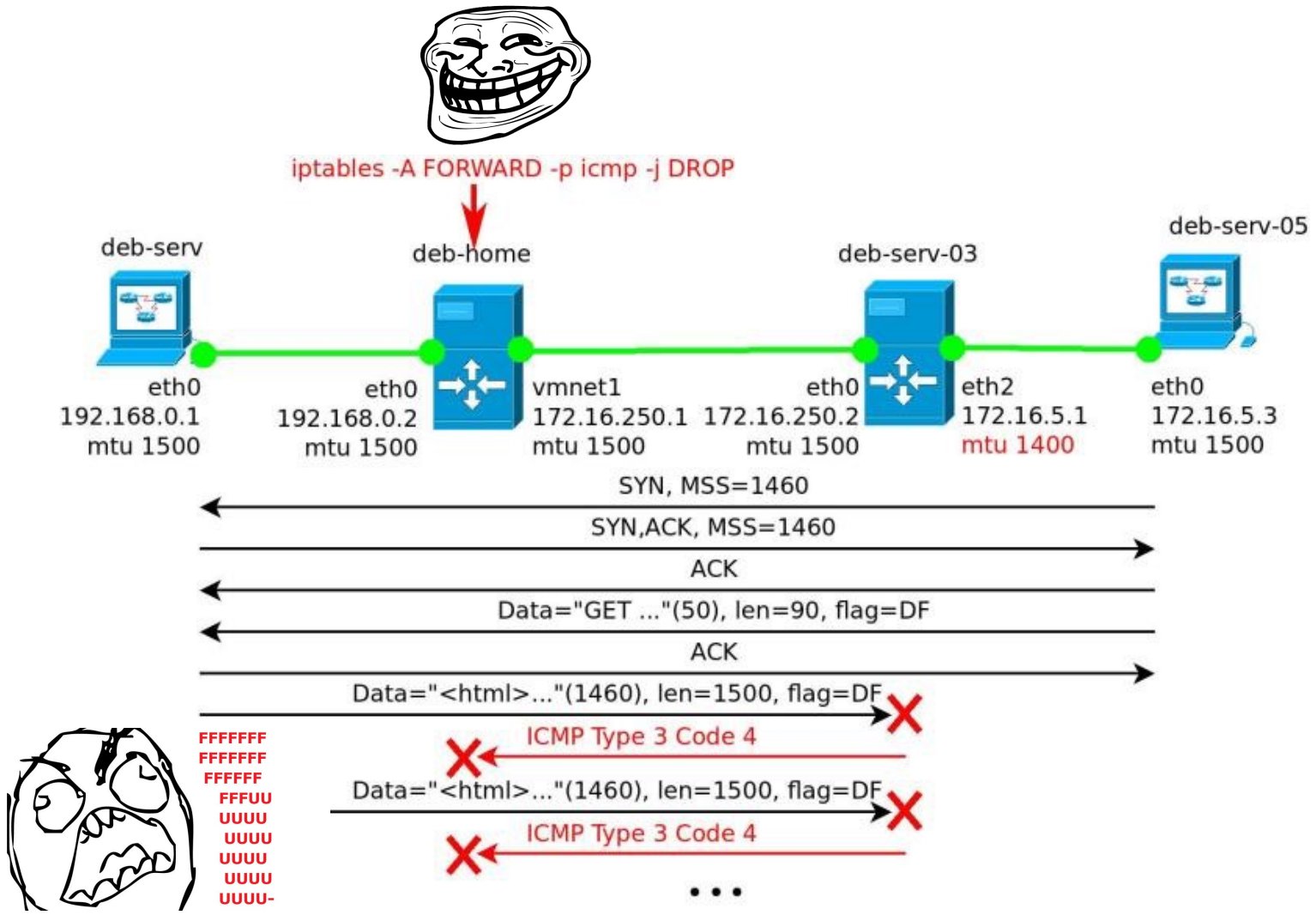
Фрагментация IP-пакета



Path MTU Discovery



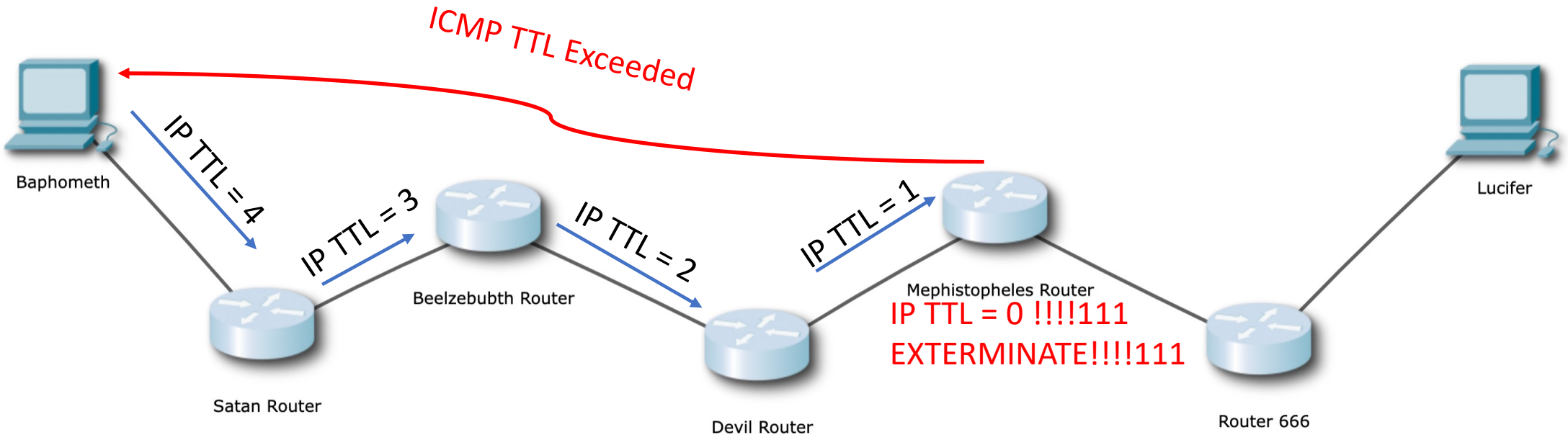
Проблема PMTU Black Hole



Видимые в web-браузере симптомы:

- Очень лёгкие сайты (как ya.ru) открываются корректно, а тяжёлые (как yandex.ru) – не открываются или открываются с искажениями
- Браузер бесконечно пытается догрузить данные, и в полузагруженном состоянии страница висит бесконечно долгое время
- Пинги и трассировка до данного сайта проходят тем не менее нормально
- Попытка сделать telnet на порт 80 тоже проходит успешно

