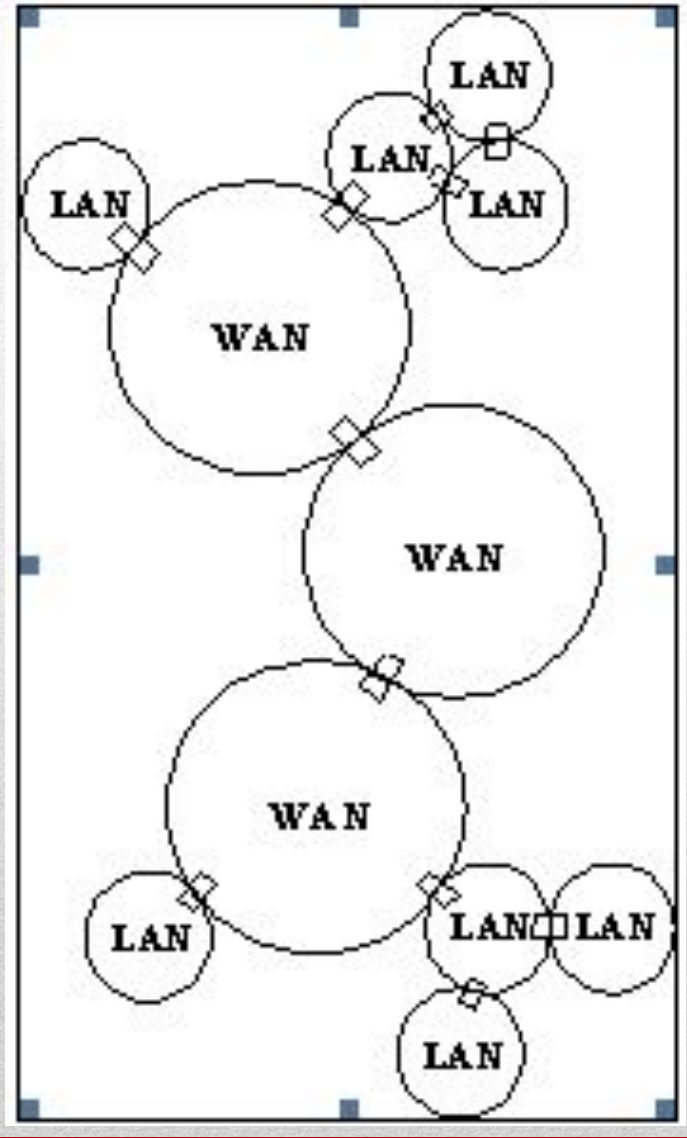
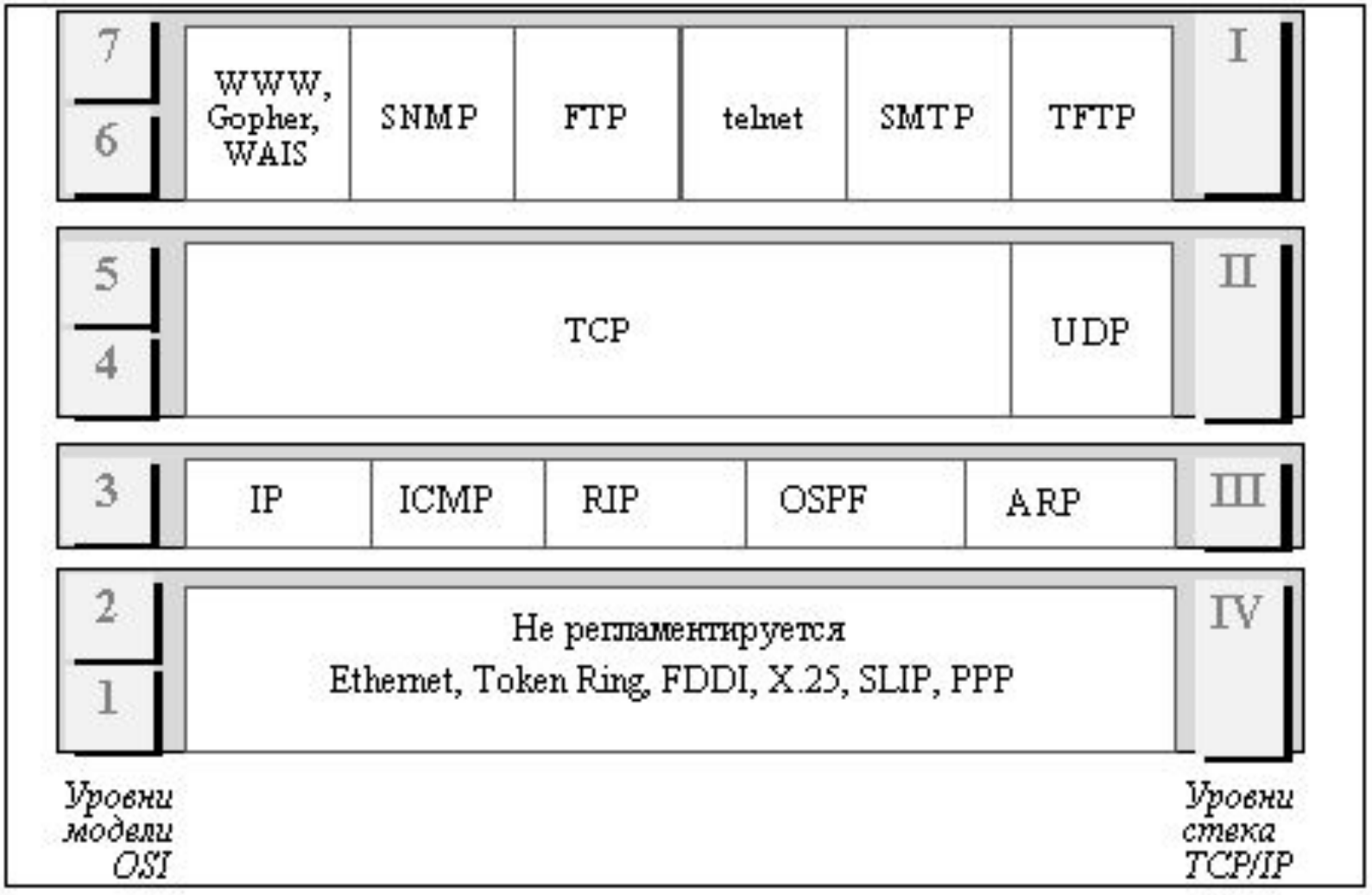
ВВЕДЕНИЕ В СТЕК TCP/IP

А.Н. Мартынюк
Кафедра КИСС ОНПУ 2014

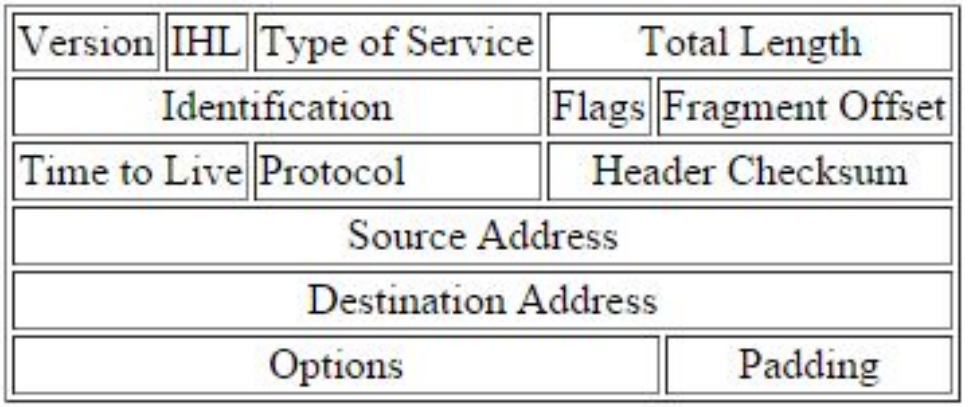
Построение составных сетей



Стек протоколов TCP/IP



Формат IP-заголовка



- Version (версия) 4 бита
- IHL (длина Internet заголовка) 4 бита
- Total Length (общая длина) 16 бит
- Identification (идентификатор) 16 бит
- Flags (различные управляющие флаги) 16 бит
- Fragment Offset (смещение фрагмента) 13 бит
- Time to Live (Время жизни) 8 бит
- Protocol (Протокол) 8 бит
- Header Checksum (Контрольная сумма заголовка) 16 бит
- Source Address (адрес отправителя) 32 бита
- Destination Address (адрес получателя) 32 бита
- Options (опции) поле переменной длины
- Padding (Выравнивание)

Структура IP-адреса

Класс A

0	N сети	N узла

Класс B

1	0	N сети	N узла

Класс C

1	1	0	N сети	N узла

Класс D

1	1	1	0	адрес группы multicast

Класс E

1	1	1	1	0	зарезервирован

Структура IP-адреса, диапазоны номеров сетей

Класс A

0	N сети	N узла

Класс B

1	0	N сети	N узла

Класс C

1	1	0	N сети	N узла

Класс D

1	1	1	0	адрес группы multicast

Класс E

1	1	1	1	0	зарезервирован

Класс	Наименьший адрес	Наибольший адрес
A	01.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0.	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Особая интерпретация IP-адресов

– если IP-адрес состоит только из двоичных нулей,

0 0 0 0 ..... 0 0 0 0

то он обозначает адрес того узла, который сгенерировал этот пакет;

– если в поле номера сети стоят 0,

0 0 0 0 0	Номер узла

то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет;

– если все двоичные разряды IP-адреса равны 1,

1 1 1 1 .....1 1

то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast);

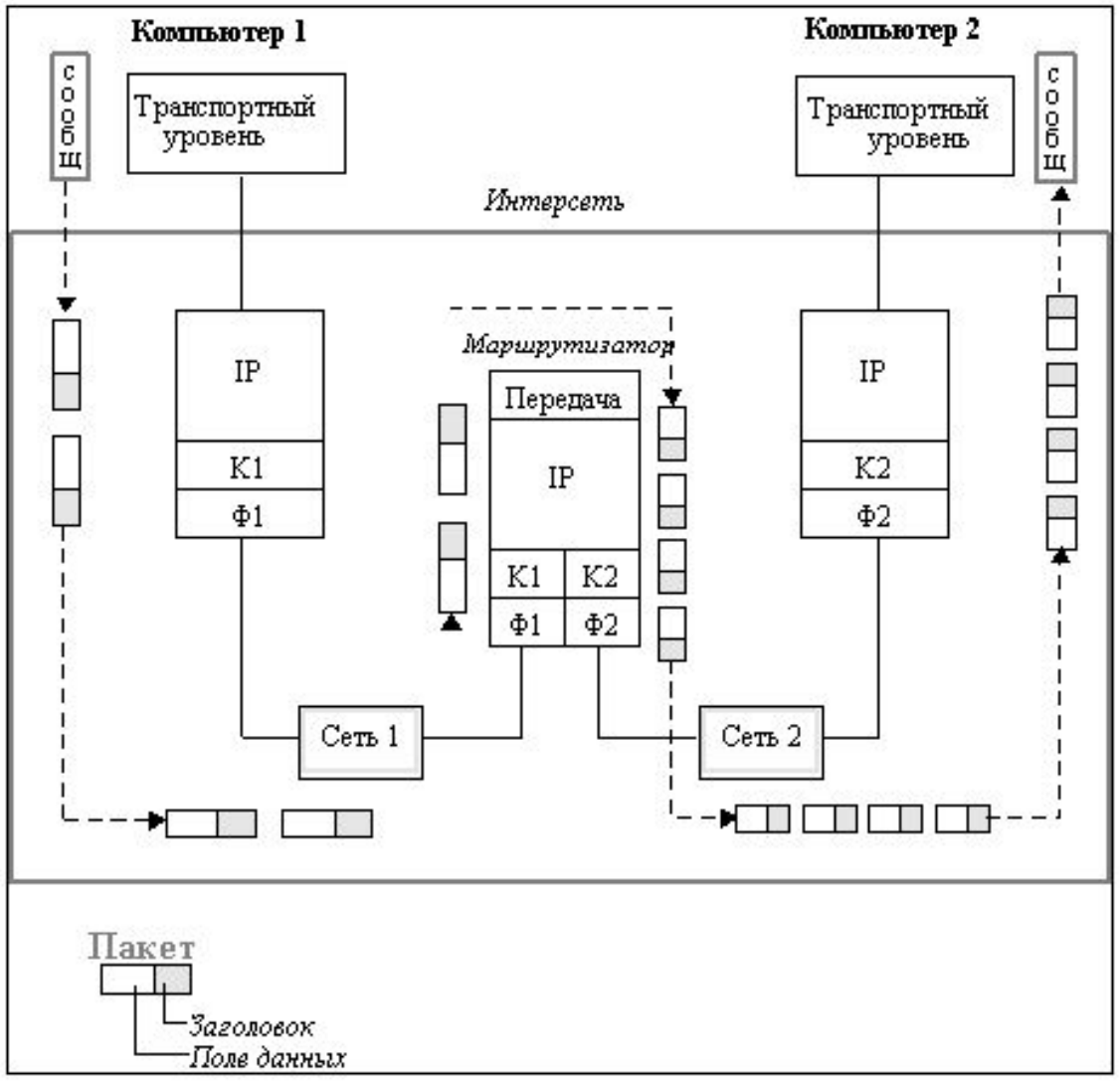
– если в поле адреса назначения стоят сплошные 1,

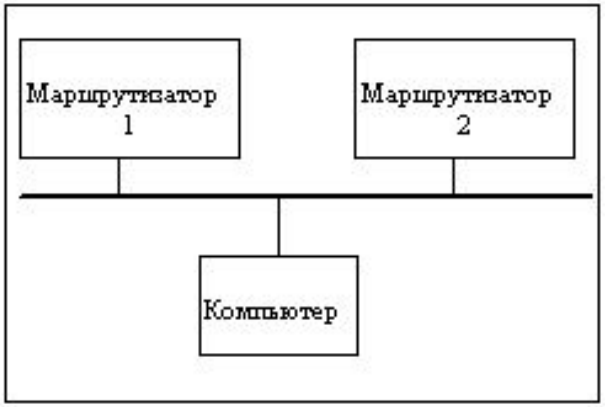
Номер сети	1111.....11

то пакет, имеющий такой адрес рассылается всем узлам сети с заданным номером. Такая рассылка называется широковещательным сообщением (broadcast);

– адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании работы программного обеспечения узла без реальной отправки пакета по сети. Этот адрес имеет название loopback.

