Теория 9 (4)

Формат сегментов TCP, SCTP, UPD

TCP

Бит	0 — 3	4 — 6	7 — 15	16 — 31
0	Порт источника, **Source Port**			Порт назначения, **Destination Port**
32	Порядковый номер, **Sequence Number (SN)**			
64	Номер подтверждения, **Acknowledgment Number (ACK SN)**			
96	Длина заголовка, (**Data offset**)	Зарезервировано	Флаги	Размер Окна, **Window size**
128	Контрольная сумма, **Checksum**			Указатель важности, **Urgent Point**
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

- Порт источника - идентифицирует приложение клиента, с которого отправлены пакеты.
- Порт назначения - идентифицирует порт, на который отправлен пакет.
- Порядковый номер - измеряется в байтах, и каждый переданный байт полезных данных увеличивает это значение на 1.
- Номер подтверждения - содержит порядковый номер байта, который отправитель данного сегмента желает получить. Это означает, что все предыдущие октеты были успешно получены.
  Каждая сторона подсчитывает свой 'Порядковый номер' для переданных данных и отдельно 'Номер подтверждения' для полученных данных. 'Порядковый номер' каждой из сторон соответствует 'Номер подтверждения' другой стороны.
- Длина заголовка - указывает значение длины заголовка, измеренное в 32-битовых словах. Минимальный размер составляет 20 байт (пять 32-битовых слов), а максимальный — 60 байт (пятнадцать 32-битовых слов).
- Зарезервировано - Зарезервировано (3 бита) для будущего использования и должно устанавливаться в ноль.
- Флаги - 9 битовых флагов
- Размер окна - Window Size самостоятельно определяет количество байт данных, после передачи которых отправитель ожидает подтверждения от получателя, что данные получены. Иначе говоря, получатель пакета располагает для приёма данных буфером длиной 'размер окна' байт.
- Контрольная сумма - Если сегмент, по которому вычисляется контрольная сумма, имеет длину не кратную 16-битам, то длина сегмента увеличивается до кратной 16-ти, за счёт дополнения к нему справа нулевых битов заполнения. Биты заполнения (0) не передаются в сообщении и служат только для расчёта контрольной суммы. При расчёте контрольной суммы значение самого поля контрольной суммы принимается равным 0.
- Указатель важности - Это поле указывает порядковый номер байта, которым заканчиваются важные данные.
- Опции - Могут применяться в некоторых случаях для расширения протокола.

- Данные - данные.

SCTP

Биты	Биты 0-7	8-15	16-23	24-31
+0	Порт источника		Порт назначения	
32	Тег проверки			
64	Контрольная сумма			
96	Тип 1 блока	Флаги 1 блока	Длина 1 блока	
128	Данные 1 блока			
...	...			
...	Тип N блока	Флаги N блока	Длина N блока	
...	Данные N блока			

SCTP пакеты имеют более простую структуру, чем пакеты TCP. Каждый пакет состоит из двух основных разделов:

1. Общий заголовок, который занимает первые 12 байт (выделены синим цветом)
2. Блоки данных, которые занимают оставшуюся часть пакета.

Первый блок отмечен зелёным цветом, и последний из блоков N (N блок) выделен красным.
Каждый блок имеет идентификатор типа, занимающий один байт. Таким образом, возможно определение не более 255 различных типов блоков. RFC 4960 определяет список типов блоков, всего на данный момент определено 15 типов. Остальная часть блока состоит из поля длины размером в 2 байта (максимальная длина, которая может содержаться в данном поле, равна 65535 байтам) и, собственно, данных. Если размер блока не кратен 4 байтам, то он заполняется нулями до размера, кратного 4 байтам.

UPD

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

Заголовок UPD состоит из четырёх полей, каждое по 2 байта (16 бит). Два из них необязательны к использованию в IPv4 (розовые ячейки), в то время как в IPv6 необязателен только порт отправителя.