Предотвращение петель в IP-сети

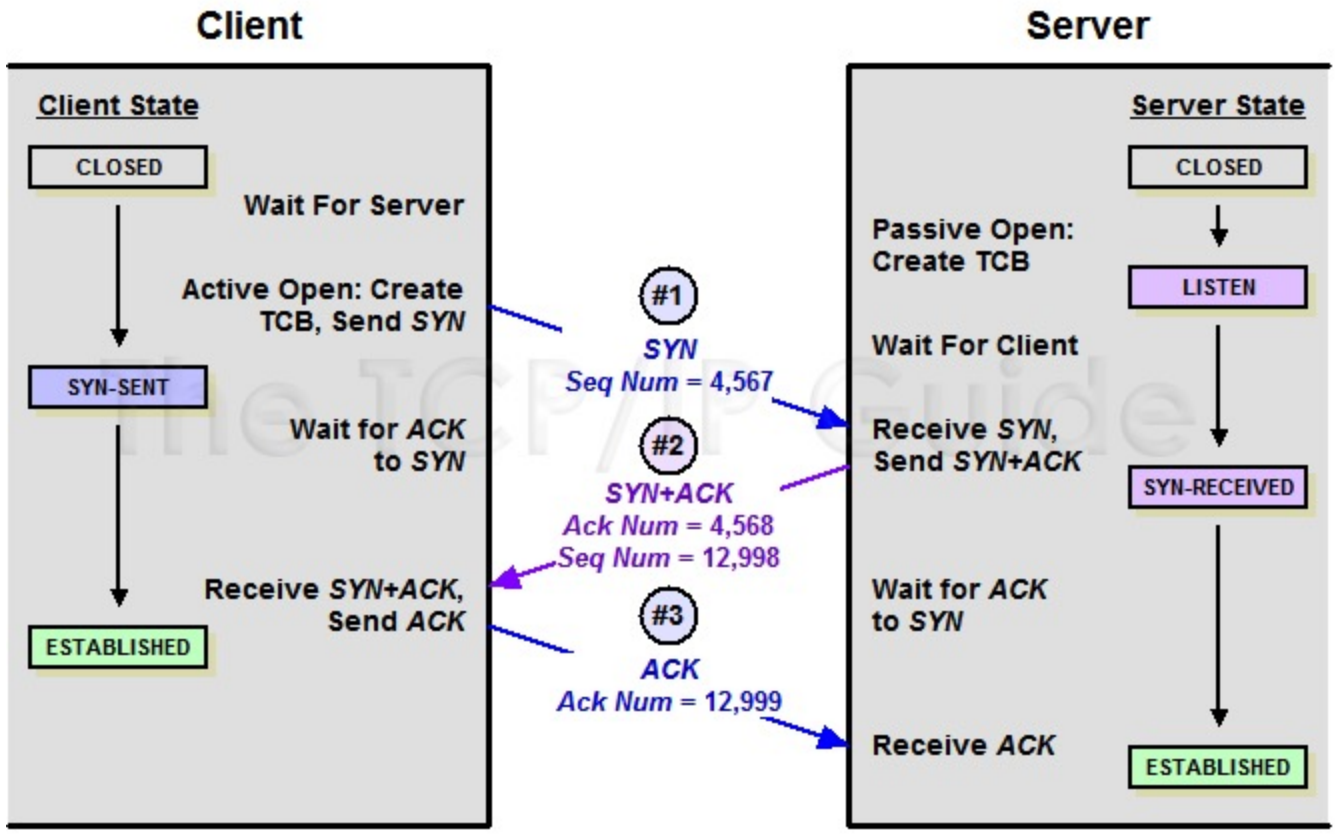


- Отправитель отправляет пакет с некоторым начальным значением IP TTL (зависит от приложения или от значений по умолчанию в операционной системе. Обычно TTL=64 или 128
- При прохождении пакета каждый маршрутизатор по пути перед отправкой уменьшает значение TTL на 1
- Если значение TTL стало равным нулю, пакет уничтожается, а отправителю отправляется ICMP-сообщение “TTL Exceeded”
- Таким образом предотвращается бесконечное циркулирование в сети пакетов из-за проблем с маршрутизацией и некорректных настроек роутеров

Transmission Control Protocol (TCP)

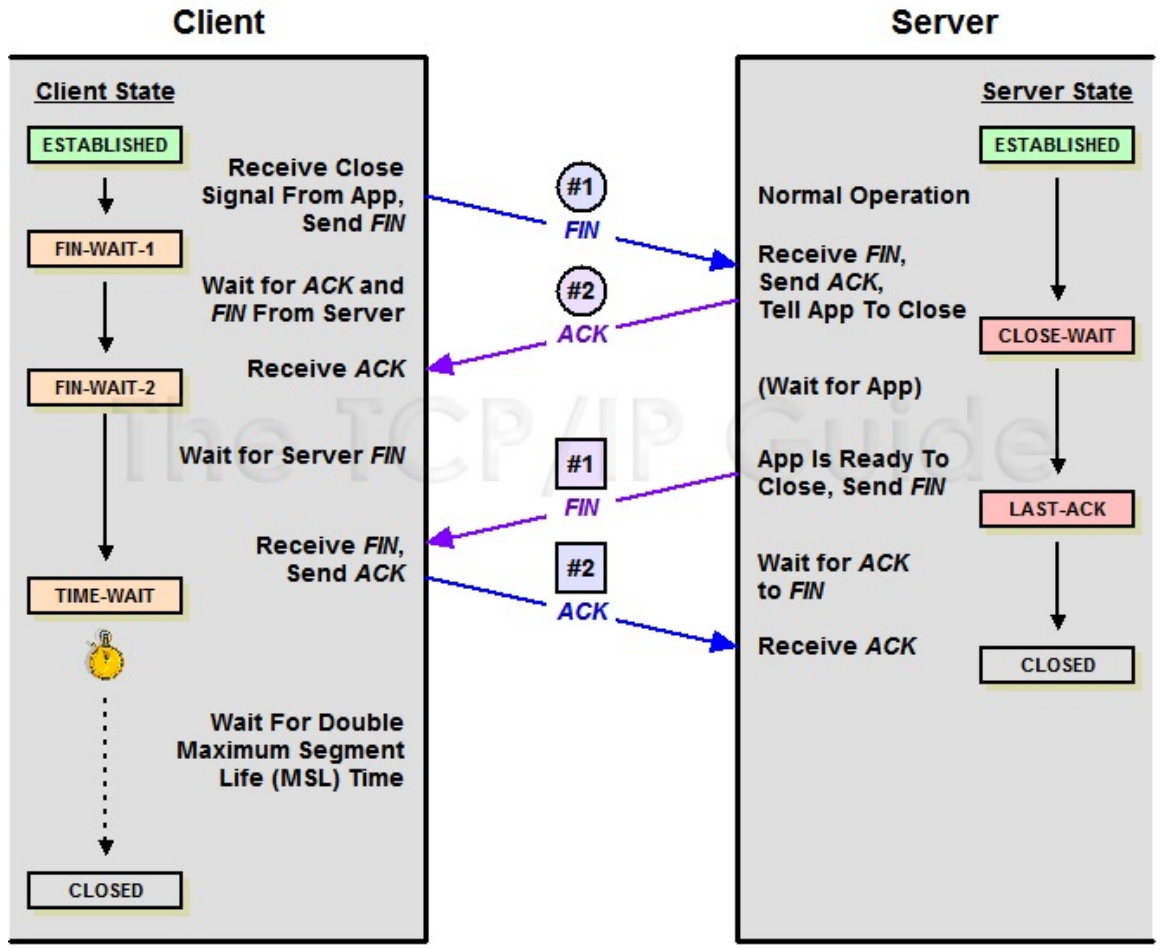
- Протокол транспортного уровня модели OSI
- Осуществляет передачу данных с предварительной установкой логического соединения
- Осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета
- Гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи

Установка TCP-соединения



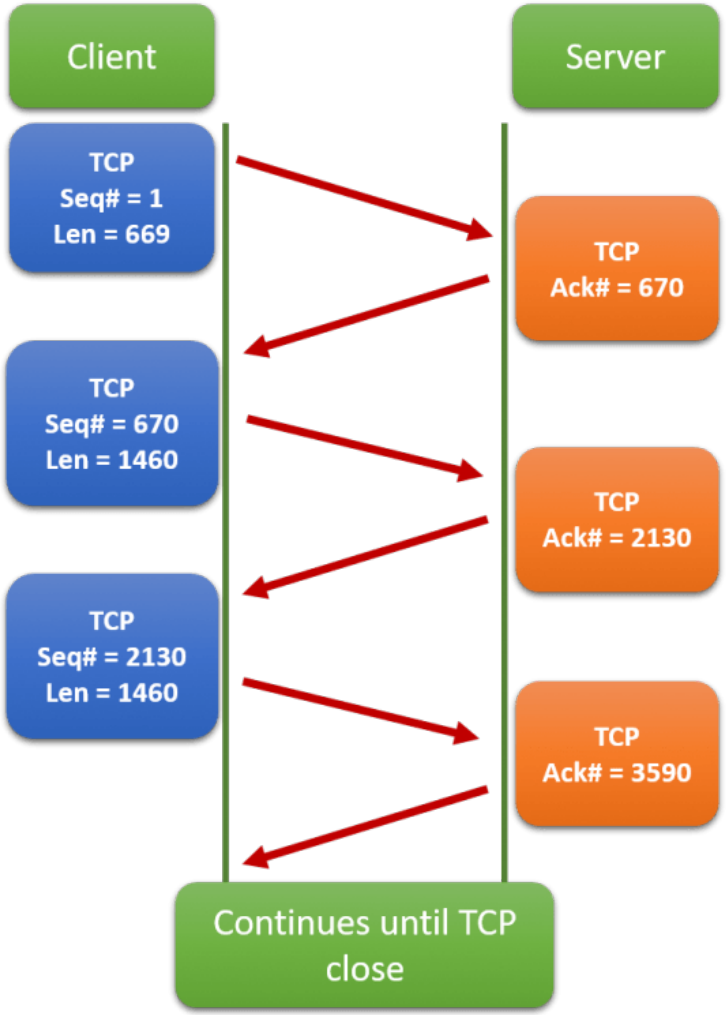
- Также называется **трёхсторонним рукопожатием (3-way handshake)**
- Клиент отправляет пакет с установленным флагом SYN
- При получении SYN-пакета сервер в ответ отправляет пакет с установленным флагом SYN, а также с установленным флагом ACK
- В ответ на пакет SYN/ACK от сервера клиент отправляет пакет с установленным флагом ACK соединение на стороне клиента переходит в состояние ”Установлено”
- После получения сервером сообщения ACK от клиента соединение на стороне сервера переходит в состояние ”Установлено”
- Можно начинать обмен полезными данными