Фрагментация IP-пакетов, выбор маршрутизатора





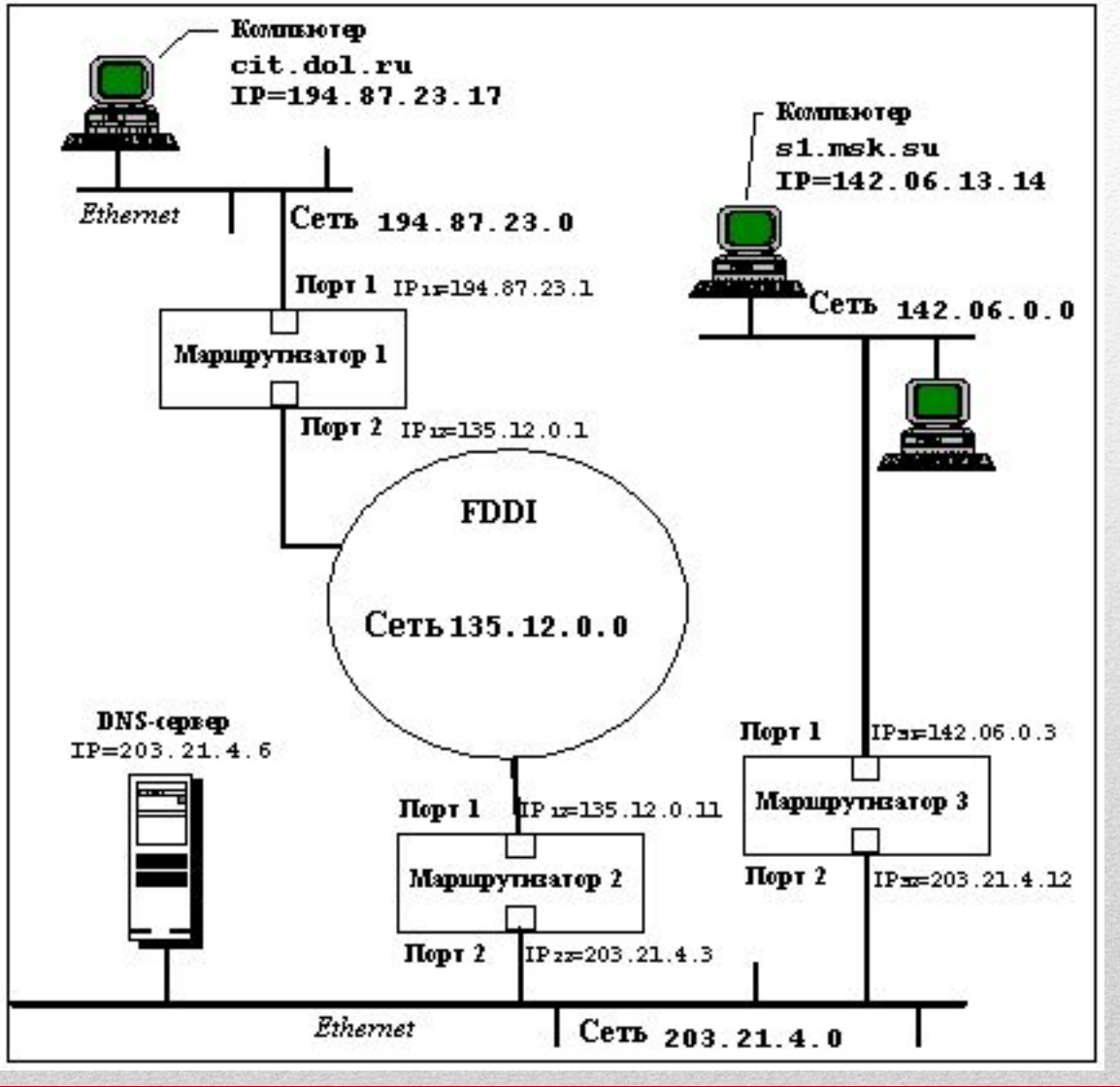
Условия фрагментации IP-пакетов

- Поле Internet идентификации (ID) используется вместе с адресами отправителя и получателя, полями протокола для идентификации фрагментов датаграммы при сборке.
- Бит флага More Fragments (MF) устанавливается, если датаграмма не является последним фрагментом.
- Поле Fragment Offset идентифицирует расположение фрагмента относительно начала в первоначальной не фрагментированной датаграмме. Единица измерения - 8 октетов.
- Стратегия фрагментации разработана так, чтобы не фрагментированная датаграмма имела нули во всех полях с информацией о фрагментации (MF=0, Fragment Offset=0). Если датаграмма фрагментируется, то выделение информации производится кусками и по границе 8 октет.
- Формат позволяет использовать 2**32=8192 фрагментов по 8 октетов каждый, а в целом 65536 октетов. Это совпадает со значением поля общей длины для датаграммы (заголовок учитывается в общей длине датаграммы, но не фрагментов).
- Когда происходит фрагментация, некоторые опции копируются, а другие остаются лишь в первом фрагменте.
- Каждый Internet модуль должен быть способен передать датаграмму из 68 октетов без дальнейшей фрагментации – Internet заголовок может включать до 60 октетов, минимальный фрагмент - 8 октетов.
- Каждый Internet - получатель должен быть в состоянии принять датаграмму из 576 октетов в качестве единого куска, либо в виде фрагментов, подлежащих сборке.

Типичная таблица маршрутов

Адрес сети назначения	Адрес следующего маршрутизатора	Номер выходного порта	Расстояние до сети назначения
56.0.0.0	198.21.17.7	1	20
56.0.0.0	213.34.12.4.	2	130
116.0.0.0	213.34.12.4	2	1450
129.13.0.0	198.21.17.6	1	50
198.21.17.0	-	2	0
213. 34.12.0	-	1	0
default	198.21.17.7	1	-

Взаимодействие компьютеров через интерсеть



Взаимодействие компьютеров через интерсеть 2

Компьютер cit.dol.ru отправляет по локальной сети кадр Ethernet, имеющий следующие поля:

DA (Ethernet)	...	DESTINATION IP	...	...
MAC_{11}		142.06.13.14		

Маршрутизатор 1 формирует кадр формата FDDI, в котором указывает MAC-адрес порта маршрутизатора 2, который он находит в своей кэш-таблице протокола ARP:

DA (FDDI)	...	DESTINATION IP	...	...
MAC_{21}		142.06.13.14		

Аналогично действует маршрутизатор 2, формируя кадр Ethernet для передачи пакета маршрутизатору 3 по сети Ethernet с IP-адресом 203.21.4.0:

DA (Ethernet)	...	DESTINATION IP	...	...
MAC_{32}		142.06.13.14		

Маршрутизатор посылает ARP-запрос по сети Ethernet с IP-адресом компьютера s1.msk.su (пусть этой информации в его кэше нет), получает ответ, содержащий адрес MAC_{s1}, и формирует кадр Ethernet, доставляющий IP-пакет по локальной сети адресату.

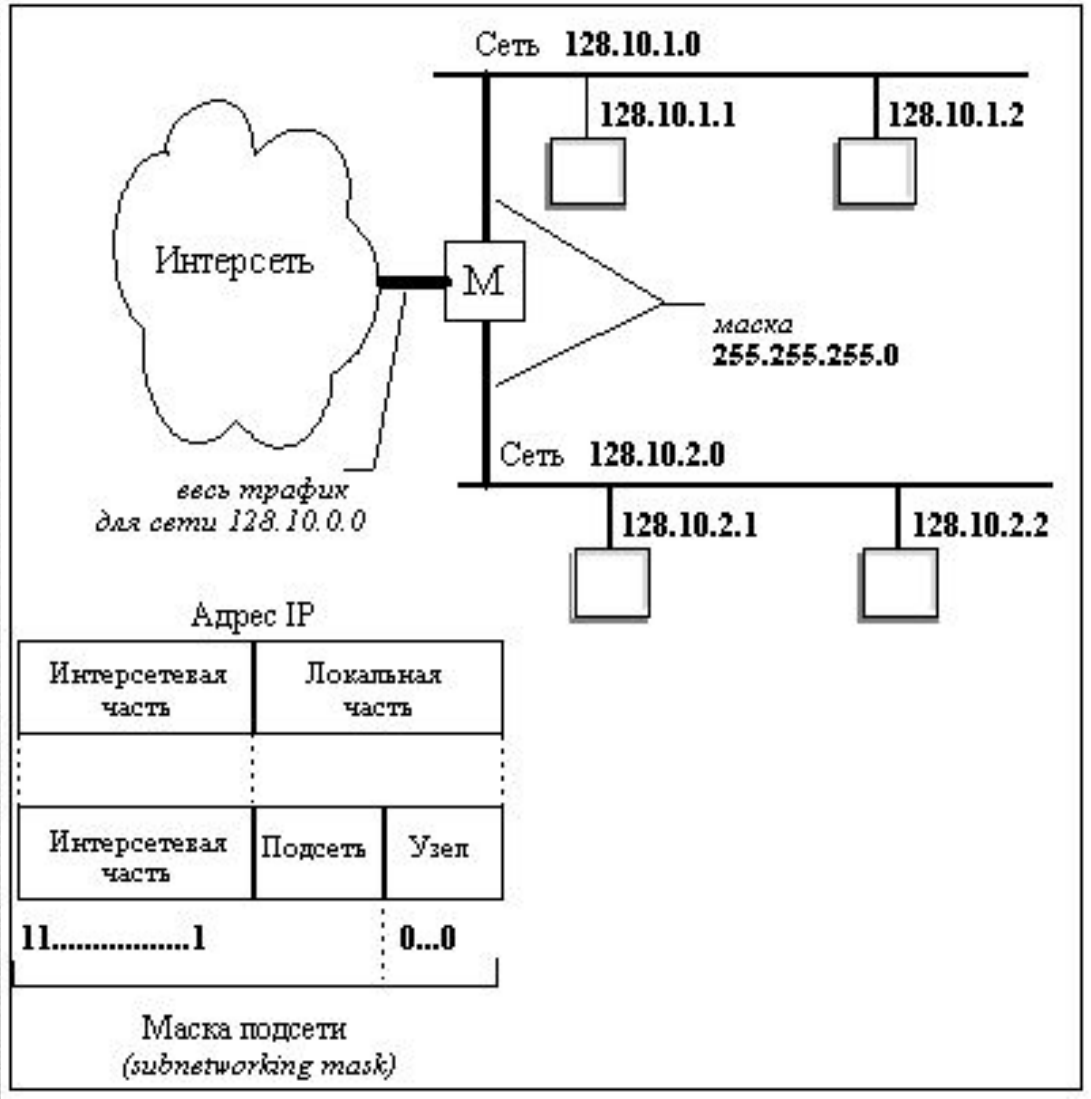
DA (Ethernet)	...	DESTINATION IP	...	...
MAC_{s1}		142.06.13.14		

Администратор получил возможность использовать вместо одного, централизованно заданного ему номера сети, четыре:

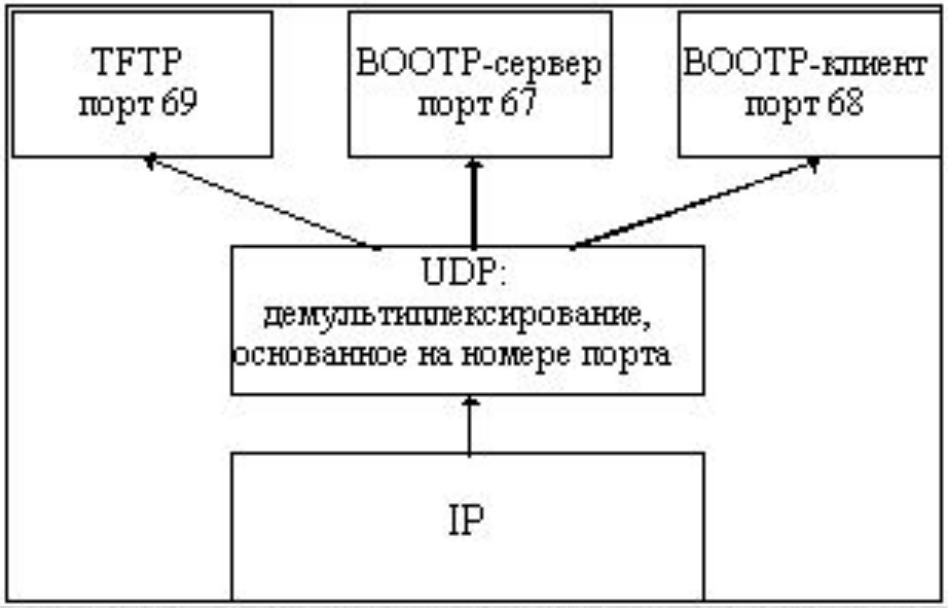
129.44.0.0 (10000001 00101100 00000000 00000000) 129.44.64.0 (10000001 00101100 01000000 00000000)

129.44.128.0 (10000001 00101100 10000000 00000000) 129.44.192.0 (10000001 00101100 11000000 00000000)

Использования масок для структурирования сети



Де мультиплексирование UDP

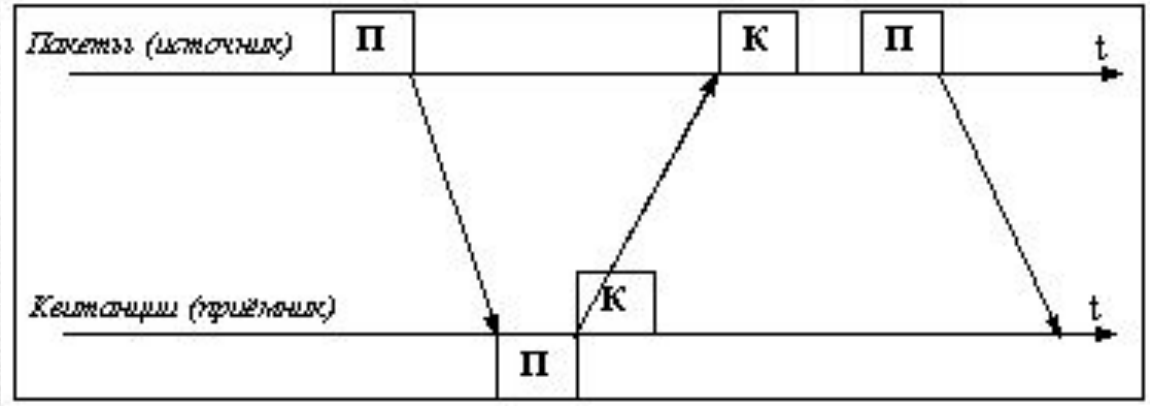


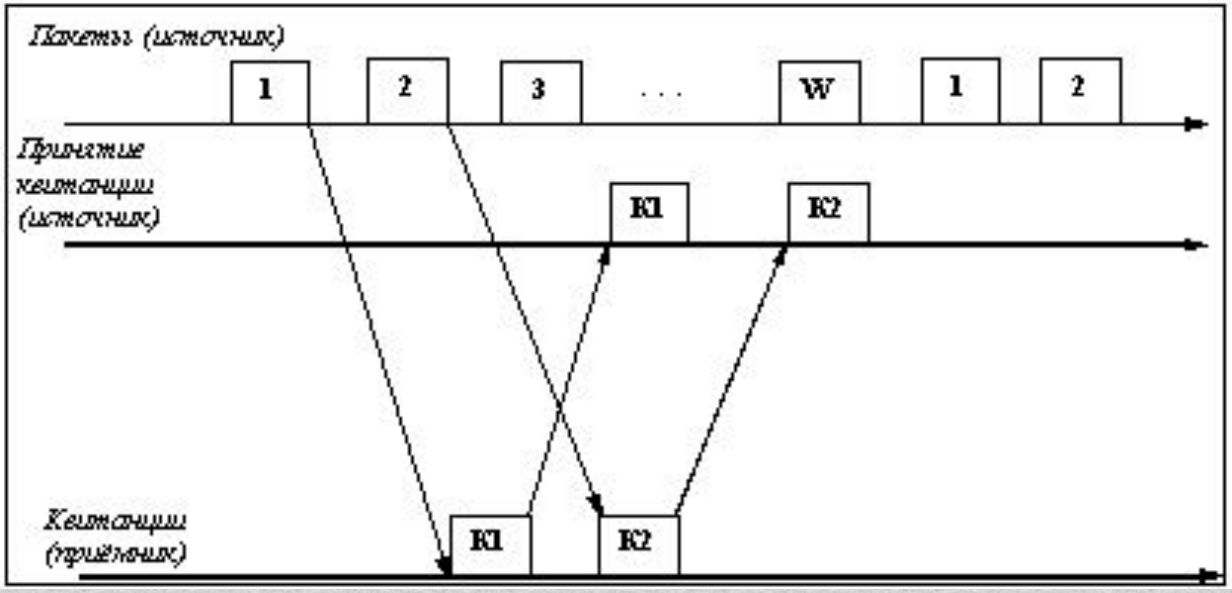
Заголовок UDP, псевдозаголовок для IPv4

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)	