- Порт отправителя - Предполагается, что это значение задаёт порт, на который при необходимости будет посылаться ответ. В противном же случае значение должно быть равным 0. Если хостом-источником является клиент, то номер порта будет, скорее всего, динамическим. Если источником является сервер, то его порт будет одним из «хорошо известных».
- Порт получателя - Аналогично порту отправителя, если хостом-получателем является клиент, то номер порта динамический, если получатель — сервер, то это будет «хорошо известный» порт.
- Поле, задающее длину всей датаграммы (заголовка и данных) в байтах. Минимальная длина равна длине заголовка — 8 байт. Теоретически, максимальный размер поля — 65535 байт для UDP-датаграммы (8 байт на заголовок и 65527 на данные). Фактический предел для длины данных при использовании IPv4 — 65507 (помимо 8 байт на UDP-заголовок требуется ещё 20 на IP-заголовок).
- Контрольная сумма - Поле контрольной суммы используется для проверки заголовка и данных на ошибки. Если сумма не сгенерирована передатчиком, то поле заполняется нулями. Поле не является обязательным для IPv4. Расчёт контрольной суммы описан в RFC 1071 - для IPv4 и в RFC 2460 - для IPv6

UDP

User Data Protocol - протокол пользовательских датаграмм.
Минимальный ориентированный на обработку сообщений протокол транспортного уровня, задокументированный в RFC 768.
UDP использует простую модель передачи, без явных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа, но гарантируется, что если они придут, то в целостном состоянии. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении
UDP обеспечивает многоканальную передачу и проверку целостности заголовка и существенных данных.

TCP алгоритм медленного пуска

Алгоритм медленного старта (Slow Start) — это часть механизма контроля перегрузки в протоколе TCP. Этот алгоритм используется для избежания отправки большего количества данных, чем сеть способна передать.

Вот как он работает:

1. Когда TCP-соединение устанавливается, размер окна устанавливается на небольшое значение (обычно равное двум размерам максимального сегмента, или MSS).

2. Затем, каждый раз, когда получено подтверждение, размер окна увеличивается на один MSS. Это продолжается до тех пор, пока размер окна не достигнет порога перегрузки.
3. После достижения этого порога, алгоритм медленного старта перестает работать, и начинает работать алгоритм предотвращения перегрузки (congestion avoidance).
4. Если во время передачи данных происходит потеря пакетов, TCP уменьшает порог перегрузки вдвое и возвращает размер окна к начальному значению, после чего процесс начинается сначала.

Этот алгоритм позволяет TCP эффективно использовать пропускную способность сети, избегая в то же время перегрузки сети. Он также помогает обеспечить справедливое распределение пропускной способности между различными TCP-соединениями.

ⓘ Небольшие сноски:

- MSS - Maximum Segmet Size. Максимальный размер сегмента. Параметр, определяющий размер полезного блока данных в TCP-пакете, без учёта длины заголовка TCP и IP
- TCP-окно - Размер буфера, содержащего копию всех отправленных устройством пакетов. Если какой-то пакет не был доставлен или поврежден, то он отправляется заново из копии в буфере.

TCP алгоритм тройного рукопожатия

Тройное рукопожатие (Three-way handshake) — это процесс, который используется в протоколе TCP для установления соединения между клиентом и сервером. Он состоит из трех шагов:

1. **SYN**: Клиент отправляет сегмент с установленным флагом SYN на сервер, указывая, что он хочет установить соединение. В этом сегменте клиент также указывает свой начальный номер последовательности (ISN).
2. **SYN-ACK**: Сервер получает сегмент SYN, увеличивает полученный номер последовательности на единицу и отправляет обратно сегмент с установленными флагами SYN и ACK. Сервер также указывает свой собственный ISN.
3. **ACK**: Клиент получает SYN-ACK, увеличивает оба номера последовательности на единицу и отправляет обратно сегмент с установленным флагом ACK. После этого устанавливается соединение, и начинается передача данных.

Номера последовательности - это порядковый номер и номер подтверждения из TCP сегмента.

TCP обеспечение надёжности

Надёжность TCP протокола в первую очередь обеспечивается нумерацией переданных и полученных байт данных, что позволяет отслеживать и исправлять потери.