Завершение TCP-соединения



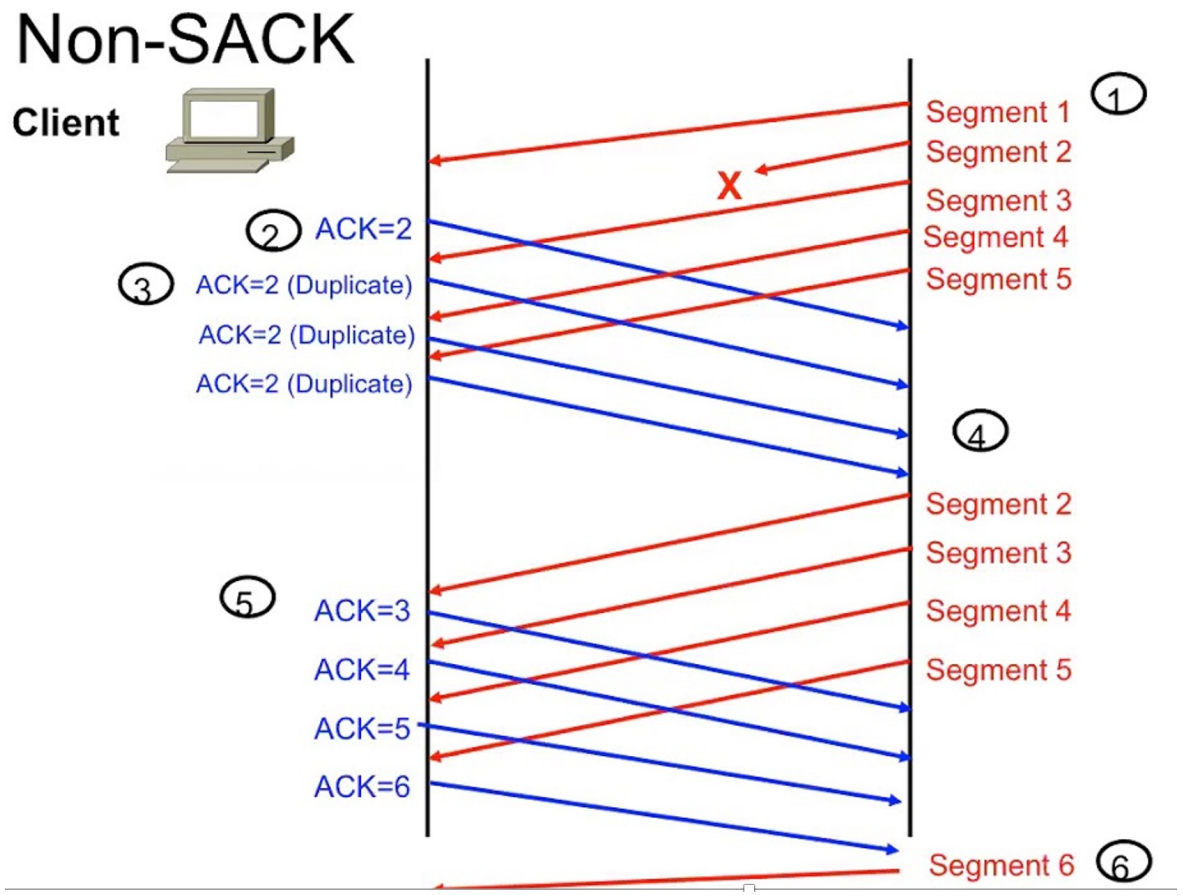
- Когда у одной из сторон стороны больше нет данных для отправки в потоке, она инициирует разрыв соединения, отправляя сегмент с установленным флагом FIN
- Удалённая сторона подтверждает получение FIN-пакета сообщением ACK. В этот момент соединение переходит в «полузакрытое» состояние. От удалённой стороны ещё могут приходить ACK-и
- Когда удалённая сторона отправит все подтверждения предыдущих пакетов, она также отправит другой стороне FIN
- Инициатор разрыва, получив FIN от удалённой стороны, отправляет в ответ ACK. С этого момента соединение на стороне инициатора закрыто (closed)
- Удалённая сторона, получив подтверждение на свой FIN, также закрывает соединение на своей стороне

Подтверждение передачи



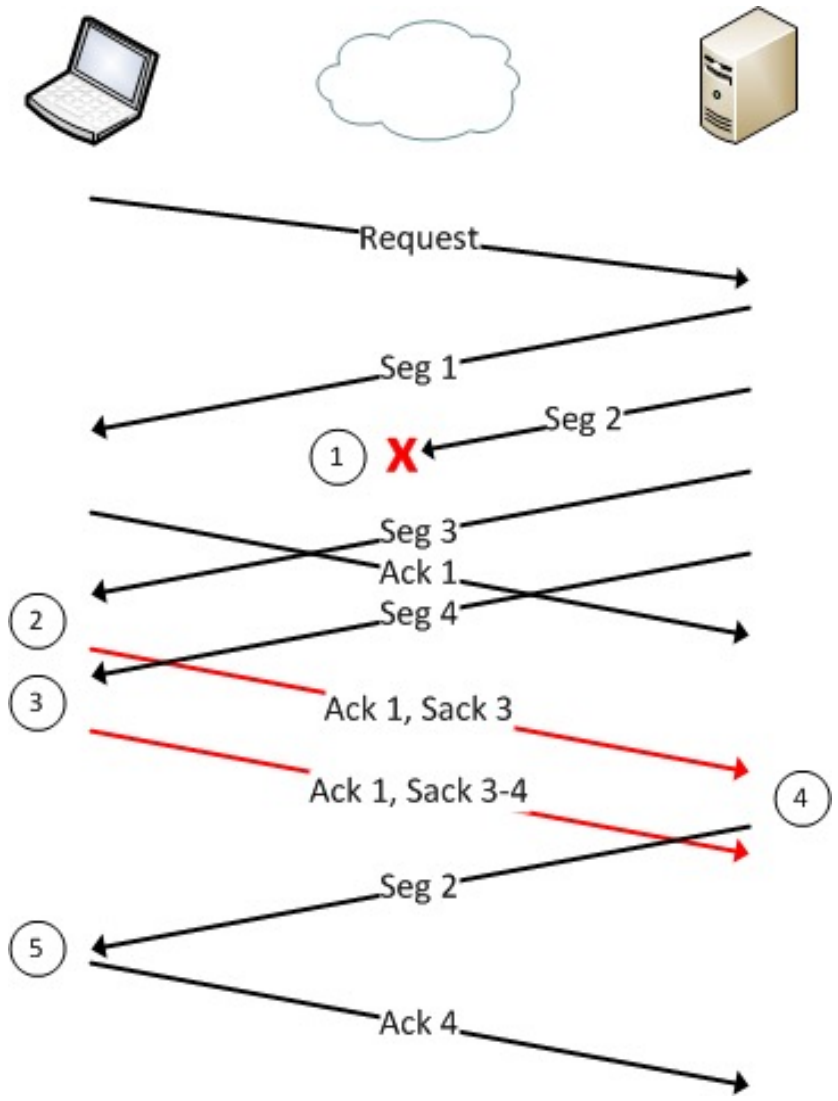
- Передающая сторона посылает сегмент с некоторым начальным значением SN (Initial Sequence Number) и длиной сегмента в байтах (Len)
- На каждый переданный сегмент удалённая сторона посылает сегмент с флагом ACK и значением ACK SN = SN + 1, говоря о том, начиная с какого байта должна быть передана следующая порция данных, тем самым подтверждая успешное получение предыдущего сегмента
- Следующий переданный сегмент будет иметь SEQ = last ACK SN и новое значение длины
- Удалённая сторона после приёма сегмента отправит подтверждение уже для нового SEQ + 1
- Так стороны будут обмениваться данными и подтверждениями пакетов вплоть до закрытия соединения
- Таким образом, номер последовательности (SN) отражает порядковый номер байта, с которого начинается передача, а номер подтверждения (ASN) – номер следующего байта, который удалённая сторона хочет получить
- Наличие SN также позволяет протоколу TCP корректно собирать обратно исходные данные из пакетов – даже если сегменты дошли не в том порядке, в котором были отправлены (out-of-order)