Биты	0 — 7	8 — 15	16 — 23	24 — 31
0	Адрес источника			
32	Адрес получателя			
64	Нули	Протокол	Длина UDP	
96	Порт источника		Порт получателя	
128	Длина		Контрольная сумма	
160+	Данные			

Подтверждение корректности и окно TCP





Заголовок пакета TCP

Бит	0 — 3	4 — 9	10 — 15	16 — 31
0	Порт источника			Порт назначения
32	Номер последовательности			
64	Номер подтверждения			
96	Длина заголовка	Зарезервировано	Флаги	Размер Окна
128	Контрольная сумма			Указатель важности
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

- **Порт источника** идентифицирует приложение клиента
- **Порт назначения** идентифицирует порт, на который отправлен пакет
- **Номер последовательности** выполняет две задачи: Если установлен флаг SYN, то это начальное значение номера последовательности — ISN (Initial Sequence Number), и первый байт данных, которые будут переданы в следующем пакете, будет иметь номер последовательности, равный ISN + 1, иначе, если SYN не установлен, первый байт данных, передаваемый в данном пакете, имеет этот номер последовательности.
- **Номер подтверждения -** если установлен флаг ACK, то поле содержит номер последовательности, ожидаемый получателем в следующий раз.
- **Длина заголовка (смещение данных)** определяет размер заголовка пакета TCP в 4-байтных (4-октетных) словах. Минимальный размер составляет 5 слов, а максимальный — 15, что составляет 20 и 60 байт соответственно. Смещение считается от начала заголовка TCP.
- **Зарезервировано** (6 бит) для будущего использования, должно устанавливаться в ноль. Из них два (5-й и 6-й) определены:
  CWR (Congestion Window Reduced) — «Окно перегрузки уменьшено» — флаг установлен отправителем, чтобы указать, что получен пакет с установленным флагом ECE. **ECE** (ECN-Echo) — «Эхо ECN» — указывает, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю о перегрузках в сети.
- **Флаги (управляющие биты)** содержит 6 битовых флагов:
  - **URG** — Поле *«Указатель важности»* задействовано (*Urgent pointer field is significant*)
  - **ACK** — Поле *«Номер подтверждения»* задействовано (*Acknowledgement field is significant*)
  - **PSH** — (*Push function*) инструктирует получателя протолкнуть данные, накопившиеся в приемном буфере, в приложение пользователя
  - **RST** — Оборвать соединения, сбросить буфер (очистка буфера) (*Reset the connection*)
  - **SYN** — Синхронизация номеров последовательности (*Synchronize sequence numbers*)
  - **FIN** (*final*, бит) — флаг, будучи установлен, указывает на завершение соединения (*FIN bit used for connection*
  - *termination*).
- **Размер окна** содержит число, определяющее в байтах размер данных, которые отправитель готов принять.
- **Контрольная сумма** — это 16-битное дополнение к сумме всех 16-битных слов заголовка(включая псевдозаголовок) и данных.
- **Указатель важности –** 16-битовое значение положительного смещения от порядкового номера в данном сегменте, указывает порядковый номер октета, которым заканчиваются важные (urgent) данные. Поле принимается во внимание только для пакетов с установленным флагом URG.
- **Опции – м**огут применяться в некоторых случаях для расширения протокола. Иногда используются для тестирования.

Состояния сеанса TCP, псевдозаголовок

Состояния сеанса TCP	
CLOSED	Начальное состояние узла. Фактически фиктивное
LISTEN	Сервер ожидает запросов установления соединения от клиента
SYN-SENT	Клиент отправил запрос серверу на установление соединения и ожидает ответа
SYN-RECEIVED	Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения
ESTABLISHED	Соединение установлено, идёт передача данных
FIN-WAIT-1	Одна из сторон (назовём её узел-1) завершает соединение, отправив сегмент с флагом FIN
CLOSE-WAIT	Другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь сегмент ACK и продолжает одностороннюю передачу
FIN-WAIT-2	Узел-1 получает ACK, продолжает чтение и ждёт получения сегмента с флагом FIN
LAST-ACK	Узел-2 заканчивает передачу и отправляет сегмент с флагом FIN
TIME-WAIT	Узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт 2*MSL секунд, перед окончательным закрытием соединения
CLOSING	Обе стороны инициировали закрытие соединения одновременно: после отправки сегмента с флагом FIN узел-1 также получает сегмент FIN, отправляет ACK и находится в ожидании сегмента ACK (подтверждения на свой запрос о разъединении)

Биты	**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**
0-31	IP-адрес отправителя (Source address)																															
32-63	IP-адрес получателя (Destination address)																															
64-95	0	0	0	0	0	0	0	0	Протокол (Protocol)								Длина TCP-сегмента (TCP length)															

Формат пакета и использование ICMP

Октет	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—3	Тип								Код								Контрольная сумма																
…	Данные (формат зависит от значений полей «Код» и «Тип»)																															

- ICMP-сообщения (тип 12) генерируются при нахождении ошибок в заголовке IP-пакета (за исключением самих ICMP-пакетов, дабы не привести к бесконечно растущему потоку ICMP-сообщений об ICMP-сообщениях).
- ICMP-сообщения (тип 3) генерируются маршрутизатором при отсутствии маршрута к адресату.
- Утилита Ping, служащая для проверки возможности доставки IP-пакетов, использует ICMP-сообщения с типом 8 (эхо-запрос) и 0 (эхо-ответ).
- Утилита Traceroute, отображающая путь следования IP-пакетов, использует ICMP-сообщения с типом 11.
- ICMP-сообщения с типом 5 используются маршрутизаторами для обновления записей в таблице маршрутизации отправителя.
- ICMP-сообщения с типом 4 используются получателем (или маршрутизатором) для управления скоростью отправки сообщений отправителем.
- При потере ICMP-пакета никогда не генерируется новый.
- ICMP-пакеты никогда не генерируются в ответ на IP-пакеты с широковещательным или групповым адресом, чтобы не вызывать перегрузку в сети (так называемый «широковещательный шторм»).
- При повреждении фрагментированного IP-пакета ICMP-сообщение отправляется только после получения первого повреждённого фрагмента, поскольку отправитель всё равно повторит передачу всего IP-пакета целиком.

Значения полей типа и причины ICMP

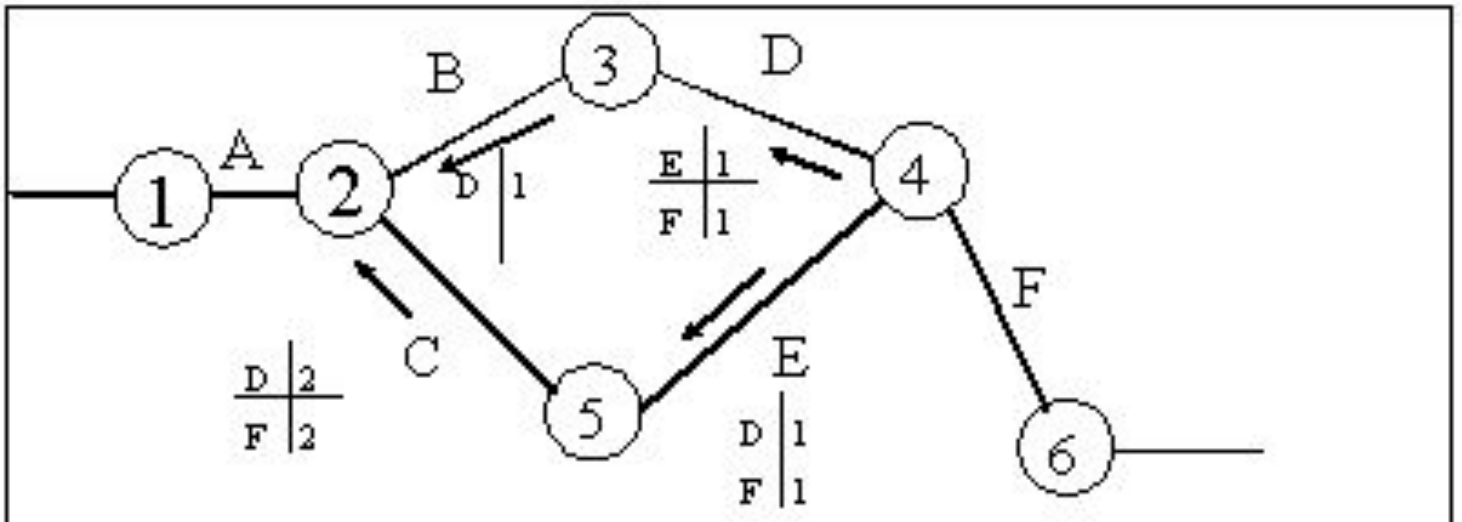
Поле типа может иметь следующие значения:

Значение	Тип сообщения
0	Эхо-ответ (Echo Replay)
3	Узел назначения недостижим (Destination Unreachable)
4	Подавление источника (Source Quench)
5	Перенаправление маршрута (Redirect)
8	Эхо-запрос (Echo Request)
11	Истечение времени дейтаграммы (Time Exceeded for a Datagram)
12	Проблема с параметром пакета (Parameter Problem on a Datagram)
13	Запрос отметки времени (Timestamp Request)
14	Ответ отметки времени (Timestamp Replay)
17	Запрос маски (Address Mask Request)
18	Ответ маски (Address Mask Replay)

Причина кодируется следующим образом:

Код	Причина
0	Сеть недостижима
1	Узел недостижим
2	Протокол недостижим
3	Порт недостижим
4	Требуется фрагментация, а бит DF установлен
5	Ошибка в маршруте, заданном источником
6	Сеть назначения неизвестна
7	Узел назначения неизвестен
8	Узел-источник изолирован
9	Взаимодействие с сетью назначения административно запрещено
10	Взаимодействие с узлом назначения административно запрещено
11	Сеть недостижима для заданного класса сервиса
12	Узел недостижим для заданного класса сервиса

Обмен маршрутной информацией по RIP



Начальная информация в узле 2:

Сеть	Расстояние	Сосед
A	1	-
B	1	-
C	1	-

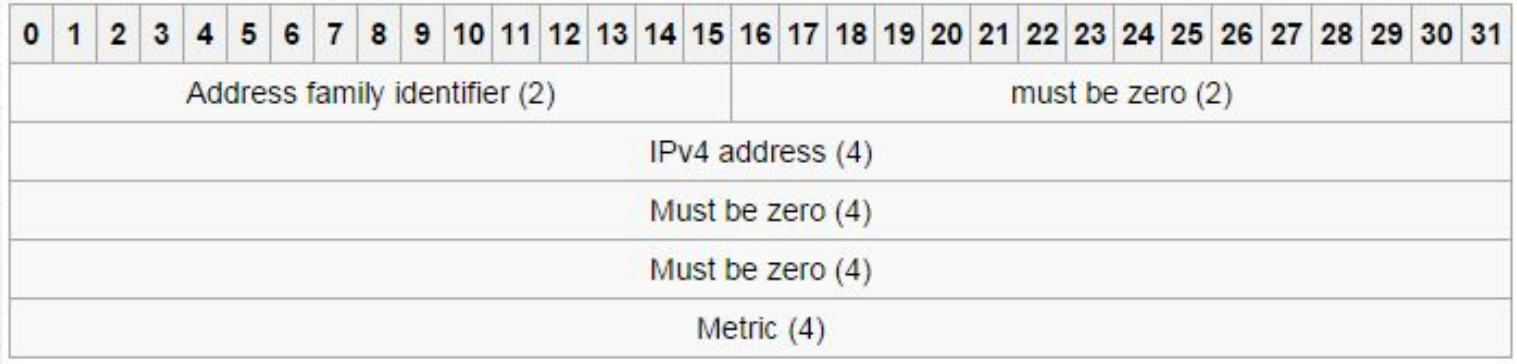
После 2-х шагов:

A	1	-
B	1	-
C	1	-
D	2	3
E	2	5
D	3	5
F	3	5

Формат пакета RIP

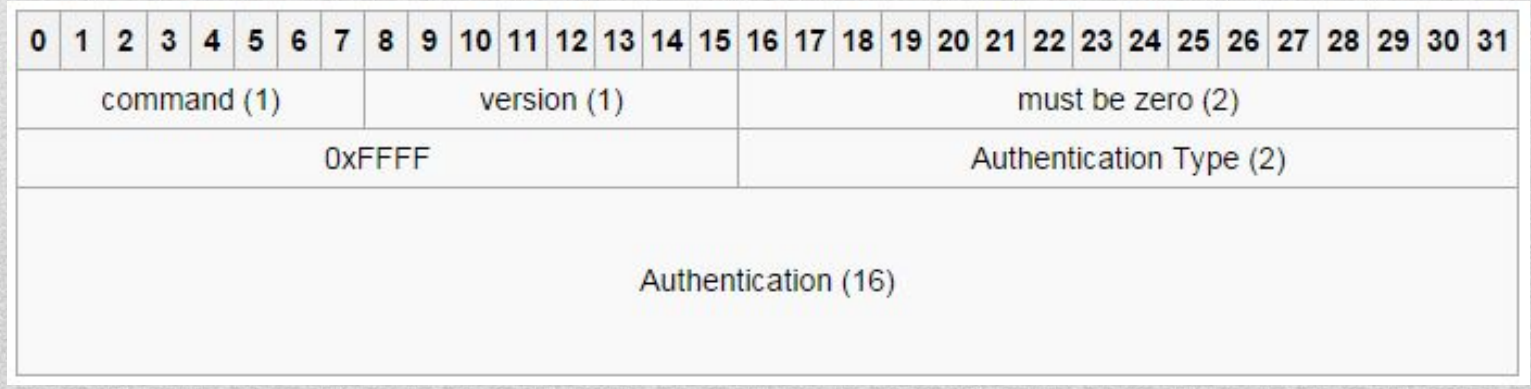
- Command — команда, определяет назначение датаграммы (1 — request; 2 — response)
- Version — номер версии, в зависимости от версии, определяется формат пакета
- Routing Domain — идентификатор RIP-системы, к которой принадлежит данное сообщение; часто — номер автономной системы. Используется, когда к одному физическому каналу подключены маршрутизаторы из нескольких автономных систем, в каждой автономной системе поддерживается своя таблица маршрутов. Поскольку сообщения RIP рассылаются всем маршрутизаторам, подключенным к сети, требуется различать сообщения, относящиеся к «своей» и «чужой» автономным системам. Поле использовалось короткое время в версии протокола RIP-2. В протоколе RIP-1 и в текущей версии RIP-2 не используется.
- RIP Entry (RTE) — запись маршрутной информации RIP. RIP пакет может содержать от 1 до 25 записей RIP Entry.