TCP таймеры

Протокол TCP использует 4 таймера:

1. Таймер повторной передачи (Retransmission Timer): Этот таймер используется при ожидании подтверждения. Когда посылается сегмент, запускается таймер повторной передачи. Если подтверждение получения сегмента прибывает раньше, чем истекает период таймера, таймер останавливается.
2. Таймер задержки ответа (Delayed ACK Timer): Этот таймер используется для повышения эффективности передачи по сети. После получения данных от клиента сервер не сразу возвращает ACK клиенту, а ожидает период времени, который обычно составляет максимум 200 мс.
3. Таймер “оставайся в живых” (Keepalive Timer): Этот таймер используется для определения, что клиент вышел из строя. Большинство версий Telnet и Rlogin серверов посылают пакеты “оставайся в живых” каждые 2 часа.
4. Таймер установления соединения (Connection-establishment Timer): Этот таймер используется при установлении соединения. В процессе установления соединения при отправке «запроса на установление пакета SYN соединения» запускается таймер, который по умолчанию равен 3 секундам.

TCP SYN Flood атака

Атака TCP SYN Flood - это вид сетевой атаки типа отказ от обслуживания (DoS), которая заключается в отправке большого количества SYN-запросов (запросов на подключение по протоколу TCP) в достаточно короткий срок.

Принцип атаки заключается в том, что злоумышленник, посылая SYN-запросы, переполняет на сервере (цели атаки) очередь на подключения. При этом он игнорирует SYN+ACK пакеты цели, не высылая ответные пакеты, либо подделывает заголовок пакета таким образом, что ответный SYN+ACK отправляется на несуществующий адрес.

В результате в очереди подключений появляются так называемые полуоткрытые соединения (half-open connection), ожидающие подтверждения от клиента. По истечении определенного тайм-аута эти подключения отбрасываются. Задача злоумышленника заключается в том, чтобы поддерживать очередь заполненной таким образом, чтобы не допустить новых подключений1. Из-за этого клиенты, не являющиеся злоумышленниками, не могут установить связь, либо устанавливают её с существенными задержками.

Однако стоит отметить, что сетевой протокол транспортного уровня SCTP, который является более современным по сравнению с TCP, использует SYN cookie и не подвержен SYN-флуд-атакам.

SCTP

Протокол, аналогичный TCP, но SCTP предает последовательность сообщений (каждое из которых является группой байт), а не передает непрерывный поток байтов, как в TCP. Одним из нововведений SCTP является многопоточность, защита от DDoS-атак, синхронное соединение между двумя хостами по двум и более независимым физическим каналам (multi-homing)

Многопоточность в SCTP позволяет передавать несколько потоков в рамках одной ассоциации. Ассоциацией в SCTP называется соединение между двумя хостами. Это отличается от протокола TCP, где данные и служебная информация передаются по одному соединению.
Это означает, что SCTP может обрабатывать несколько независимых потоков данных в рамках одного соединения, что улучшает производительность и эффективность передачи данных, особенно в сетях с высокой задержкой. Каждый поток обрабатывается отдельно, так что если один поток блокируется или теряет пакеты, это не влияет на другие потоки.

Мультихоуминг в SCTP - это функция, которая позволяет одной ассоциации SCTP использовать несколько IP-адресов на каждом конце. Это означает, что если у вас есть несколько сетевых интерфейсов (например, несколько сетевых карт или несколько IP-адресов на одной сетевой карте), SCTP может использовать все эти интерфейсы в рамках одной ассоциации.
Это приводит к увеличению отказоустойчивости, поскольку если один из интерфейсов или пути становится недоступным, SCTP автоматически переключится на другой доступный интерфейс или путь. Это также может улучшить производительность, поскольку SCTP может распределять нагрузку между несколькими интерфейсами.

SCTP алгоритм четверного рукопожатия

Алгоритм четверного рукопожатия SCTP состоит из следующих шагов:

1. Клиент отправляет сигнал INIT на сервер, чтобы инициировать ассоциацию.
2. При получении сигнала INIT сервер отправляет ответ INIT-ACK клиенту. Этот сигнал INIT-ACK содержит состояние cookie. Это состояние cookie должно содержать код аутентификации сообщений (MAC), а также временную метку, соответствующую созданию cookie, срок жизни состояния cookie и информацию, необходимую для установления ассоциации. MAC вычисляется сервером на основе секретного ключа, известного только ему.
3. При получении этого сигнала INIT-ACK клиент отправляет ответ COOKIE-ECHO, который просто повторяет состояние cookie.
4. После проверки подлинности состояния cookie с использованием секретного ключа сервер затем выделяет ресурсы для ассоциации, отправляет ответ COOKIE-ACK, подтверждающий сигнал COOKIE-ECHO, и переводит ассоциацию в состояние ESTABLISHED.