Повторная передача



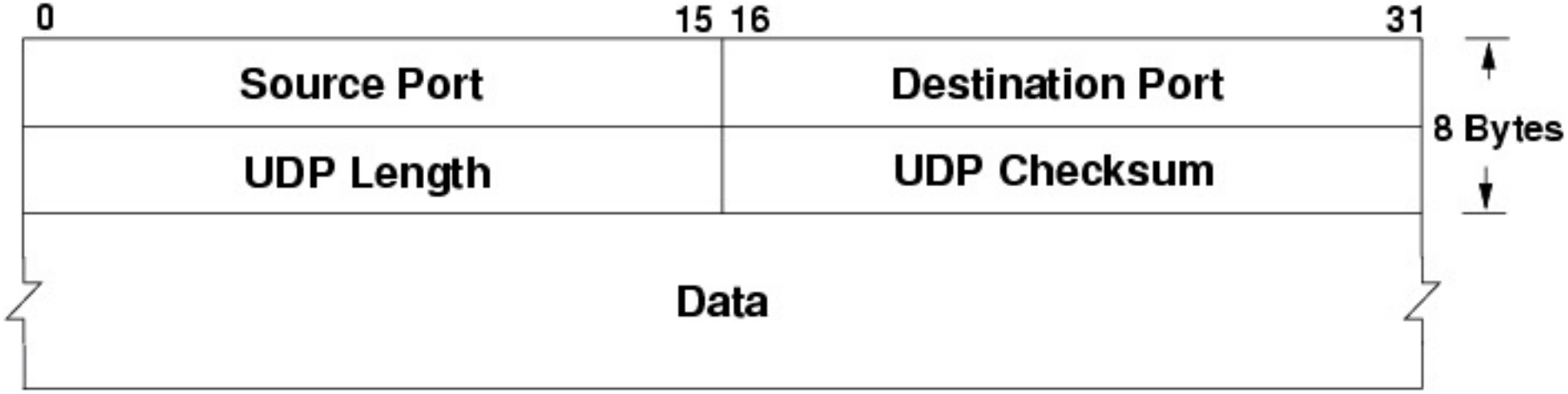
- В случае утери какого-либо из сегментов получающая сторона будет на каждый приходящий сегмент отсылать дубликат последнего отправленного ACK (с ASN, соответствующим началу утерянного сегмента)
- Таким образом, передающая сторона вынуждена передавать все данные начиная с утерянных
- Утеря данных может быть обнаружена как путём получения дублирующихся ACK, так и по таймауту получения ACK

Selective ACK (SACK)



- Selective ACK сохраняет в поле TCP Options информацию об уже принятых сегментах, что позволяет идентифицировать конкретный утерянный сегмент
- SACK является необязательным механизмом и должен согласовываться при установлении соединения
- Описан в RFC 2018
- Позволяет избежать лишних перепосылок данных, которые уже были приняты

Протокол UDP



- UDP — минимальный ориентированный на обработку сообщений протокол транспортного уровня
- UDP не предоставляет никаких гарантий доставки сообщения для вышестоящего протокола и не сохраняет состояния или очерёдности отправленных сообщений
- UDP обеспечивает механизм портов и проверку целостности заголовка и существенных данных с помощью контрольных сумм. Надёжная передача в случае необходимости должна реализовываться пользовательским приложением
- UDP не имеет средств контроля перегрузок. Приложения с большой пропускной способностью могут вызвать коллапс, если только они не реализуют меры контроля на прикладном уровне.

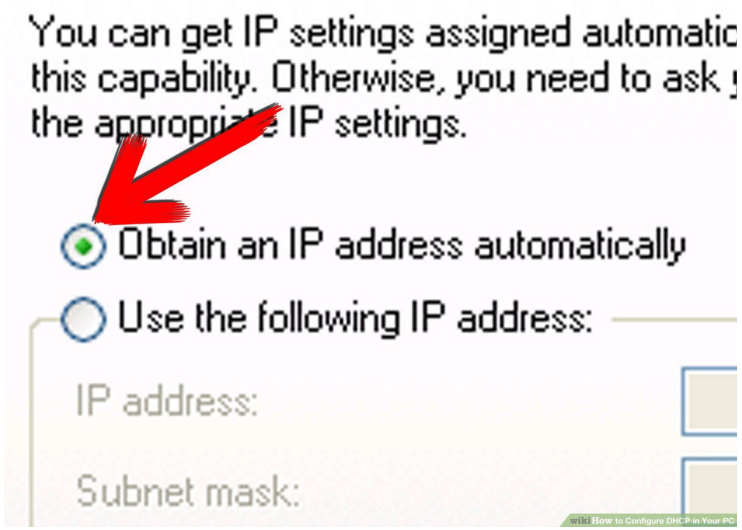
Области применения UDP

- За счёт отсутствия установления соединения UDP быстрее приступает к передаче полезных данных
- UDP проще реализуем как протокол, имеет меньший overhead по служебным заголовкам (8 байт UDP против 20 байт TCP) и меньшую вычислительную сложность, а реализации его занимают мало места, что упрощает его применение в крайне ограниченных в ресурсах системах (таких как микроконтроллеры)
- UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP там, где предпочтительнее потерять пакеты, чем ждать задержавшиеся (пример – голосовой или стриминговый трафик)
- На базе UDP можно строить надёжную доставку данных, если контроль доставки становится ответственностью приложения. Такой выбор делается тогда, когда нужны не все функции TCP – например, как сегментация и контроль очерёдности доставки, если нужно надёжно передать малое количество данных
- За счёт отсутствия концепции соединения, предполагающей детерминированного получателя, UDP подходит как транспорт для **многоадресной (multicast)** и **широковещательной (broadcast)** передачи

Размер окна и контроль перегрузок. Медленный старт (slow start)