Формат пакета RIP Entry для RIP-1, аутентификация



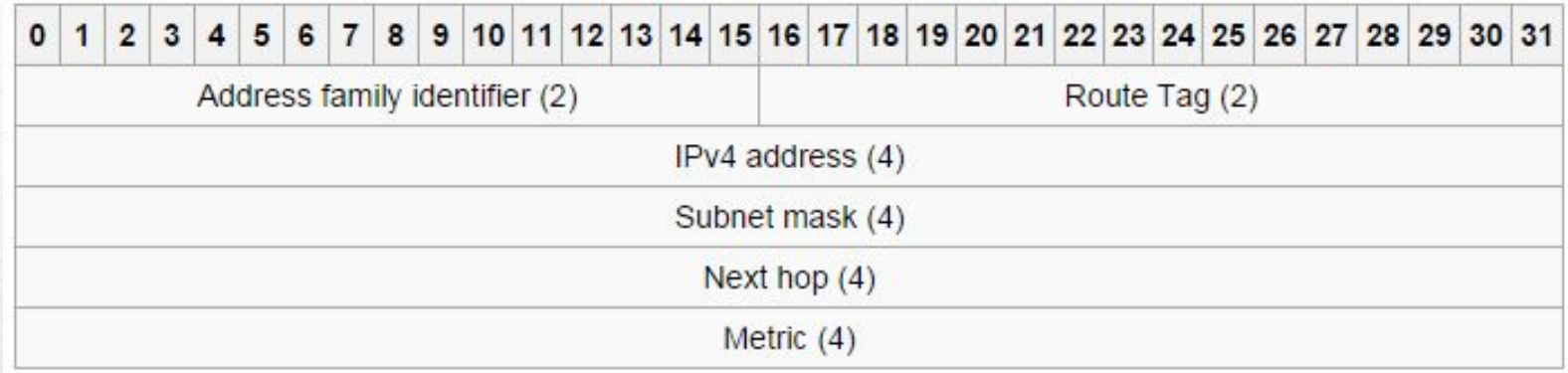
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Address family identifier (2)																must be zero (2)															
IPv4 address (4)																															
Must be zero (4)																															
Must be zero (4)																															
Metric (4)																															

- Address family identifier (AFI) — тип адреса, обычно поддерживается только запись AF_INET, которое равно 2 (т. е. используется для протокола IP).
- Must be zero — должно быть нулём.
- IPv4 address — IP адрес места назначения (хост или сеть)
- Metric — метрика маршрута



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
command (1)								version (1)								must be zero (2)															
0xFFFF																Authentication Type (2)															
Authentication (16)																															

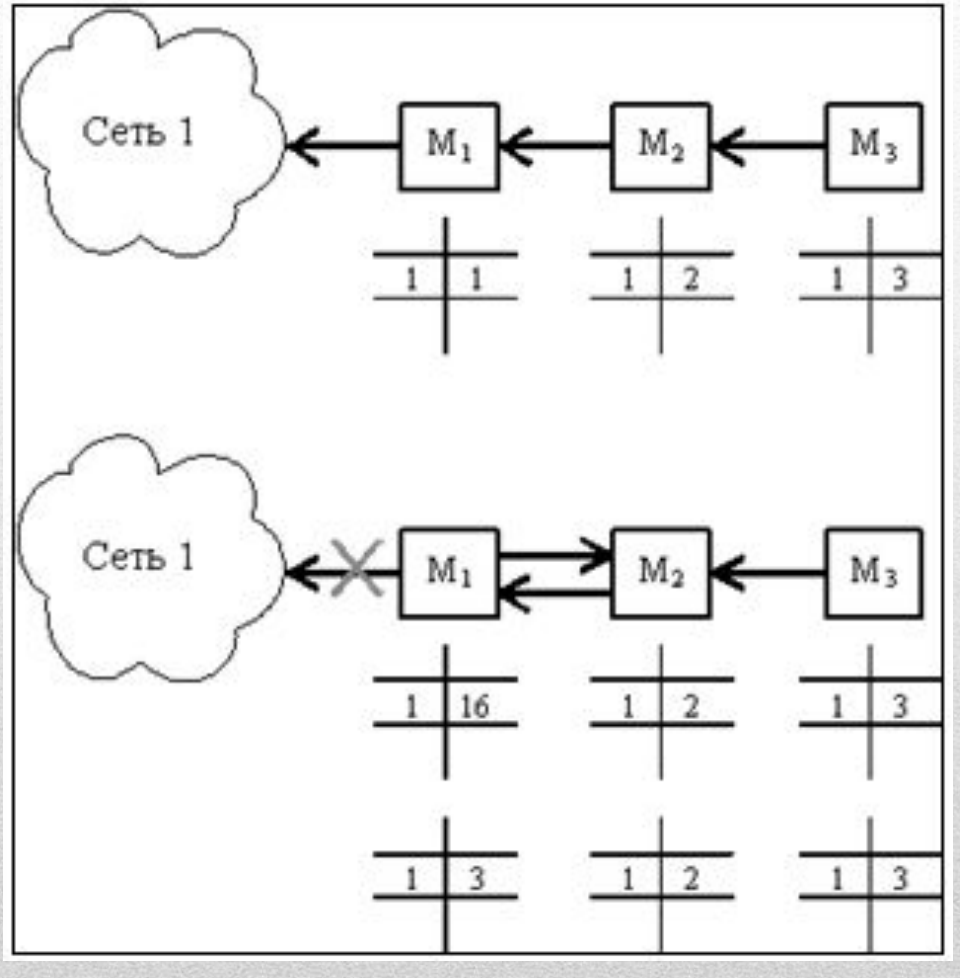
Формат пакета RIP Entry для RIP-2



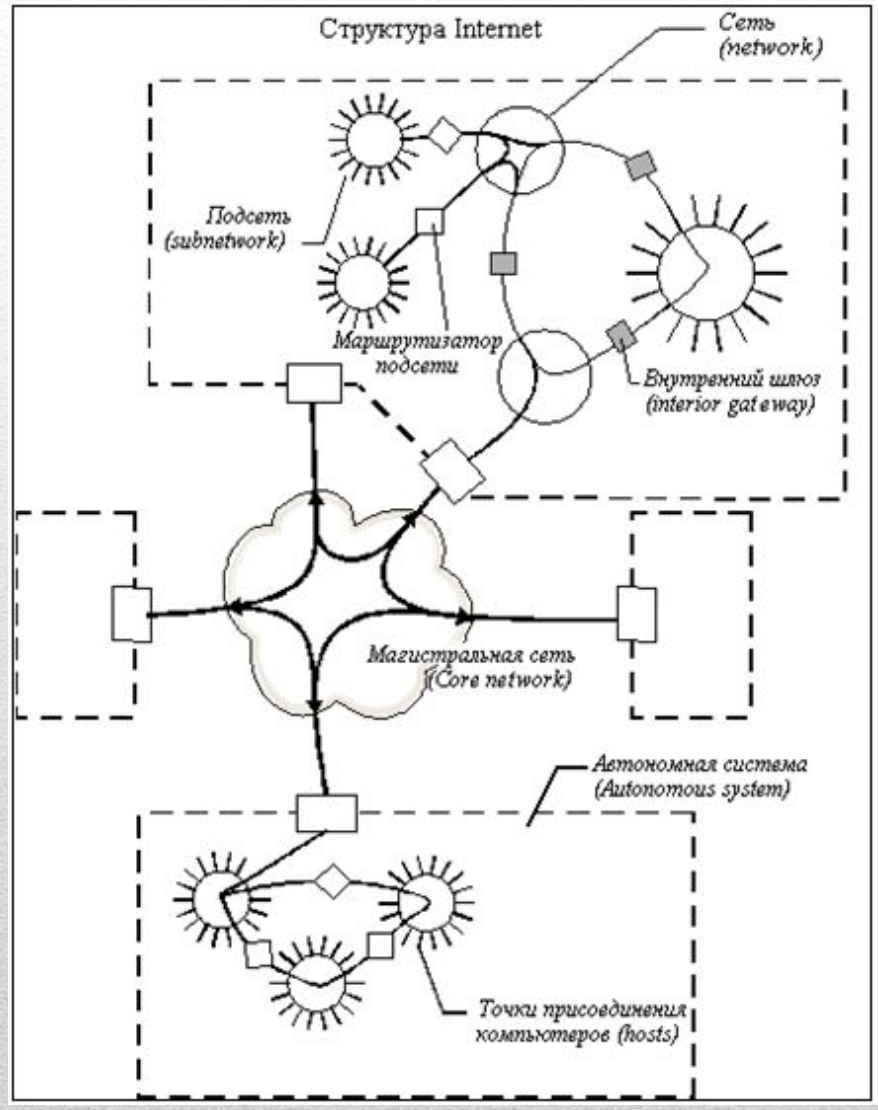
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Address family identifier (2)																Route Tag (2)															
IPv4 address (4)																															
Subnet mask (4)																															
Next hop (4)																															
Metric (4)																															

- Address Family Identifier (AFI) — тип адреса, обычно поддерживается только запись AF_INET, которое равно 2 (т.е. используется для протокола IP).
- Route Tag (RT) — тег маршрута. Предназначен для разделения «внутренних» маршрутов от «внешних», взятых, например, из другого IGP или EGP.
- IP Address — IP адрес места назначения.
- Subnet Mask — маска подсети
- Next Hop — следующий хоп. Содержит IP адрес маршрутизатора к месту назначения. Значение 0.0.0.0 — хопом к месту назначения является отправитель пакета. Необходимо, если протокол RIP не может быть запущен на всех маршрутизаторах.
- Metric — метрика маршрута.