Этот алгоритм четверного рукопожатия разработан таким образом, чтобы клиент (или инициатор) и сервер (или респондент) могли независимо доказать друг другу, что они знают секретный ключ, не раскрывая его.

IPv4 и IPv6

IPv4

Формат пакета:

Отступ	Октет	0								1								2								3							
Октет	Бит	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Версия				Размер заголовка				Differentiated Services Code Point						Explicit Congestion Notification		Размер пакета (полный)															
4	32	Идентификатор																Флаги			Смещение фрагмента											
8	64	Время жизни								Протокол								Контрольная сумма заголовка															
12	96	IP-адрес источника																															
16	128	IP-адрес назначения																															
20	160	Опции (если размер заголовка > 5)																															
20 или 24+	160 или 192+	Данные																															

- Версия

  > Первым полем заголовка пакета является версия протокола размером в четыре бита. Для IPv4 это 4.

- Размер заголовка (Internet Header Length)

  > Следующие четыре бита содержат размер заголовка пакета в 32-битных словах. Поскольку число опций непостоянно, указание размера важно для отделения заголовка от данных. Минимальное значение равно 5 (5×32=160 бит, 20 байт), максимальное — 15 (60 байт).

- Differentiated Services Code Point (DSCP)

  > Изначально называлось «тип обслуживания» (Type of Service, ToS), в настоящее время определяется RFC 2474 как «Differentiated Services». Используется для разделения трафика на классы обслуживания, например, для установки чувствительному к задержкам трафику, такому как VoIP, большего приоритета.

- Указатель перегрузки (Explicit Congestion Notification, ECN)

  > Предупреждение о перегрузке сети без потери пакетов. Является необязательной функцией и используется, только если оба хоста её поддерживают.

- Размер пакета

  > 16-битный полный размер пакета в байтах, включая заголовок и данные. Минимальный размер равен 20 байтам (заголовок без данных), максимальный — 65535 байт. Хосты должны поддерживать передачу пакетов размером до 576 байт, но современные реализации обычно поддерживают гораздо больший размер. Пакеты большего размера, чем поддерживает канал связи, фрагментируются.

- Идентификатор

  > Преимущественно используется для идентификации фрагментов пакета, если он был фрагментирован. Существуют эксперименты по его использованию для других целей, таких как добавление

> информации о трассировке пакета для упрощения отслеживания пути пакета с подделанным адресом источника.

- Флаги

> Поле размером три бита, содержащее флаги контроля над фрагментацией. Биты, от старшего к младшему, означают:
> 0: Зарезервирован, должен быть равен 0.
> 1: Не фрагментировать
> 2: У пакета ещё есть фрагменты
> Если установлен флаг «не фрагментировать», то в случае необходимости фрагментации такой пакет будет уничтожен. Может использоваться для передачи данных хостам, не имеющим достаточных ресурсов для обработки фрагментированных пакетов. Флаг «есть фрагменты» должен быть установлен в 1 у всех фрагментов пакета, кроме последнего. У нефрагментированных устанавливается в 0 — такой пакет считается собственным последним фрагментом.

- Смещение фрагмента

> Поле размером в 13 бит, указывает смещение поля данных текущего фрагмента относительно начала поля данных первого фрагментированного пакета в блоках по 8 байт. Позволяет задать до $(213-1)\times8=65528$ байт смещения. При учёте размера заголовка итоговое смещение может превысить максимальный размер пакета (65528 + 20 = 65548 байт). Первый фрагмент в последовательности имеет нулевое смещение.

- «Время жизни» (Time to Live, TTL) пакета

> Определяет максимальное количество маршрутизаторов на пути следования пакета. Наличие этого параметра не позволяет пакету бесконечно ходить по сети. Каждый маршрутизатор при обработке пакета должен уменьшить значение TTL на единицу. Пакеты, время жизни которых стало равно нулю, уничтожаются, а отправителю посылается сообщение ICMP Time Exceeded. На отправке пакетов с разным временем жизни основана трассировка их пути прохождения (traceroute). Максимальное значение TTL=255. Обычное начальное значение TTL=64 (зависит от ОС).