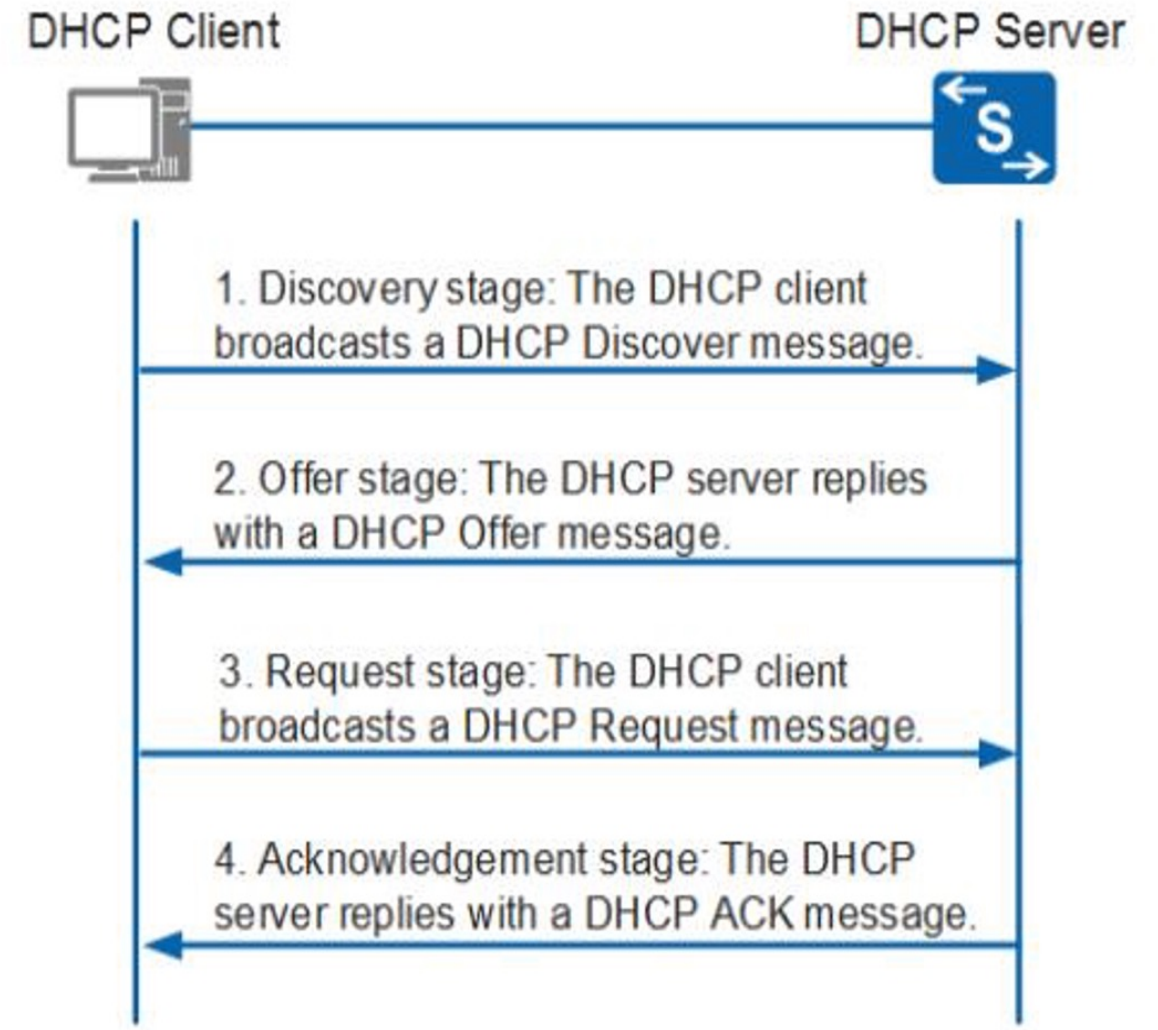
- **Размер TCP окна (TCP Window Size, Receiver Window Size, RWND)** – количество байт, которое принимающая сторона готова принять в настоящий момент без подтверждения. На стадии установления соединения рабочая станция и сервер обмениваются значениями максимального размера TCP окна. Передаётся в TCP-заголовке.
- **Размер окна перегрузки (Congestion Window Size, CWND)** – количество байт, которое готова передавать передающая сторона
- **«Скользящее окно»** - механизм, позволяющий отправлять некоторое количество пакетов **(окно перегрузки, congestion window)** не дожидаясь прихода ACK
- **RTT (Round Trip Time)** – время, за которое посылается пакет к получателю и приходит отправителю на него ответ
- **Медленный старт (slow start) –** механизм оптимальной утилизации канала, заключающийся в экспоненциальном увеличении размеров окна перегрузки с 1 MSS (maximum segment size) в 2 раза с каждым принятым ACK (сначала до 2xMSS, потом до 4 и т.д.) вплоть до значений RWND
- При обнаружении признаков перегрузок (потеря пакетов, duplicate ACKs, таймауты) **механизм slow start выключается, размер CWND снижается в 2 раза и начинает расти вновь очень медленно** – не экспоненциально, а увеличиваясь на единицу с каждым RTT

Dynamic Host Configuration Protocol (DHCP)



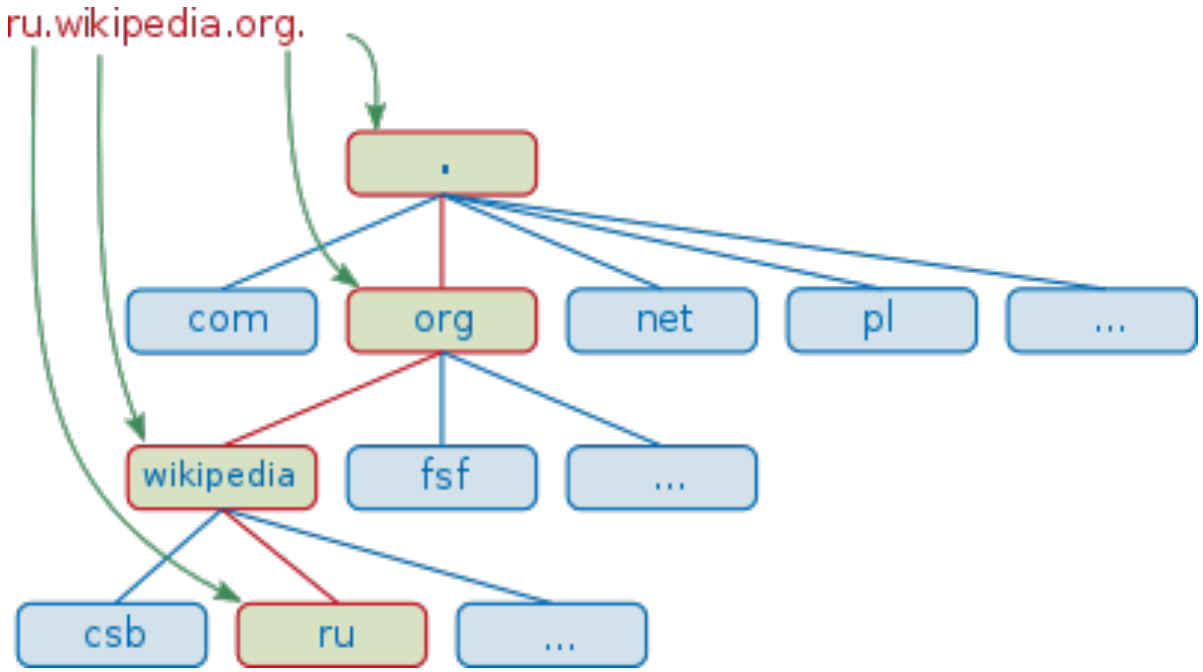
- Клиент-серверный протокол прикладного уровня, позволяющий автоматически назначать настройки IP (адреса, маску подсети, основной шлюз, DNS-сервера) устройствам в локальной сети
- Также может передавать в виде DHCP Options и другие различные параметры (NTP-сервера, размер MTU, имя хоста и т.д.)
- Описан в RFC2131 (версия для IPv6 описана в RFC8415)
- За DHCP-сервером зарезервирован UDP-порт 68, а за DHCP-клиентом UDP-порт 67

Принцип работы протокола DHCP



- Клиент посылает broadcast-запрос **DHCPDISCOVER** со своим MAC-адресом на UDP-порт 67 с портом отправителя 68
- DHCP-сервер, получив запрос от клиента, резервирует IP-адрес из своего пула и отправляет его клиенту в сообщении **DHCPOFFER**
- DHCP-клиент подтверждает предложенный IP-адрес сообщением **DHCPREQUEST**, проверив наличие такого адреса в сети ARP-запросом. Если хост с таким адресом уже имеется в сети, хост обязан ответить **DHCPDECLINE**
- DHCP-сервер подтверждает выдачу адреса сообщением **DHCPACK** и сохраняет статус адреса в **базе данных аренды (lease db)**
- Сообщения **DHCPOFFER/DHCPACK** в сторону клиента могут быть как в виде unicast, так и broadcast
- Аренда адреса происходит на определённый срок – **время аренды (lease time)**. Впоследствии получивший адрес клиент периодически обновляет аренду сообщением DHCPREQUEST без предварительного DHCPDISCOVER

Domain Name System (DNS)