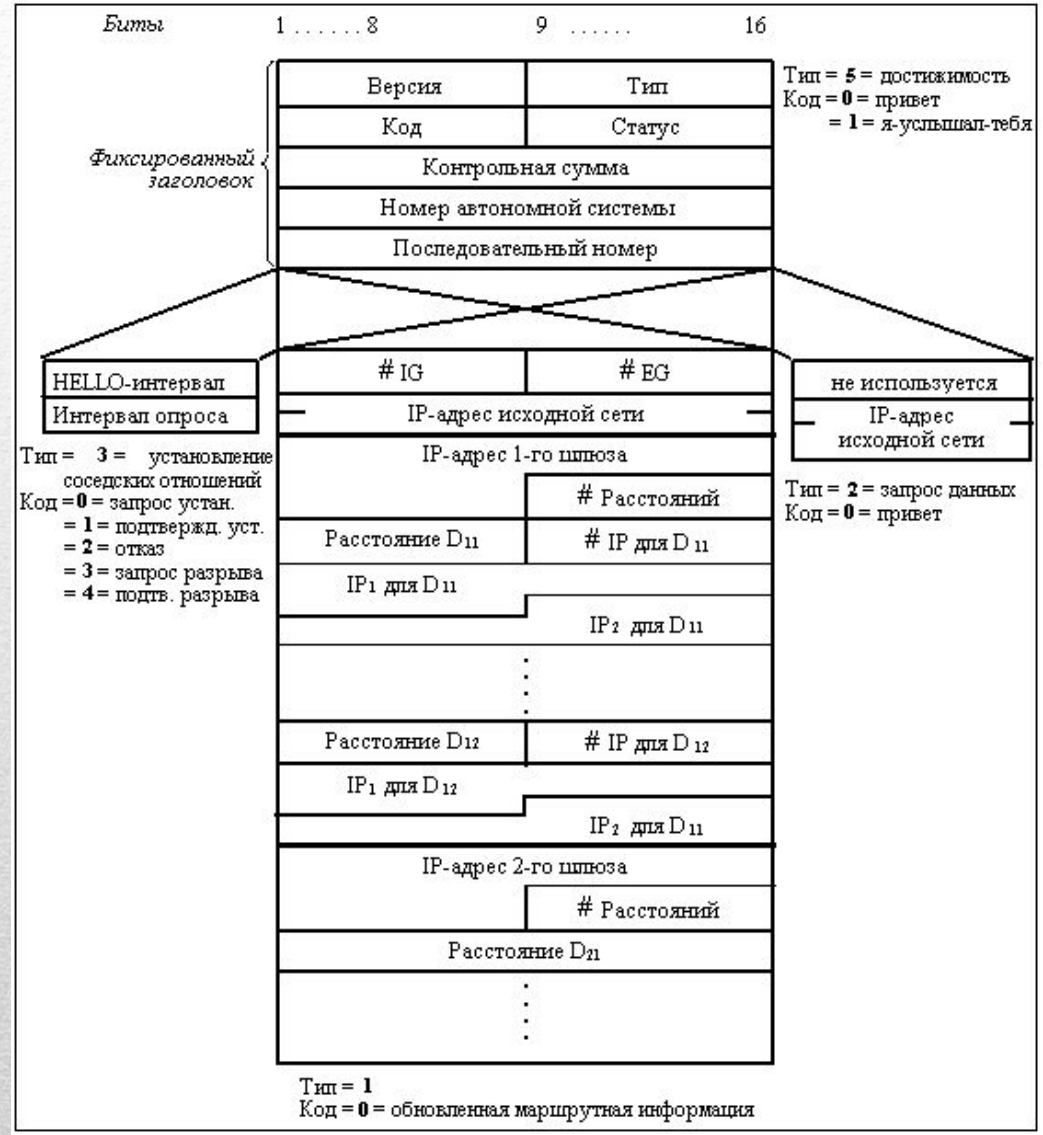
Неустойчивая работа RIP



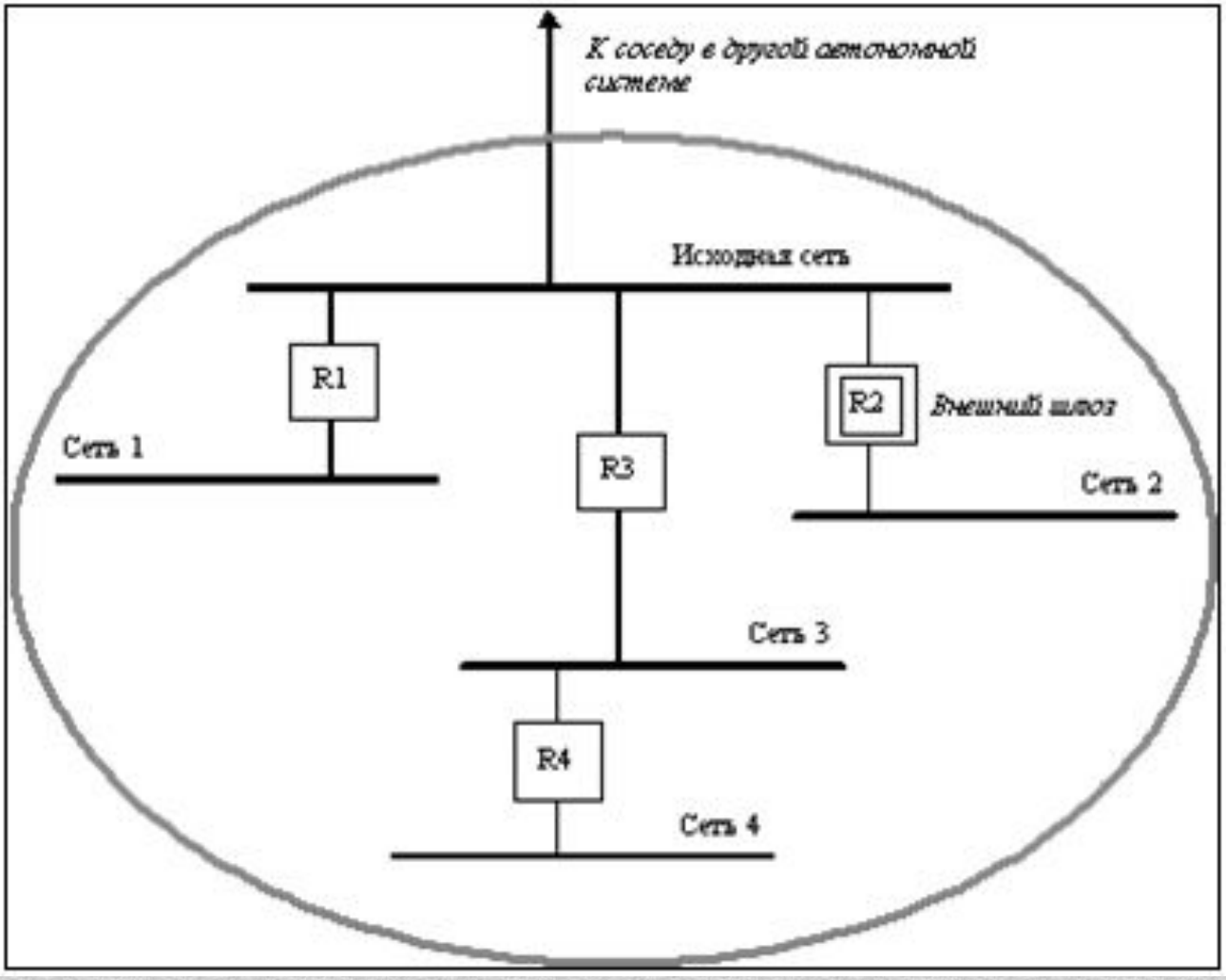
Архитектура сети Internet



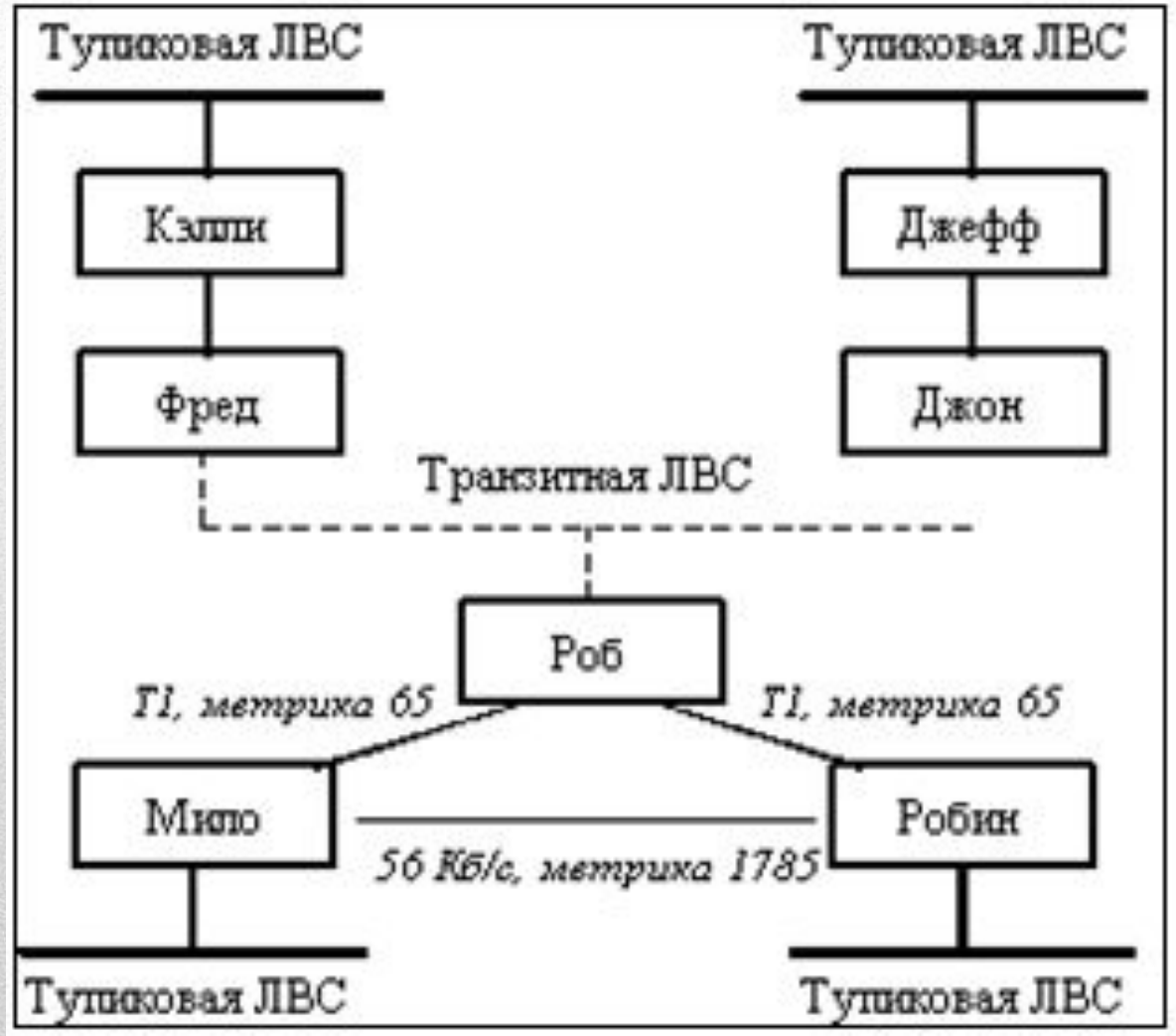
Формат сообщения протокола EGP



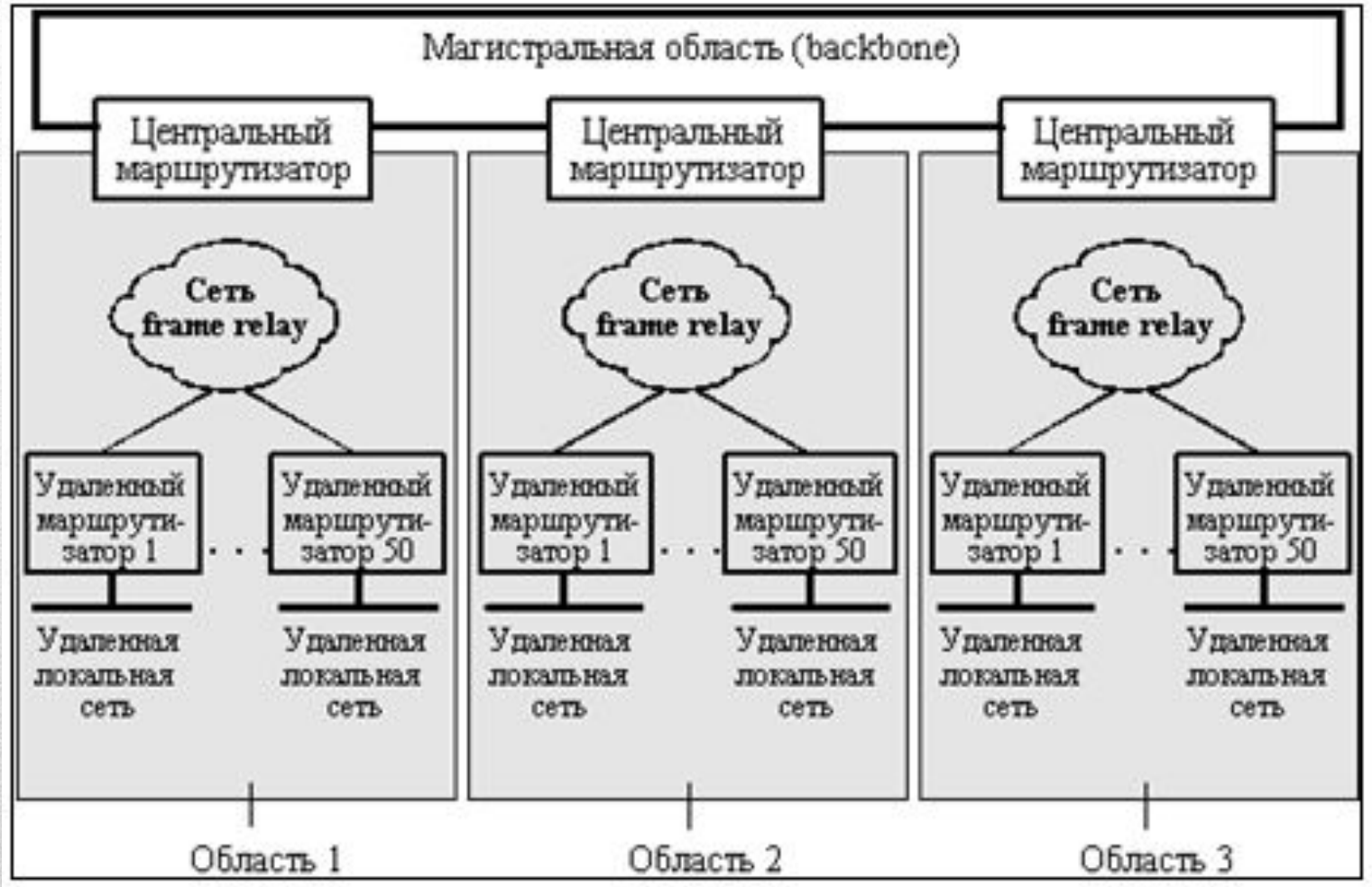
Пример автономной системы



Гипотетическая сеть с OSPF маршрутизаторами



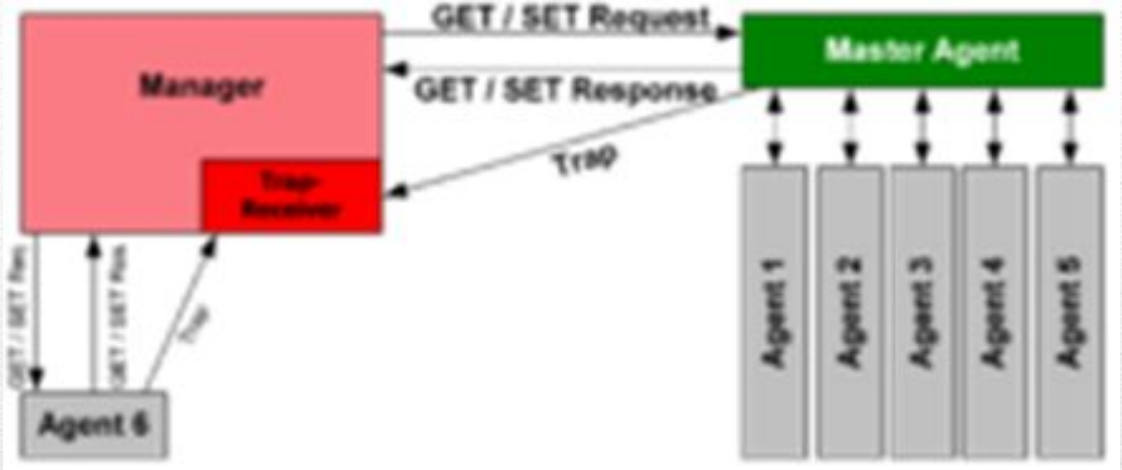
Большая сеть с топологией звезда



Simple Network Management Protocol

- SNMP (Simple Network Management Protocol) — стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP.
- Поддерживающие SNMP устройства – маршрутизаторы, коммутаторы, серверы, рабочие станции, принтеры, модемные стойки и другие.
- Протокол используется в системах сетевого управления для контроля подключенных к сети устройств на предмет условий, которые требуют внимания администратора.
- SNMP определен Инженерным советом интернета (IETF) как компонент TCP/IP. Он состоит из набора стандартов для сетевого управления, включая протокол прикладного уровня, схему баз данных и набор объектов данных.
- SNMP предоставляет данные для управления в виде переменных, описывающих конфигурацию управляемой системы. Переменные могут быть запрошены (иногда заданы) управляющими приложениями.

Принципы SNMP коммуникаций



- Управляемые протоколом SNMP сети состоят из трех ключевых компонентов:
- управляемое устройство;
- агент — программное обеспечение, запускаемое на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства;
- система сетевого управления (Network Management System, NMS) — программное обеспечение, взаимодействующее с менеджерами для поддержки комплексной структуры данных, отражающей состояние сети.