- Протокол

> Указывает, данные какого протокола IP содержит пакет (например, TCP или ICMP). Присвоенные номера протоколов можно найти на сайте IANA.

- Контрольная сумма заголовка

> 16-битная контрольная сумма, используемая для проверки целостности заголовка. Каждый хост или маршрутизатор сравнивает контрольную сумму заголовка со значением этого поля и отбрасывает пакет, если они не совпадают. Целостность данных IP не проверяет — она проверяется протоколами более высоких уровней (такими, как TCP или UDP), которые тоже используют контрольные суммы.
> Поскольку TTL уменьшается на каждом шаге прохождения пакета, сумма тоже должна вычисляться на каждом шаге. Метод пересчёта контрольной суммы определён в RFC 1071.

- Адрес источника

  > 32-битный адрес отправителя пакета. Может не совпадать с настоящим адресом отправителя из-за трансляции адресов.

- Адрес назначения

  > 32-битный адрес получателя пакета. Также может меняться при трансляции адресов.

- Опции

  > За адресом назначения может следовать поле дополнительных опций, но оно используется редко. Размер заголовка в этом случае должен быть достаточным, чтобы вместить все опции (с учётом дополнения до целого числа 32-битных слов). Присвоенные номера опций размещаются на сайте IANA.

Фрагментация в IPv4 - это процесс, при котором Интернет-протокол (IP) разбивает пакеты на более мелкие части (фрагменты), чтобы полученные части могли проходить через ссылку с меньшим максимальной единицей передачи (MTU), чем исходный размер пакета1. Фрагменты затем повторно собираются принимающим хостом1.
Чтобы хост-получатель мог осуществлять повторную сборку дейтаграмм, разработчики IPv4 поместили в дейтаграмму поля идентификации, флага и фрагментации2. Когда дейтаграмма создается, хост-отправитель маркирует ее номером-идентификатором, а также помещает в нее адреса отправителя и получателя2.
Возможно возникновение ситуации, когда размер пакета превысит возможности узла системы связи. В этом случае протокол предусматривает возможность дробления пакета на уровне IP в процессе доставки

✓ **Адреса**

В IPv4 адреса разделены по классам:

- Класс A

  > 0.XXX.XXX.XXX — 127.XXX.XXX.XXX
  > Первый бит адреса равен нулю, таким образом, класс A занимает половину всего адресного пространства. Адрес сети занимает 7 бит, адрес узла — 24 бита, следовательно класс A содержит 128 подсетей по 16 777 216 адресов в каждой.
  > По умолчанию зарезервирована, не маршрутизируется в интернете и используется для построения локальных и корпоративных сетей.

- Класс B

  > 128.0.XXX.XXX — 191.255.XXX.XXX
  > Адрес начинается с битов 1,0, таким образом, класс B занимает четверть всего адресного пространства. Адрес сети занимает 14 бит, адрес узла — 16, следовательно класс B содержит 16 384 подсетей по 65 536 адресов в каждой.
  > Зарезервирована для «канальных» адресов.

- Класс C

  > 192.0.0.XXX — 223.255.255.XXX
  > Адрес начинается с битов 1,1,0, таким образом, класс C занимает 1/8 адресного пространства. Адрес сети занимает 21 бит, адрес узла — 8 бит, следовательно класс C содержит 2 097 152 сетей по 256 адресов в каждой.
  > Зарезервирована для примеров в документации.

- Класс D

  > 224.XXX.XXX.XXX — 239.XXX.XXX.XXX
  > Адрес начинается с битов 1,1,1,0. Класс D занимает 1/16 адресного пространства. Используется для многоадресной рассылки.

- Класс E

  > 240.XXX.XXX.XXX — 255.XXX.XXX.XXX
  > Адрес начинается с битов 1,1,1,1. Такие адреса запрещены.
  > Зарезервировано для использования в будущем.

Со временем классовая система оказалось неэффективной и дополнилась бесклассовой адресацией. Была введена дополнительная метрика - маска подсети, определяющая, соколько бит адреса отводится на адрес сети, а сколько на адрес узла.