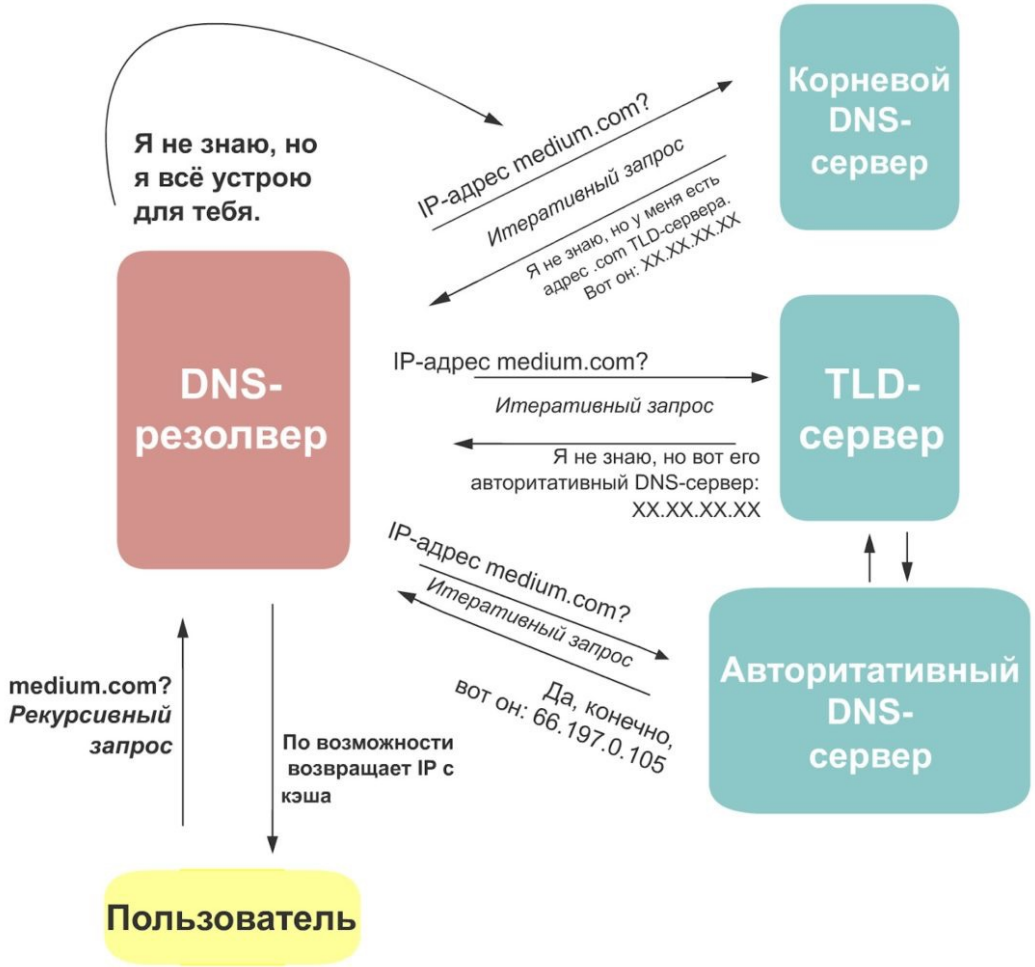


- **DNS** – распределённая иерархическая клиент-серверная система для получения информации о **доменах** и протокол для доступа к ней
- Чаще всего используется для получения IP-адреса по доменному имени хоста, но также может хранить получения информации о маршрутизации почты или обслуживающих узлах для протоколов в **домене**
- **Доменное имя** — символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети Интернет, в составе вышестоящей по иерархии такой области
- DNS представляет возможность доступа к ресурсам в IP-сетях (и в сети Интернет в частности) по человеко-читаемым именам
- За DNS зарезервированы UDP и TCP-порт 53
- Описан в RFC 1034 и RFC 1035

Основные понятия DNS

- **Домен** - узел в дереве имён, вместе со всеми подчинёнными ему узлами (если таковые имеются), то есть именованная *ветвь* или *поддерево* в дереве имён. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего уровня (пример – host1.example.com)
- **Доменная зона** – это часть пространства имен, которая содержит адреса конкретных доменов. Пример – для host1.example.com доменной зоной будет example.com
- **Ресурсная запись** - единица хранения и передачи информации внутри зоны DNS. Каждая ресурсная запись имеет *имя* , *тип* и *поле данных*, формат и содержание которого зависит от *типа*.
- **Домен верхнего уровня (top-level domain, TLD)** – верхний уровень иерархии доменных имён. Является начальной точкой отсчёта (справа налево), с которой начинается доменное имя в Интернете. Примеры - **домены по назначению (.com, .net, .org)** или **географические (.ua, .ge, .fi)**
- **Корневая зона или зона «.»** - зона, содержащая информацию обо всех TLD
- **Корневой сервер DNS** – группа из нескольких географически разнесённых DNS-серверов, обслуживающих корневую зону
- **DNS-сервер** – ПО и/или узел сети, отвечающий на DNS-запросы. Может отвечать на запросы сам, предоставляя информацию из своей базы, или перенаправлять запросы к другим DNS-серверам
- **DNS-клиент или DNS-резолвер** – программное обеспечение, обращающееся к DNS-серверам за какой-либо информацией
- **Авторитативный DNS-сервер** – сервер, содержащий имена узлов и поддомены конкретной DNS-зоны
- **Авторитативный ответ** – ответ от DNS-сервера, на котором есть полная информация о зоне домена
- **Неавторитативный ответ** – ответ от сервера, не являющегося владельцем зоны (как правило, от кэширующего сервера - сервера, сохраняющего прошлые ответы в течение некоторого времени)

Рекурсивный и итеративный DNS-запросы



- **Рекурсивный запрос** – это запрос на выполнение полного преобразования имени FQDN в адрес IP. Если сервер DNS имеет информацию о записи, он ответит клиенту, завершив запрос ответом. Если сервер DNS не знает ответа, он может выполнить несколько итеративных запросов к **корневым серверам системы DNS**

- **Итеративный запрос** является запросом на преобразование только части имени FQDN. Например, если система запрашивает сервер DNS для получения адреса IP имени hello.example.com и сервер DNS не имеет информации об адресе этого узла, сначала он выполнит итеративный запрос к одному из **корневых серверов DNS** к серверу зоны com., чтобы узнать адрес сервера DNS для зоны example.com

- Авторитативный D

Основные типы ресурсных записей DNS

- **A-запись** — задает преобразование имени хоста в IP-адрес
- **AAAA-запись** — задает преобразование имени хоста в IPV6-адрес
- **MX-запись** — определяет почтовый ретранслятор для доменного имени, т.е. узел, который обработает или передаст дальше почтовые сообщения, предназначенные адресату в указанном домене
- **NS-записи** — определяют DNS-серверы, которые являются авторитативными для данной зоны.
- **CNAME-запись** — определяет отображение псевдонима в каноническое имя узла, которое потом будет разрешено согласно A- или AAAA-записи. Позволяет хостам иметь несколько имён
- **SRV-запись** — позволяет получить имя для искомой службы, а также протокол, по которому эта служба работает.
- **TXT-запись** — содержит общую текстовую информацию. Эти записи могут использоваться в любых целях, например, для указания месторасположения хоста.

Примеры разных DNS-записей

- **A- и AAAA-записи:**

```
example.com.          1044    IN      A       93.184.216.34
example.com.          1041    IN      AAAA    2606:2800:220:1:248:1893:25c8:1946
```

- **MX-записи**

```
yandex.ru.            1663    IN      MX      10 mx.yandex.ru.
```

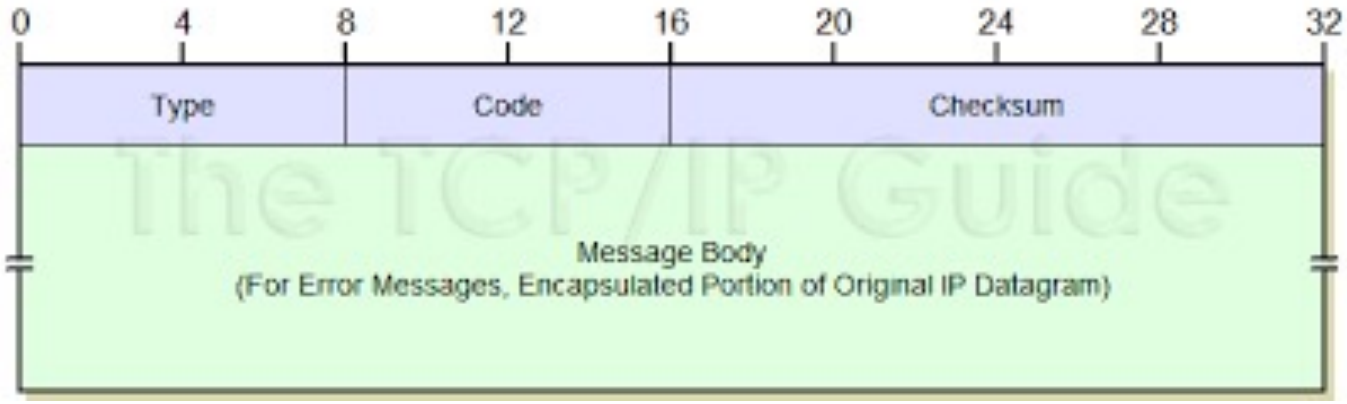
- **PTR-записи**

```
3.1.100.93.in-addr.arpa. 1773       IN      PTR     name1.sknt.ru.
```