Базы управляющей информации (MIB)

- Адреса объектов устройств определяются в цифровом формате, для упрощения применяется базы управляющей информации (MIB).
- Базы MIB описывают структуру управляемых данных на подсистеме устройства; они используют иерархическое пространство имен, содержащее идентификаторы объектов (OID-ы).
- Каждый OID состоит из двух частей: текстового имени и SNMP адреса в цифровом виде.
- Базы MIB являются необязательными и выполняют вспомогательную роль по переводу имени объекта из человеческого формата (словесного) в формат SNMP (цифровой, похоже на DNS сервера).
- Так как структура объектов на устройствах разных производителей не совпадает, без базы MIB невозможно определить цифровые SNMP адреса нужных объектов. Базы MIB используют нотацию, заданную в ASN.1

Детали протокола SNMP

- SNMP работает на прикладном уровне TCP/IP (7 уровень модели OSI).
- Агент SNMP получает запросы по UDP-порту 161.
- Менеджер может посылать запросы с любого доступного порта источника на порт агента.
- Ответ агента будет отправлен назад на порт источника на менеджере.
- Менеджер получает уведомления (Traps и InformRequests) по порту 162.
- Агент может генерировать уведомления с любого доступного порта, при использовании TLS или DTLS запросы получаются по порту 10161, а ловушки отправляются на порт 10162.
- В SNMPv1 указано пять основных протокольных единиц обмена (protocol data units - PDU). Еще две PDU, GetBulkRequest и InformRequest, введены в SNMPv2 и перенесены в SNMPv3.
- Все PDU протокола SNMP построены следующим образом:

IP header (IP-заголовок)	UDP header (UDP-заголовок)	version (версия)	community (пароль)	PDU-type (PDU-тип)	request-id (id запроса)	error-status (статус ошибки)	error-index (индекс ошибки)	variable bindings (связанные переменные)

Семь протокольных единиц обмена SNMP (четыре первых)

- GetRequest - Запрос от менеджера к объекту для получения значений переменной или списка переменных. Переменные указываются в поле variable bindings (values не используется). Получение значений переменной выполнено агентом как Атомарная операция. Менеджеру возвращен Response (ответ) с текущими значениями.
- SetRequest - Запрос от менеджера к объекту для изменения переменной или списка переменных. Связанные переменные указываются в теле запроса. Изменения переменных выполнены агентом как атомарная операция. Менеджеру возвращен Response с (текущими) новыми значениями переменных.
- GetNextRequest - Запрос от менеджера к объекту для обнаружения доступных переменных и их значений. Менеджеру возвращен Response со связанными переменными для переменной, которая является следующей в базе MIB в лексиграфическом порядке. Обход всей базы MIB агента произведен итерационным использованием GetNextRequest, начиная с OID 0. Строки таблицы прочтены, если указать в запросе OID колонок в связанных переменных.
- GetBulkRequest - Улучшенная версия GetNextRequest. Запрос от менеджера к объекту для итераций GetNextRequest. Менеджеру возвращен Response с несколькими связанными переменными, обойденными начиная со связанной переменной (переменных) в запросе. Специфичные для PDU поля non-repeaters и max-repetitions используются для контроля за поведением ответа. GetBulkRequest введен в SNMPv2.

Семь протокольных единиц обмена SNMP (три последних)

- Response - Возвращает связанные переменные и значения от агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest и InformRequest. Уведомления об ошибках обеспечиваются полями статуса ошибки и индекса ошибки. Эта единица использовалась как ответ и на get-, и на set-запросы, но названа GetResponse в SNMPv1.
- Trap - Асинхронное уведомление от агента - менеджеру. Включает текущее значение sysUpTime, OID, определяющий тип trap (ловушки), и необязательные связанные переменные. Адресация получателя для ловушек определяется переменными trap-конфигурации в MIB. Формат trap-сообщения изменен в SNMPv2 и PDU, в SNMPv2-Trap.
- InformRequest - Асинхронное уведомление от менеджера - менеджеру или от агента - менеджеру. Уведомления от менеджера - менеджеру возможны в SNMPv1 (с помощью Trap), но SNMP работает на протоколе UDP - доставка не гарантирована, не сообщается о потерянных пакетах. InformRequest уже отправляет назад подтверждение о получении. Получатель отвечает Response, повторяющим информацию из InformRequest. PDU введен в SNMPv2.

Сравнение версий SNMP

- SNMPv1, изначальная реализация протокола SNMP, работает с протоколами UDP, IP, CLNS, DDP и IPX, используется и де-факто является протоколом сетевого управления в Интернет-сообществе. SNMPv1 имеет низкую безопасность, аутентификация клиентов производилась с помощью «общей строки» (community string, пароля), которая передавалась в открытом виде.
- SNMPv2 включает улучшения производительности, безопасности, конфиденциальности и связях между менеджерами. Протокол ввел GetBulkRequest, альтернативу итерационному применению GetNextRequest для получения большого количества управляющих данных через один запрос. Система безопасности на основе сторон из SNMPv2 не получила распространение, как сложная. SNMPv2c несовместим с SNMPv1 в двух областях: форматы сообщений и операции протокола - сообщения SNMPv2c используют отличные форматы заголовка и протокольных единиц данных (PDU), а также две новые операции протокола.

Стратегии сосуществования SNMPv1/v2c

Прокси-агенты - агент SNMPv2 может действовать как прокси-агент от имени управляемых протоколом SNMPv1 устройств:

- система сетевого управления (Network management system, NMS) SNMPv2 выдает команды, предназначенные для SNMPv1-агента;
- NMS посылает SNMP-сообщение прокси-агенту SNMPv2;
- прокси-агент без изменения направляет сообщения Get, GetNext и Set агенту SNMPv1;
- сообщения GetBulk преобразуются прокси-агентом в сообщения GetNext, после чего направляются агенту SNMPv1;
- прокси-агент отображает trap-сообщения SNMPv1 в trap-сообщения SNMPv2, после чего направляет их NMS.

Двуязычные SNMPv2-NMS поддерживают SNMPv1 и SNMPv2.

- Для поддержки окружения с двойным управлением управляющее приложение в двуязычной NMS должно связаться с агентом.
- NMS анализирует хранящуюся в локальной базе данных информацию для определения, поддерживает агент SNMPv1 или SNMPv2.
- На основе полученной информации, NMS связывается с агентом, используя соответствующую версию SNMP.

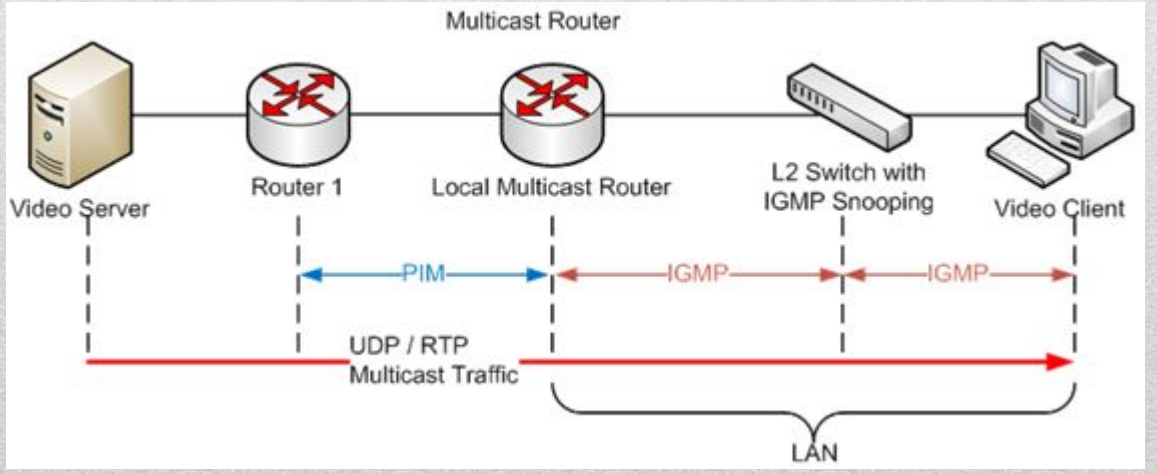
Стратегии сосуществования SNMPv1/v2c – 2 языка NMS

SNMPv3 добавляет в SNMP защиту и улучшения в удаленной настройке. Каждое SNMPv3-сообщение содержит параметры безопасности, которые закодированы как строка октетов. Значение параметров зависит от используемой модели безопасности. SNMPv3 предоставляет :

- конфиденциальность - шифрование пакетов для предотвращения перехвата несанкционированным источником;
- целостность - целостность сообщений, для предотвращения изменения пакета в пути, включая дополнительный механизм защиты от повторной передачи перехваченного пакета;
- аутентификацию - чтобы убедиться, что сообщение пришло из правильного источника.

Internet Group Management Protocol

- IGMP (Internet Group Management Protocol) — протокол управления групповой (multicast) передачей в сетях, основанной на протоколе IP, используется локальными маршрутизаторами и IP-узлами для организации сетевых устройств в группы.
- IGMP- часть спецификации групповой передачи пакетов в IPv4-сетях. расположен на сетевом уровне, может использоваться для потокового видео и онлайн-игр.
- IGMP уязвим к атакам, брандмауэры позволяют пользователю его отключить протокол, если в нем нет необходимости.



Стандарты и реализации IGMP

- Существует три версии IGMP: IGMPv1, IGMPv2 и IGMPv3.
- Улучшение в IGMPv3 относительно IGMPv2 – фильтрация IP-адресов, позволяющая узлу сообщить, с каких адресов он хочет получать пакеты.
- Протокол IGMP реализован в виде серверной и клиентской частей, первая выполняется на маршрутизаторе, вторая — в узле сети, получающем групповой трафик.
- Клиент посылает уведомление о принадлежности группе локальному маршрутизатору, маршрутизатор находится в ожидании уведомлений и периодически рассылает клиентам запросы.
- Между локальными и удаленными маршрутизаторами используется протокол Protocol Independent Multicast (PIM) для направления группового трафика от сервера к клиентам групповой передачи.
- Все ОС поддерживают клиентскую часть протокола, для серверной части IGMP используются процессы – IGMP-маршрутизаторы. XORP позволяет превратить компьютер в маршрутизатор групповой передачи.

Запросы принадлежности IGMPv3

Запросы принадлежности (Membership Query Message) рассылаются маршрутизаторами, чтобы для каждого узла определить его принадлежность к каким-либо группам (group membership state) и список источников информации, от которых данный узел хочет получать сообщения (reception state). Есть три типа запросов:

- Общие запросы (General Queries) — позволяют получить полную информацию для каждого из узлов. Маршрутизатор периодически рассылает эти запросы всем системам, подключенным к его сети.
- Запросы с указанием группы (Group-Specific Queries) — используются для определения состояния подписки для заданной группы узлов. Такие запросы рассылаются по соответствующему групповому адресу.
- Запросы с указанием группы и источника (Group-and-Source-Specific Queries) — позволяет для каждого узла заданной группы определить, какие сообщения из всех, посылаемых заданными источниками, этот узел хочет получать..

Структура пакетов запросов принадлежности IGMPv3

Октет	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—3	Type = 0x11								Max Resp Code								Checksum															
4—7	Group Address																															
8—11	Resv				S		QRV		QQIC								Number of Sources (N)															
12—15	Source Address [1]																															
...	...																															
	Source Address [N]																															

- Код макс. ответа (Max Resp Code) — максимальное время (в 1/10 секунды) ожидания ответа для запросу. Значение < 128 используется напрямую, значение ≥ 128 интерпретируется как экспонента с мантиссой.
- Контрольная сумма (Checksum) — 16-битная контрольная сумма для всего IGMP-сообщения.
- Групповой адрес (Group Address) — групповой адрес в запросах с указанием группы. При общем запросе поле устанавливается равным нулю.
- Resv — поле зарезервировано, обнуляется при посылке и игнорируется при получении.

Структура пакетов запросов принадлежности IGMPv3

- Флаг S (Прекратить серверную обработку, Suppress Router-side Processing) — установка флага указывает маршрутизаторам, получившим данное сообщение, прекратить обновления по таймеру.
- QRV (Переменная надежности запрашивающего, Querier's Robustness Variable) — содержит переменную надежности (Robustness Variable), используемую. посылающим устройством. Маршрутизаторы обновляют переменные надежности в соответствии с последним полученным запросом, пока это поле ненулевое.
- QQIC (Код интервала запроса, Querier's Query Interval Code) — значение поля указывает интервал между запросами (Query Interval). Значение < 128 используется напрямую, значение ≥ 128 интерпретируется как экспонента с мантиссой.
- Количество источников (Number of Sources, N) — определяет число адресов источников, присутствующих в запросе. Для общих запросов и запросов с указанием группы это значение равно нулю. Для запросов с указанием группы и источника это поле ненулевое, ограничено значением MTU сети.
- Адрес источника [i] (Source Address) — массив индивидуальных IP-адресов источников данных.

Структура отчётов принадлежности IGMPv3

Октет	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—3	Type = 0x22								Reserved								Checksum															
4—7	Reserved																Number of Group Record (M)															
8—11	Group Record [1]																															
...	...																															
	Group Record [M]																															

- Reserved — устанавливается в ноль при передаче и игнорируется при приёме;
- Number of Group Record — количество полей Group Record в сообщении;
- Group Record — блок полей с информацией о членстве отправителя в группе, имеющий вид:

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
Record Type								Aux Data Len								Number of Sources (N)															
Multicast Address																															
Source Address [1]																															
...																															
Source Address [N]																															
Auxilary Data																															

Структура отчётов принадлежности IGMPv3

- Record Type — тип записи:
  - Текущее состояние — посылается в ответ на запрос, сообщает о текущем режиме фильтрации, относительно указанного группового адреса, принимает значения MODE_IS_INCLUDE и MODE_IS_EXCLUDE;
  - Изменение режима — посылается при изменении режима фильтрации, принимает значения CHANGE_TO_INCLUDE_MODE и CHANGE_TO_EXCLUDE_MODE;
  - Изменение списка источников — посылается при изменении списка источников без изменения режима фильтрации:
    - ALLOW_NEW_SOURCES — в режиме INCLUDE адреса добавляются к списку, в режиме EXCLUDE — удаляются из списка;
    - BLOCK_OLD_ SOURCES — в режиме EXCLUDE адреса добавляются к списку, в режиме INCLUDE — удаляются из списка.
- Aux Data Len — длина дополнительных данных в 32-битных словах.
- Number of Sources — количество адресов источников данных;
- Multicast Address — групповой адрес, к которому относится информация в записи;
- Source Address — массив индивидуальных IP-адресов источников данных;
- Auxilary Data — дополнительная информация, не должна использоваться в текущей версии протокола.

Структура других пакетов IGMPv3

Тип	Наименование	Описание
0x12	Version 1 Membership report	Должны поддерживаться для совместимости с предыдущими версиями
0x16	Version 2 Membership report	
0x17	Version 2 Leave Group	
0x13	Distance Vector Multicast Routing Protocol	Экспериментальный протокол

Address Resolution Protocol

- ARP (Address Resolution Protocol) — протокол определения MAC-адреса по известному IP-адресу.
- Хосты А (IP-адрес 10.0.0.1) и Б (IP-адрес 10.22.22.2) соединены Ethernet. А желает переслать пакет на Б, IP-адрес Б ему известен. Ethernet не работает с IP-адресами, поэтому А для передачи через Ethernet нужен MAC-адрес Б в Ethernet.
  - Стек IP А проверяет кэш ARP на наличие зарегистрированной в нём информации об узле-получателе Б.
  - Если такой записи нет, А отправляет широковещательный запрос ARP-request всем компьютерам в одном с ним сегменте Ethernet: «Хост с IP-адресом 10.22.22.2, сообщите свой MAC-адрес хосту с IP-адресом 10.0.0.1».
  - Ethernet доставляет ARP-request всем устройствам в том же сегменте Ethernet, в том числе и Б.
  - Б отвечает А на ARP-request ответом ARP-reply, где сообщает свой MAC-адрес (напр. 00:ea:d1:11:f1:11)
  - Получив в ARP-reply MAC-адрес Б, А обновит свой кэш ARP и может передавать ему данные через Ethernet.
- Распространение ARP получил в сетях IP, построенных поверх Ethernet – в 100 % случаев при таком сочетании используется ARP.
- В семействе протоколов IPv6 ARP не существует, его функции возложены на ICMPv6.