IPv6

✓ Фомат пакета:

Отступ в октетах		0								1								2								3							
	Отступ в битах	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
C	96																																
10	128																																
14	160																																
18	192	Destination Address																															
1C	224																																
20	256																																
24	288																																

- Version: версия протокола; для IPv6 это значение равно 6 (значение в битах — 0110).
- Traffic Class: приоритет пакета (8 бит). Это поле состоит из двух значений. Старшие 6 бит используются DSCP для классификации пакетов. Оставшиеся два бита используются ECN для контроля перегрузки.
  Flow Label: метка потока.
- Payload Length: (16 бит) размер данных в октетах, не включая данный заголовок, но включая все расширенные заголовки.
- Next Header: задаёт тип расширенного заголовка (англ. IPv6 extension), который идёт следующим. В последнем расширенном заголовке поле Next Header задаёт тип транспортного протокола (TCP, UDP и т. д.)
- Hop Limit: аналог поля time to live в IPv4 (8 бит).
- Source Address и Destination Address: адрес отправителя и получателя соответственно; по 128 бит.

С надеждой, что современные технологии канального и транспортного уровней обеспечивают достаточный уровень обнаружения ошибок, заголовок не содержит контрольной суммы.

IPv6-пакеты никогда не фрагментируются маршрутизаторами. Пакеты, чей размер превышает MTU сетевого подключения уничтожаются и отправителю посылается сообщение Packet too Big (ICMPv6 тип 2). Подобное поведение в IPv4 происходит, если установлен бит Don’t Fragment.

✓ Адреса

Нет обратной совместимости с IPv4
Существуют различные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast).
Адреса типа Unicast хорошо всем известны. Пакет, посланный на такой адрес, достигает в точности интерфейса, который этому адресу соответствует.
Адреса типа Anycast синтаксически неотличимы от адресов Unicast, но они адресуют группу интерфейсов. Пакет, направленный такому адресу, попадёт в ближайший (согласно метрике маршрутизатора) интерфейс. Адреса

Anycast могут использоваться только маршрутизаторами.
Адреса типа Multicast идентифицируют группу интерфейсов. Пакет, посланный на такой адрес, достигнет всех интерфейсов, привязанных к группе многоадресного вещания.
Широковещательные адреса IPv4 (обычно xxx.xxx.xxx.255) выражаются адресами многоадресного вещания IPv6. Крайние адреса подсети IPv6 (например, xxxx: xxxx: xxxx: xxxx:0:0:0:0 и xxxx: xxxx: xxxx: xxxx: ffff: ffff: ffff: ffff для подсети /64) являются полноправными адресами и могут использоваться наравне с остальными.
Группы цифр в адресе разделяются двоеточиями (например, fe80:0:0:0:200:f8ff: fe21:67cf). Незначащие старшие нули в группах могут быть опущены. Большое количество нулевых групп может быть пропущено с помощью двойного двоеточия (fe80::200:f8ff: fe21:67cf). Такой пропуск должен быть единственным в адресе.

Типы Unicast-адресов

- Глоабльные

  > Соответствуют публичным IPv4-адресам. Могут находиться в любом не занятом диапазоне. В настоящее время региональные интернет-регистраторы распределяют блок адресов 2000::/3 (с 2000:: по 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)

- Link-Local

  > Соответствуют автосконфигурированным с помощью протокола APIPA IPv4 адресам. Начинаются с FE80:.
  > Используется:
  > 1. В качестве исходного адреса для Router Solicitation(RS) и Router Advertisement(RA) сообщений, для обнаружения маршрутизаторов.
  > 2. Для обнаружения соседей (эквивалент ARP для IPv4).
  > 3. Как next-hop-адрес для маршрутов.

- Unique-Local

  > RFC 4193, соответствуют внутренним IP-адресам, которыми в версии IPv4 являлись 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16. Начинаются с цифр FCxx: и FDxx:.