- **CNAME-записи**

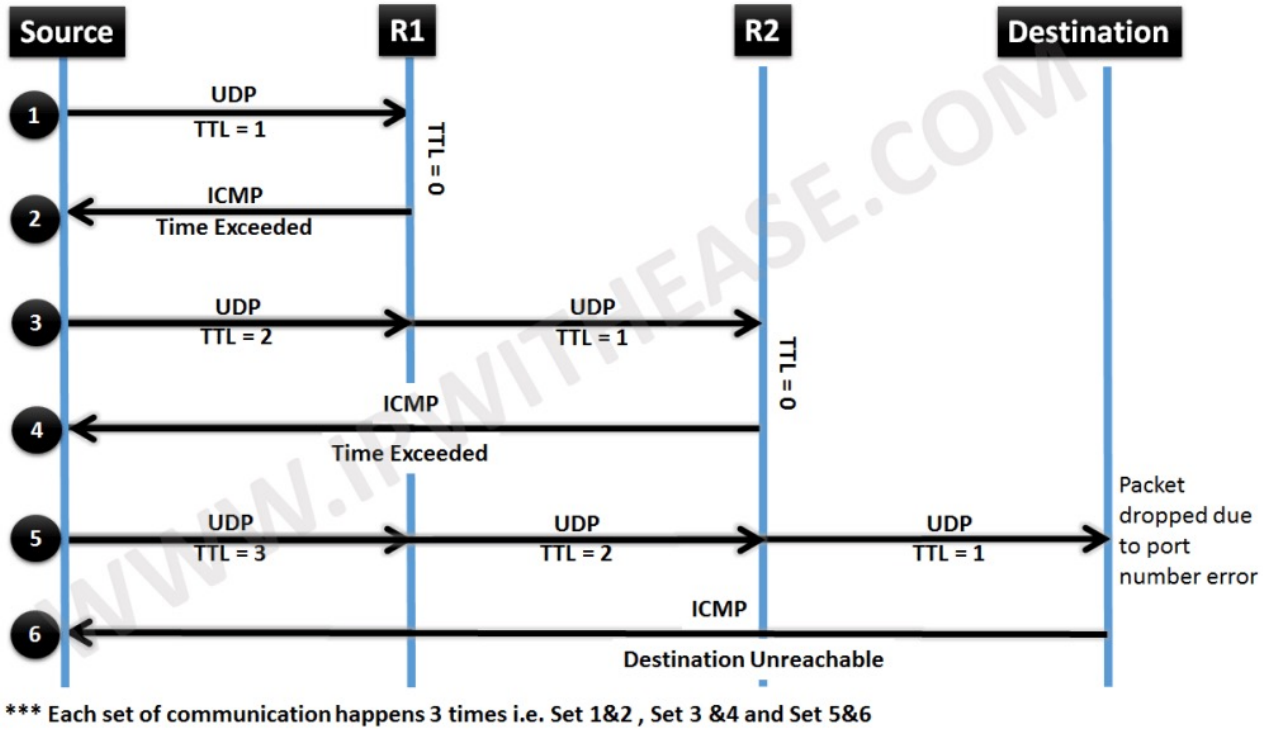
```
gmail.google.com.     1784    IN      CNAME   www3.l.google.com.
www3.l.google.com.    194     IN      A       216.58.208.110
```

Internet Control Message Protocol (ICMP)



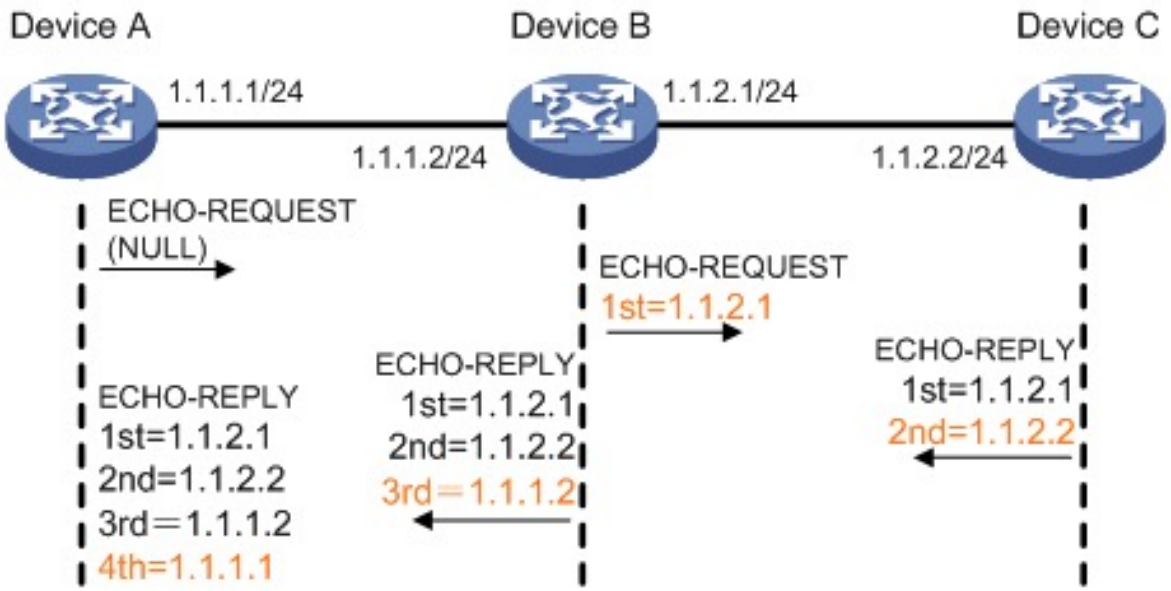
- Служебный сетевой протокол, входящий в стек протоколов TCP/IP
- В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна или хост, или маршрутизатор не отвечают
- Работает поверх протокола IP
- Описан в RFC 792

Утилита Traceroute (Tracert, Tracepath)



- Утилита Traceroute служит для воссоздания маршрута следования пакета до конечного узла
- Для определения промежуточных маршрутизаторов traceroute отправляет целевому узлу серию пакетов со значением поля TTL=1. Такие пакеты отбрасывает первый же маршрутизатор возвращая обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. На каждом шаге TTL увеличивается на единицу, пока пакет не достигнет конечного узла
- Помимо адреса маршрутизаторов, фактически отправивших ответ, утилита фиксирует также время между отправкой пакета и получением ответа
- Может работать как через UDP, так и через ICMP

Утилита Ping



- Утилита для проверки целостности и качества соединений в сетях на основе TCP/IP. Позволяет определять двусторонние задержки по маршруту, уровень потерь пакетов
- Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, от англ. Round Trip Time)
- В разговорной речи «пингом» называют также время в миллисекундах, затраченное на передачу пакета информации в компьютерных сетях от клиента к серверу и обратно от сервера к клиенту

Административное устройство Интернета

- **Internet Corporation for Assigned Names and Numbers, ICANN** – независимая международная организация, решающая общие вопросы, связанные с доменными именами. Также выполняет функции управления общим пространством IP-адресов **(IANA, Internet Assigned Numbers Authority)**
- **Региональный регистратор (Regional Internet Registry, RIR)** - организация, которой IANA делегирует вопросы адресации и маршрутизации (в т.ч. распределением блоков IP-адресов и номеров автономных систем) в Интернете в определённом регионе. На данный момент RIR'ов пять – **ARIN** (Северная Америка), **RIPE** (Европа и Ближний Восток), **APNIC** (Азия и Тихоокеанский регион), **LACNIC** (Латинская Америка), **AfriNIC** (Африка)
- **Локальный регистратор (Local Internet Registry, LIR) -** организация, занимающаяся распределением адресного пространства пользователям сетей (сервис-провайдерам и их абонентам) и оказанием сопутствующих регистрационных услуг. Как правило, локальными регистраторами управляют крупные сервис-провайдеры и корпоративные сети
- **Регистратор доменных имён (Domain Registry)** — организация, уполномоченная создавать (регистрировать) новые доменные имена и продлевать срок действия уже существующих доменных имён в домене, для которого установлена обязательная регистрация – в основном, для поддоменов верхнего уровня. Регистрацией новых TLD в корневой зоне занимается ICANN
- **Автономная система (Autonomous System, AS)** - система IP-сетей и маршрутизаторов, находящихся под единым административным управлением и имеющих единую политику маршрутизации с Интернетом. Как правило, это интернет-провайдеры или организации, обладающие собственными блоками IP-адресов. Автономные системы имеют номер, получаемый от RIR или от LIR (к примеру AS28751 – Magticom). Номер AS имеет технический смысл – он используется и передаётся в протоколе внешней маршрутизации BGP