Address Resolution Protocol , вариации ARP

- Записи в кэше ARP статические и динамические. Пример выше описывает динамическую запись кэша. Можно создавать статические записи командой:
  - arp -s <IP-адрес> <MAC-адрес>
- Динамические записи в таблице ARP остаются в кэше в течение 2-х минут. Если за это время произошла повторная передача данных по этому адресу, то время хранения записи в кэше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кэше просуществует до 10 минут.
- После этого запись будет удалена из кэша, и будет отправлен повторный запрос ARP..
- В действительности время хранения записей в ARP таблице и метод хранения выбирается программно (операционной системой), при желании его можно изменить.
- ARP в основном используется для сопоставления IP- и MAC-адресов, но его можно использовать для разрешения MAC-адресов для различных адресов протоколов 3-го уровня (Layer 3 protocols addresses).
- ARP был адаптирован также для разрешения других видов адресов 2-го уровня (Layer 2 addresses); например, ATMARP используется для разрешения ATM NSAP-адресов в Classical IP over ATM протоколе.

InARP, RARP

- Inverse Address Resolution Protocol (Inverse ARP или InARP) — протокол для получения адресов сетевого уровня (IP адресов) других рабочих станций по их адресам канального уровня (часто DLCI в Frame Relay сетях), используется в Frame Relay и ATM сетях.
- ARP переводит адреса сетевого уровня в адреса канального уровня, в то же время InARP можно рассматривать как инверсию ARP.
- InARP реализован как расширение ARP. Форматы пакетов этих протоколов одни и те же, различаются лишь коды операций и заполняемые поля.
- Reverse ARP (RARP), как и InARP, переводит адреса канального уровня в адреса сетевого уровня и используется для получения логических адресов самих станций отправителей, в то время как в InARP отправитель знает свои адреса и запрашивает логический адрес другой станции.
- От RARP отказались в пользу BOOTP, который был в свою очередь заменён DHCP.

Структура пакета ARP

+	Bits 0 — 7	8 — 15	16 — 31
0	Hardware type (HTYPE)		Protocol type (PTYPE)
32	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
64	Sender hardware address (SHA)		
?	Sender protocol address (SPA)		
?	Target hardware address (THA)		
?	Target protocol address (TPA)		

- В Ethernet в ARP-пакетах используется EtherType 0x0806, пакеты рассылаются широковещательно MAC-адрес — FF:FF:FF:FF:FF:FF. В качестве SHA, SPA, THA, & TPA используются 32-битные слова — реальная длина определяется устройством и протоколом.
  - Hardware type (HTYPE) – Канальный протокол передачи данных имеет свой номер, который хранится в этом поле. Например, Ethernet имеет номер 0x0001.
  - Protocol type (PTYPE) - Код сетевого протокола. Для IPv4 будет записано 0x0800.
  - Hardware length (HLEN) - Длина физического адреса в байтах. Адреса Ethernet - 6 байт.
  - Protocol length (PLEN) - Длина логического адреса в байтах. IPv4 адреса - 4 байта.
  - Operation - Код операции отправителя: 1 в случае запроса и 2 в случае ответа.
  - Sender hardware address (SHA) - Физический адрес отправителя.
  - Sender protocol address (SPA) - Логический адрес отправителя.
  - Target hardware address (THA) - Физический адрес получателя. Поле пусто при запросе.
  - Target protocol address (TPA) - Логический адрес получателя.

Serial Line Internet Protocol SLIP

- **SLIP ()** — протокол канального уровня для доступа через низкоскоростные линии и инкапсуляцию IP-пакетов, использующий коммутируемые соединения последовательных портов типа точка-точка.
- Для установления связи необходимо задать IP-адреса, т.к. в SLIP нет системы обмена адресной информацией.
- В принимаемом потоке бит SLIP определяет признаки начала и конца пакета, по которым собирает IP-пакеты и передаёт верхнему уровню.
- При отправке IP-пакеты переформатируются и посимвольно отправляются получателю через последовательную линию.
- Для передачи используется конфигурация UART: 8 бит данных (8 data bits), без паритета (no parity), аппаратное управление каналом передачи (EIA hardware flow control) или трёхпроводный нуль-модемный кабель (3-wire null-modem — CLOCAL mode).
- Передача данных байт-ориентированная – IP-пакет разбивается на байты. Границей SLIP-кадра является уникальный флаг END (0xC0), что поддерживается байт-стаффингом (byte stuffing) внутри кадра с ESC-последовательностью 0xDB, байт END (0xC0) заменяется последовательностью (0xDB, 0xDC), а байт ESC (0xDB) — последовательностью (0xDB, 0xDD).
- Недостатки: отсутствие адресации, индексации, коррекции, компрессии.

Point-to-Point Protocol

- PPP (Point-to-Point Protocol) — двухточечный протокол канального уровня (Data Link) модели OSI на основе HDLC, используемый для установления прямой связи между двумя узлами сети с возможностью аутентификации соединения, шифрования и сжатия данных. PPP PPP.
- PPP используется на многих типах физических сетей: нуль-модемный кабель, телефонная линия, сотовая связь и т. д.
- Подвиды протокола PPP :
  - Point-to-Point Protocol over Ethernet (PPPoE) для подключения по Ethernet, иногда через DSL;
  - Point-to-Point Protocol over ATM (PPPoA) для подключения по ATM Adaptation Layer 5 (AAL5), как альтернатива PPPoE для DSL.
- PPP - это семейство протоколов:
  - протокол управления линией связи (LCP);
  - протокол управления сетью (NCP);
  - протоколы аутентификации (PAP, CHAP);
  - многоканальный протокол PPP (MLPPP).

Автоматическая настройка PPP с помощью LCP

- Link Control Protocol (LCP) обеспечивает автоматическую настройку интерфейсов на каждом конце и опционально – аутентификацию.
- LCP устанавливает и завершает соединения, позволяя узлам определять настройки соединения, поддерживает и байто- и бито-ориентированные кодировки и работает поверх PPP – PPP-сеть должна быть до LCP.
- LCP обнаруживает закольцованные связи, используя «magic numbers» в составе LCP-сообщений. Если линия закольцована, узел получает сообщение LCP с его собственным магическим числом вместо получения сообщения с магическим числом клиента.
- Challenge-handshake authentication protocol (CHAP) - предпочтительный для соединений с провайдерами, иногда используется устаревший Password authentication protocol (PAP), другой вариант аутентификации через PPP - Extensible Authentication Protocol (EAP).
- С помощью Internet Protocol Control Protocol (IPCP) поверх установленного PPP-соединения настраивается дополнительная сеть.
- Internet Protocol Version 6 Control Protocol (IPv6CP) получит большее распространение в будущем, когда IPv6 заменит IPv4.

Особенности и конфигурационные опции PPP

- Управляемые LCP параметры:
  - **Аутентификация -** использует Challenge Handshake Authentication Protocol (CHAP) предпочтительный для проведения аутентификации в PPP или реже Password Authentication Protocol (PAP), другой вариант для аутентификации – Extensible Authentication Protocol (EAP).
  - **Сжатие –** увеличивает пропускную способность PPP соединения за счет сжатия данных в кадре, известными алгоритмами сжатия PPP кадров являются Stacker и Predictor.
  - **Обнаружение ошибок** – включает Quality-Protocol и помогает выявить петли обратной связи посредством Magic Numbers.
  - **Многоканальность –** Multilink PPP (MLPPP, MPPP, MLP) предоставляет методы для распространения трафика через несколько физических каналов, при одном логическом соединении, позволяет расширить пропускную способность и обеспечивает балансировку нагрузки.

Многопротокольная поддержка NCP

- PPP позволяет работать нескольким протоколам сетевого уровня на одном канале связи – внутри PPP-соединения могут передаваться потоки данных различных сетевых протоколов (IP, IPX и т. д.), а также данные протоколов канального уровня локальной сети.
- Для каждого сетевого протокола используется Network Control Protocol (NCP) который его конфигурирует (согласовывает некоторые параметры протокола).
- NCP используется для определения настроек сетевого уровня, таких как сетевой адрес или настройки сжатия, после того как соединение было установлено.

Кадр PPP

- Кадр PPP начинается и завершается флагом 0x7E. Затем следует байт адреса и байт управления, равные 0xFF и 0x03 соответственно. В связи с вероятностью совпадения байтов внутри блока данных с зарезервированными флагами существует система автоматической корректировки «проблемных» данных с последующим восстановлением.

Флаг 0x7E	Адрес 0xFF	Управление 0x03	Данные	Контрольная сумма	Флаг 0x7E
1	1	1	1494	2	1

- Поля «Флаг», «Адрес» и «Управление» (заголовок кадра HDLC) могут быть опущены и не передаваться, но это если PPP в процессе конфигурирования (используя LCP) договорится об этом. Если PPP инкапсулирован в L2TP-пакеты, то поле «Флаг» не передается.

Типы кадров PPP

- Поле «Данные» PPP кадра разбито ещё на два поля:
  - флаг протокола, который определяет тип данных до конца кадра;
  - сами данные.

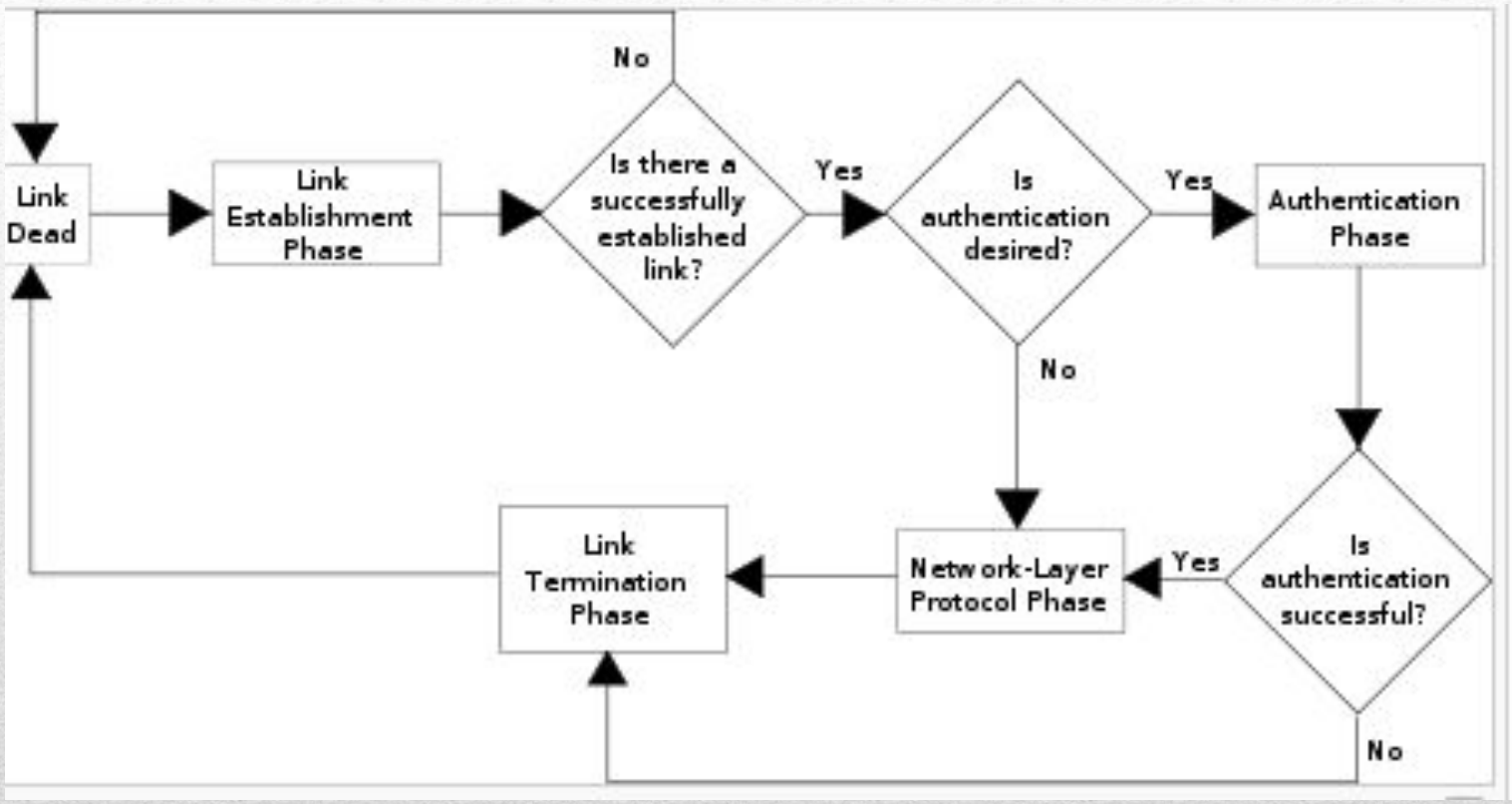
Протокол 0xXXXX	Данные
1 или 2	0 и более

- Флаги протокола от 0x0XXX до 0x3XXX идентифицируют протоколы сетевого уровня (IP протоколу соответствует флаг 0x0021).
- Флаги протокола от 0x4XXX до 0x7XXX идентифицируют протоколы с низким уровнем трафика.
- Флаги протокола от 0x8XXX до 0xBXXX идентифицируют протоколы управления сетью (NCP).
- Флаги протокола от 0xCXXX до 0xEXXX идентифицируют управляющие протоколы, так 0xC021 обозначает, что кадр содержит данные протокола управления соединением LCP.

Активации канала PPP и ее фазы

- **Link Dead** – и связь нарушена, либо одна из сторон указала не подключаться.
- **Link Establishment Phase –** проводится настройка Link Control, если настройка была успешной, управление переходит в фазу аутентификации, либо в фазу Network-Layer Protocol, в зависимости от того, требуется ли аутентификация.
- **Authentication Phase –** необязательна, позволяет сторонам проверить друг друга перед установкой соединения, если проверка успешна, управление переходит в фазу Network-Layer Protocol.
- **Network-Layer Protocol Phase –** вызывается NCP для желаемого протокола, так IPCP используется для установки IP сервисов, передача данных по всем успешно установленным протоколам также проходит в этой фазе. Закрытие сетевых протоколов тоже включается в эту фазу.
- **Link Termination Phase –** закрывает соединение, вызывается в случае ошибок аутентификации, если было настолько много ошибок контрольных сумм, что обе стороны решили закрыть соединение, если соединение неожиданно оборвалось, либо если пользователь отключился. Фаза пытается закрыть соединение аккуратно.