Нотация:

> Адреса IPv6 отображаются как восемь четырёхзначных шестнадцатеричных чисел (то есть групп по четыре символа), разделённых двоеточием. Пример адреса:

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

> 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. Например, 2001:0db8:0000:0000:0000:0000:ae21:ad12 может быть сокращён до 2001:db8::ae21:ad12, или 0000:0000:0000:0000:0000:0000:ae21:ad12 может быть сокращён до ::ae21:ad12.
> Сокращению не могут быть подвергнуты 2 разделённые нулевые группы из-за возникновения неоднозначности.
> Также есть специальная нотация для записи встроенного и отображённого IPv4 на IPv6. В ней последние 2 группы знаков заменены на IPv4-адрес в его формате. Пример:
>
> ```
> ::ffff:192.0.2.1
> ```

В некоторых случаях, IPv6-адрес может быть сформирован на основе MAC-адреса устройства, но этот процесс стал менее распространенным из-за проблем с приватностью. Вместо этого, многие устройства теперь используют случайные или псевдослучайные значения для последней части своего IPv6-адреса

Протокол ARP

2 уровень модели OSI
Протокол ARP (Address Resolution Protocol) - это протокол, используемый для определения сетевого адреса уровня интерфейса (обычно MAC-адреса) по известному IP-адресу. Это необходимо для того, чтобы устройства в сети могли отправлять пакеты друг другу.

Вот как это работает:

1. Когда устройство хочет отправить пакет другому устройству в сети, оно сначала проверяет свою ARP-таблицу, чтобы увидеть, есть ли у него уже MAC-адрес целевого устройства.
2. Если MAC-адрес не известен, устройство отправляет широковещательный ARP-запрос в сеть, спрашивая “Кто имеет этот IP-адрес?”
3. Устройство с запрошенным IP-адресом отвечает на ARP-запрос, отправляя свой MAC-адрес обратно.
4. Первоначальное устройство затем обновляет свою ARP-таблицу с новым MAC-адресом и может продолжить отправку пакетов.

Протокол RARP

2 уровень модели OSI
Протокол RARP (Reverse Address Resolution Protocol) используется для определения IP-адреса устройства по его физическому (MAC) адресу. Это может быть полезно для устройств, которые не имеют возможности хранить свой IP-адрес, например, для дисковых станций или систем, загружающихся по сети.

Вот как это работает:

1. Клиентская машина отправляет широковещательный запрос RARP в сеть, указывая свой MAC-адрес и запрашивая соответствующий ему IP-адрес.
2. Сервер RARP, который слушает такие запросы в сети, отвечает, отправляя запрошенный IP-адрес.
3. Клиентская машина затем использует этот IP-адрес для своей сетевой конфигурации.

Служба каталогов NDS

Служба каталогов NDS (NetWare Directory Services) - это глобальная справочная служба, опирающаяся на распределенную объектно-ориентированную базу данных сетевых ресурсов. Многоуровневая база данных содержит информацию обо всех пользователях, группах пользователей, принтерах, томах и компьютерах.

С точки зрения службы каталогов NetWare (NDS), сеть - это не совокупность ЭВМ, а интегрированная информационная среда. При регистрации в сети клиент получает доступ ко всем ее ресурсам. Для повышения надежности и сокращения времени доступа к сети база данных службы каталогов NetWare распределена по сети. Отдельные фрагменты базы данных могут располагаться на разных серверах, и могут быть задублированы на нескольких серверах, имеются специальные сетевые средства для синхронизации изменений в этих секциях

ICMP

ICMP (Internet Control Message Protocol) используется для отправки сообщений об ошибках и операционной информации, таких как эхо-запрос (ping) и эхо-ответ (pong).

ICMP Flood Attack - это вид DoS-атаки (Denial of Service), при которой злоумышленник отправляет большое количество ICMP-пакетов (например, эхо-запросов) на целевую систему с целью перегрузить ее и сделать недоступной для других пользователей.

Перенаправление шлюза

Перенаправление шлюза - это процесс, при котором пакеты, предназначенные для одного шлюза, перенаправляются на другой шлюз.

Эхо запрос - Эхо ответ

Эхо-запрос и эхо-ответ - это два типа ICMP-сообщений. Эхо-запрос (обычно известный как “ping”) отправляется от одного компьютера к другому с целью проверить доступность целевого компьютера. Эхо-ответ (или “pong”) - это ответ целевого компьютера на эхо-запрос.

Подавление трафика

Подавление трафика - это процесс уменьшения или ограничения объема сетевого трафика, обычно с целью предотвращения перегрузки сети или серверов. Это может быть осуществлено с помощью различных методов, включая ограничение скорости, блокировку определенных типов трафика или приоритизацию определенных типов трафика.