Основные сетевые утилиты

- **ping** – утилита для тестирования доступности узла и качества канала
- **traceroute и tracert** – служит для построения маршрута прохождения пакетов до конечного узла
- **nslookup, dig** – служат для запросов к DNS
- **netstat и ss** - утилиты командной строки, выводящие состояние TCP- и UDP-сокетов (как входящих, так и исходящих), таблицы маршрутизации, сетевую статистику по протоколам, и прочую другую статистику
- **whois** – утилита и одноимённый протокол для получения регистрационных и других данных о владельцах доменных имён, IP-адресов и автономных систем из публичных серверов (к примеру, whois-серверов регистраторов)
- **arp** – просмотр и манипуляции с arp-таблицей
- **route** – просмотр и манипуляции с таблицей маршрутизации
- **ip** (из комплекта iproute2) – комбинированная утилита, заменяет утилиты arp, route и ряд других

Протокол IPv6

- **IPv6** (*Internet Protocol version 6*) — новая версия протокола IP призванная решить проблемы, с которыми столкнулась версия IPv4 при её использовании в Интернете, за счёт целого ряда принципиальных изменений. Протокол был разработан IETF и описан в RFC 8200
- Длина адреса IPv6 составляет 128 бит, в отличие от адреса IPv4, длина которого равна 32 битам и может обеспечить до $5 \cdot 10^{28}$ адресов на каждого жителя Земли
- Убран ряд механизмов, затрудняющих работу маршрутизаторов

Формат адреса IPv6

2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d/64

2001::1/64

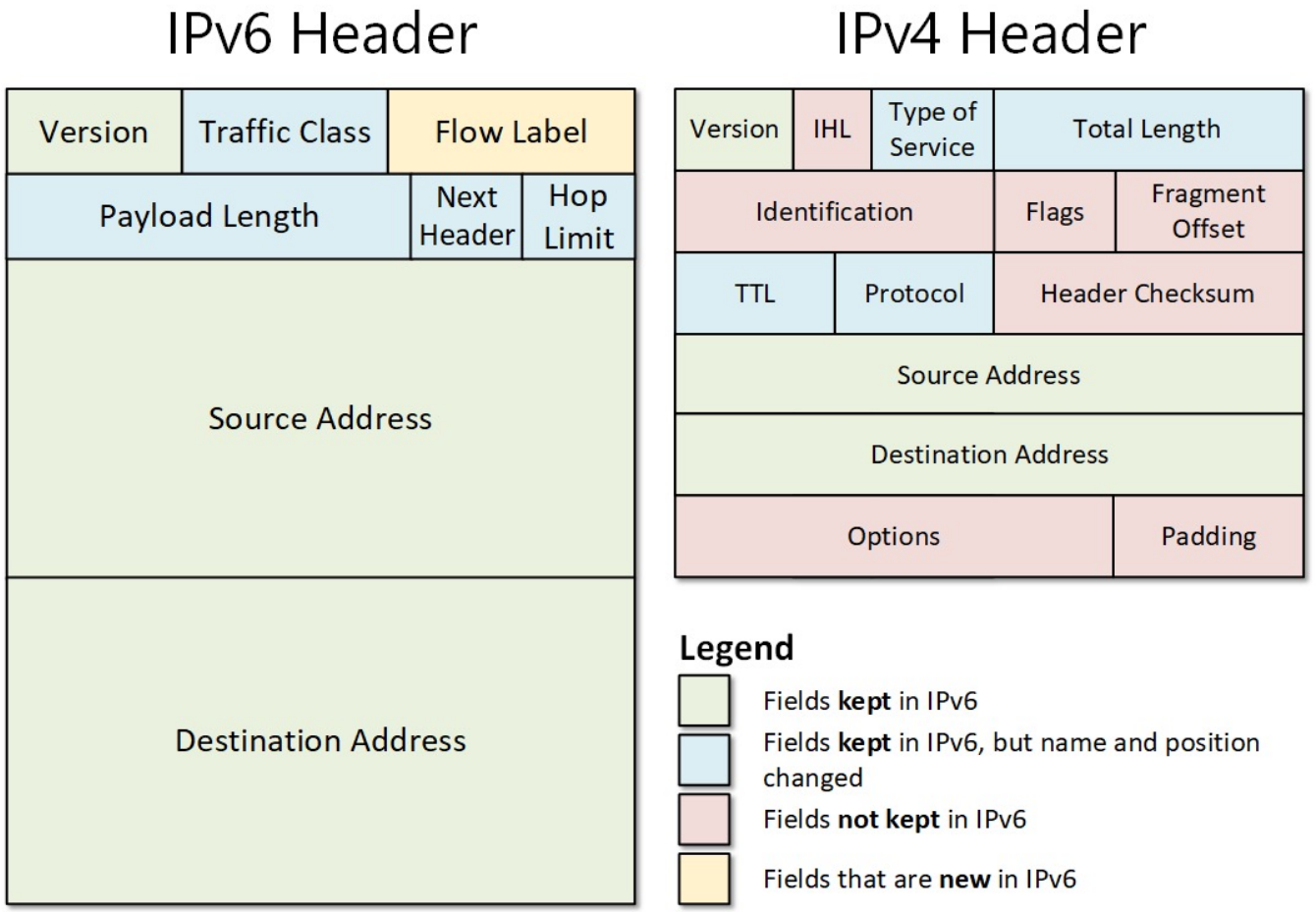
2001:0:0:1::/64

- Адреса IPv6 отображаются как восемь четырёхзначных шестнадцатеричных чисел (то есть групп по четыре символа), разделённых двоеточием
- Если две и более групп подряд равны 0000, то они могут быть опущены и заменены на двойное двоеточие (::). Незначащие старшие нули в группах могут быть опущены. Сокращению не могут быть подвергнуты 2 разделённые нулевые группы из-за возникновения неоднозначности
- Маска подсети представляется только в виде префикса CIDR

Изменения в протоколе IPv6

- Длина адреса увеличена до 16 байт, что увеличивает пространство адресов с 4млрд адресов до 5 x 10 ^ 28 (около 79 228 162 514 264 337 593 543 950 336 октиллионов - не считая служебных). Это означает, что протокол обеспечит возможность использования более 300 млн IP-**адресов** на каждого жителя Земли, что убирает необходимость в NAT (хотя поддержка осталась)
- Исчезло понятие broadcast-трафика. Протокол IPv6 полагается на multicast-трафик и активно его использует. Протокол ARP в связи с этим заменён механизмом **Neighbor Discovery**, работающим через multicast. Нет выделенных адресов network и broadcast, первый и последний адреса являются валидными адресами хостов
- Появилась возможность полноценной автоконфигурации хостов без наличия в сети DHCP-сервера (**Stateless Address Auto-configuration, SLAAC)** с выдачей валидных IPv6-адресов в подсети, соответствующих mac-адресу устройства **(eui-64 address)**, и обнаружения маршрутизаторов в сети с помощью механизмов Router Solicit/Router Advertisement с передачей доп. параметров (например, dns-серверов)
- Размер минимально маршрутизируемой сети *настоятельно рекомендовано* делать /64 (18,446,744,073,709,551,616 адресов) для возможности автоконфигурации через eui-64

Заголовок IPv6



- Адреса увеличились с 4 байт до 16 (128 бит)
- Исчезло поле Header Checksum во избежание пересчёта контрольной суммы на каждом промежуточном узле из-за изменения TTL. Целостность пакета предлагается проверять на канальном уровне
- TTL переименовано в Hop Limit для лучшего отражения смысла поля (оно отражает не фактическое время жизни пакета, а количество промежуточных узлов, через которые пакет проходит
- Были убраны поля Identification, Flags и Fragment Offset, поскольку IPv6 не поддерживает фрагментацию на транзитных узлах и требует либо применения PMTU Discovery, либо использования минимально рекомендуемого MTU=1280
- Были убраны поля IHL и Options. Механизм расширения заголовка возложен на поле Next Header
- Добавлено 20-битное необязательное поле Flow Label - Используя это поле, транзитным узлам передается информация о необходимости поддерживать один и тот же путь для потока пакетов, что поможет избежать их переупорядочивания
- Поле Total Length переименовано в Payload Length поскольку это поле не включает в себя длину самого заголовка IPv6