Диаграмма фаз PPP



Internet Protocol version 6

- **IPv6** — новая версия IP, призванная решить проблемы, с которыми столкнулся IPv4 при использовании в Интернет за счёт адреса 128 бит.
- Применение IPv6 (сеть /64 на абонента; используется только unicast-адресация) обеспечит более 300 млн IP-адресов на жителя Земли.
- Из IPv6 убраны функции, усложняющие работу маршрутизаторов:
  - Маршрутизаторы не фрагментируют пакет - пакет отбрасывается с ICMP-уведомлением о превышении MTU. Передающая сторона обречена на использование технологии Path MTU discovery. Для лучшей работы протоколов, требовательных к потерям, минимальный MTU поднят до 1280 байт. Фрагментация опционная (вынесена из основного заголовка в дополнительный) и возможна только по инициативе передающей стороны.
  - Из IP-заголовка исключена контрольная сумма - канальные (Ethernet) и транспортные (TCP и UDP) протоколы имеют свои контрольные суммы. Модификация поля *hop limit* (или *TTL* в IPv4) на каждом маршрутизаторе в IPv4 приводила к ее постоянному перерасчету.
- Улучшения IPv6 при увеличенном заголовке (всего вдвое):
  - в сверхскоростных сетях возможна поддержка джамбограмм — до 4 гигабайт;
  - Time to Live переименовано в Hop Limit;
  - появились метки потоков и классы трафика;
  - появилось многоадресное вещание.

Автоконфигурация IPv6

- При инициализации сетевому интерфейсу назначается локальный IPv6-адрес, состоящий из префикса fe80::/10 и идентификатора интерфейса в младшей части адреса - 64-битного идентификатора EUI-64 (MAC-адреса), используемого в пределах сетевого сегмента для обмена ICMPv6 пакетами.
- Для настройки других адресов узел запрашивает информацию о настройках сети у маршрутизаторов ICMPv6-сообщением «Router Solicitation» на их групповой адрес. Маршрутизаторы отвечают ICMPv6-сообщением «Router Advertisement», где содержится информация о сетевом префиксе, адресе шлюза, адресах рекурсивных DNS серверов, MTU,…. Объединяя сетевой префикс и идентификатор интерфейса, узел получает новый адрес. Для защиты идентификатор интерфейса заменяется на псевдослучайное число.
- Для большего административного контроля может быть использован DHCPv6, позволяющий маршрутизатору назначать узлу конкретный адрес.
- Провайдерами используется функция делегирования префиксов клиенту для его простого перехода от провайдера к провайдеру без изменения настроек.

Метки потоков IPv6

- Поле «Метка потока» в IPv6 упрощает процедуру маршрутизации однородного потока пакетов. Поток — это последовательность пакетов, посылаемых отправителем определённому адресату. Все пакеты потока подвергаются определённой обработке, задаваемой дополнительными заголовками.
- Допускается несколько потоков между отправителем и получателем. Метка потока присваивается узлом-отправителем как псевдослучайное 20-битное число. У пакетов потока – одинаковые заголовки, обрабатываемые маршрутизатором.
- При получении первого пакета с меткой потока маршрутизатор анализирует дополнительные заголовки, выполняет предписанные ими функции и запоминает результаты обработки (адрес следующего узла, опции заголовка переходов, перемещение адресов в заголовке маршрутизации…) в кэше. Ключ записи – комбинация адреса источника и метки потока. Следующие пакеты с этим ключом обрабатываются с учётом информации кэша без анализа всех полей заголовка.
- Время жизни записи в кэше ≤ 6 секунд, даже если пакеты потока продолжают поступать. При обнулении записи в кэше и получении следующего пакета потока пакет обрабатывается обычно – для него вновь формируется запись в кэше. Время жизни может быть определено отправителем с помощью протокола управления или опций заголовка переходов и может превышать 6 секунд.
- Обеспечение безопасности в IPv6 осуществляется с использованием IPSec, поддержка которого обязательна для IPv6.

QoS и безопасность IPv6

- Приоритет пакетов маршрутизаторы определяют на основе первых шести бит поля Traffic Class. Первые три бита определяют класс трафика, оставшиеся биты - приоритет удаления. Чем больше значение, тем выше приоритет.
- Разработчики IPv6 рекомендуют использовать для определённых категорий приложений следующие коды класса трафика.

Класс трафика	Назначение
0	Нехарактеризованный трафик
1	Заполняющий трафик (сетевые новости)
2	Несущественный информационный трафик (электронная почта)
3	Резерв
4	Существенный трафик (FTP, HTTP, NFS)
5	Резерв
6	Интерактивный трафик (Telnet, X-terminal, SSH)
7	Управляющий трафик (Маршрутная информация, SNMP)

- В отличие от SSL и TLS, протокол IPSec позволит шифровать любые данные (в том числе UDP) без необходимости какой-либо поддержки со стороны прикладного ПО.

Основы адресации IPv6

- Существуют разные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast).
- Адреса Unicast известны – пакет, посланный на такой адрес, достигает в точности интерфейса, который этому адресу соответствует.
- Адреса Anycast синтаксически неотличимы от Unicast, но адресуют группу интерфейсов – пакет, направленный Anycast, попадёт в ближайший (по метрике маршрутизатора) интерфейс. Anycast используются только маршрутизаторами.
- Адреса Multicast идентифицируют группу интерфейсов – пакет, посланный на Multicast, достигнет всех интерфейсов, привязанных к группе многоадресного вещания.
- Широковещательные адреса IPv4 (обычно xxx.xxx.xxx.255) выражаются адресами многоадресного вещания IPv6.
- Адреса разделяются двоеточиями (напр. fe80:0:0:0:200:f8ff:fe21:67cf). Большое количество нулевых групп может быть пропущено с помощью двойного двоеточия (fe80::200:f8ff:fe21:67cf). Такой пропуск - единственный в адресе.

Типы Unicast адресов IPv6

- Global - соответствуют публичным IPv4 адресам, могут находиться в любом не занятом диапазоне. Региональные интернет-регистраторы распределяют блок адресов 2000::/3 (с 2000:: по 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF).
- Link-Local - соответствуют автоконфигурируемым с помощью протокола APIPA IPv4 адресам. Начинаются с FE80. Используется:
  - в качестве исходного адреса для Router Solicitation (RS) и Router Advertisement (RA) сообщений, для обнаружения маршрутизаторов;
  - для обнаружения соседей (эквивалент InARP для IPv4);
  - как next-hop адрес для маршрутов
- Unique-Local – 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 FC00 и FD00.

Типы Multicast адресов IPv6

Адреса Multicast бывают двух типов:

- Назначенные (*Assigned multicast*) – специальные адреса, назначение которых предопределено, это зарезервированные для определённых групп устройств мультикастовые адреса. Отправляемый на такой адрес пакет будет получен всеми устройствами, входящими в группу.
- Запрошенные (*Solicited multicast*) – остальные адреса, которые устройства могут использовать для прикладных задач. Адрес этого типа автоматически появляется, когда на некотором интерфейсе появляется юникастовый адрес. Адрес формируется из сети FF02:0:0:0:0:1:FF00::/104, оставшиеся 24 бита – такие же как у настроенного юникастового адреса.

Формат пакета IPv6

- Пакеты состоят из управляющей информации для доставки пакета адресату, и полезных данных, которые требуется переслать.
  - Управляющая информация делится на содержащуюся в основном фиксированном заголовке и в необязательных дополнительных заголовках.
  - Полезные данные - это дейтаграмма или фрагмент протокола старшего транспортного уровня, могут быть и данные сетевого уровня (ICMPv6), или же канального уровня (OSPF).
- IPv6-пакеты передаются с помощью протоколов канального уровня, таких как Ethernet, который инкапсулирует каждый пакет в кадр. Но IPv6-пакет может быть передан с помощью туннельного протокола более высокого уровня (6to4 или Teredo).
- Маршрутизаторы не фрагментируют IPv6-пакеты, когда пакет больше MTU подключения, узлы реализуют механизм Path MTU discovery для определения размера MTU пути. Иначе используется минимально допустимый в IPv6-сетях MTU, равный 1280 октетам. Конечные узлы могут фрагментировать пакет перед отправкой, если он больше, чем MTU пути.

Фиксированный заголовок IPv6

- Фиксированный заголовок IPv6-пакета состоит из 40 октетов (320 бит):

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																																

- Описание полей:
  - Version: версия протокола; для IPv6 это значение равно 6 (значение в битах — 0110).
  - Traffic Class: приоритет пакета (8 бит). В поле две части. Старшие 6 бит используются DSCP для классификации пакетов. Младшие два бита используются ECN для контроля перегрузки.
  - Flow Label: метка потока.
  - Payload Length: в отличие от поля Total Length в протоколе IPv4 данное поле не включает фиксированный заголовок пакета (16 бит).
  - Next Header: задаёт тип расширенного заголовка IPv6, который идёт следующим. В последнем расширенном заголовке поле Next Headerзадаёт тип транспортного протокола (TCP, UDP…)
  - Hop Limit: аналог поля time to live в IPv4 (8 бит).
  - Source Address и Destination Address: адрес отправителя и получателя соответственно; по 128 бит.

Расширенные заголовки IPv6

- Расширенные заголовки содержат дополнительную информацию и размещены между фиксированным заголовком и заголовком протокола более высокого уровня. Тип первого расширенного заголовка указывается в поле *Next Header* фиксированного заголовка, а каждый расширенный заголовок имеет аналогичное поле в котором хранится тип следующего расширенного заголовка. В поле *Next Header* последнего заголовка находится тип протокола более высокого уровня, находящегося в качестве полезных данных.
- Расширенный заголовок имеет размер в октетах, кратный 8. Некоторые заголовки необходимо расширить до нужного размера.
- Расширенные заголовки должны быть обработаны только конечным узлом, за исключением заголовка *Hop-By-Hop Options*, который должен быть обработан каждым промежуточным узлом на пути пакета, включая отправителя и получателя. Если расширенных заголовков в пакете несколько, они сортируются как указано в таблице ниже. Все расширенные заголовки - необязательны и не должны появиться в пакете более одного раза, за исключением заголовка *Destination Options (*дважды).
- Если узел не может обработать какой-то расширенный заголовок, то он отбрасывает пакет и отправляет сообщение *Parameter Problem* (ICMPv6 тип 4, код 1). Если в поле *Next Header* расширенного заголовка будет 0, узел должен сделать то же.

Расширенные заголовки IPv6

Расширенный заголовок	Тип	Описание
Hop-by-Hop Options	0	Параметры, которые должны быть обработаны каждым транзитным узлом.
Destination Options	60	Параметры которые должны быть обработаны только получателем.
Routing	43	Позволяет отправителю определять список узлов, которые пакет должен пройти.
Fragment	44	Заголовок содержит информацию по фрагментации пакета.
Authentication Header (AH)	51	Содержит информацию, используемую для аутентификацию большей части пакета. См. IPsec.
Encapsulating Security Payload (ESP)	50	Осуществляет шифрование данных для безопасных подключений. См. IPsec.

Hop-by-hop Options и Destination Options IPv6

- Расширенный заголовок Hop-by-hop Options нужен для передачи дополнительных опций, обрабатываемых каждым узлом на пути пакета, включая отправителя и получателя.
- Расширенный заголовок Destination Options нужен для передачи дополнительных опций конечному узлу или узлам.
- Формат заголовка у обоих расширенных заголовков одинаков.

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	Next Header								Hdr Ext Len								Options															

- Next Header (8 бит) - тип следующего расширенного заголовка или тип протокола, передаваемого в качестве полезных данных.
- Hdr Ext Len (8 бит) - размер заголовка в восьми-октетных блоках, исключая первый блок.
- Options - поле переменной длины, хранящее одну или несколько опций в формате **TLV** (Type-Length-Value).

TLV-кодированные опции IPv6

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	Option Type								Opt Data Len								Option Data															

- Option Type (8 бит) - тип опции. Старшие два бита указывают, что делать, если узел не может распознать опцию:
  - 0 (00) — Пропустить эту опцию и продолжить обработку заголовка.
  - 1 (01) — Отбросить пакет.
  - 2 (10) — Отбросить пакет и отправить сообщение Parameter Problem (ICMPv6 тип 4 код 2) даже если пакет направлен на групповой адрес.
  - 3 (11) — Отбросить пакет и отправить сообщение Parameter Problem (ICMPv6 тип 4 код 2) только если пакет направлен не на групповой адрес.Opt Data Len (8 бит): Длина поля Option Data в октетах.
- Option Data - поле переменной длины, хранящее данные указанного типа.

Routing IPv6

- Расширенный заголовок Routing используется для указания списка транзитных узлов, через которые должен пройти пакет.

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	Next Header								Hdr Ext Len								Routing Type								Segments Left							
4	32	Type-specific Data																															

- Next Header (8 бит) - тип следующего расширенного заголовка или тип протокола, передаваемого в качестве полезных данных.
- Hdr Ext Len (8 бит) - размер заголовка в восьми-октетных блоках, исключая первый блок.
- Routing Type (8 бит) - подтип заголовка.
- Segments Left (8 bits) - количество ещё не посещенных узлов из списка.
- Type-specific Data - поле переменной длины, конкретный формат поля зависит от содержимого поля Routing Type.

Routing IPv6

- Расширенный заголовок Routing используется для указания списка транзитных узлов, через которые должен пройти пакет.

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	Next Header								Hdr Ext Len								Routing Type								Segments Left							
4	32	Type-specific Data																															

  - Next Header (8 бит) - тип следующего расширенного заголовка или тип протокола, передаваемого в качестве полезных данных.
  - Hdr Ext Len (8 бит) - размер заголовка в восьми-октетных блоках, исключая 1-й блок.
  - Routing Type (8 бит) - подтип заголовка.
  - Segments Left (8 bits) - количество ещё не посещенных узлов из списка.
  - Type-specific Data - поле переменной длины, конкретный формат поля зависит от содержимого поля Routing Type.
- Подтип заголовка 0 является устаревшим - заголовок может использоваться для DoS-атаки. Если значение поля Segments Left равно нулю, узел игнорирует расширенный заголовок Routing и приступает к обработке следующих расширенных заголовков. Если значение поля Segments Left не равно нулю, узел должен отбросить пакет и отправить сообщение Parameter Problem (ICMPv6 тип 4, код 0).

Fragment IPv6

- Чтобы отправить пакет, превышающий MTU пути, отправитель разбивает пакет на фрагменты. Расширенный заголовок Fragment содержит информацию для сборки получателем оригинального 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	Next Header								Reserved								Fragment Offset													Res		M
4	32	Identification																															

- Next Header (8 бит): Тип следующего расширенного заголовка или тип протокола, передаваемого в качестве полезных данных.
- Reserved (8 бит): Зарезервировано, должно быть инициализировано нулём.
- Fragment Offset (13 бит): Смещение фрагмента в восьми-октетных блоках относительно начала фрагментируемой части пакета.
- Res (2 бита): Зарезервировано, должно быть инициализировано нулём.
- M (1 бит): Будут ли ещё фрагменты. Если 0, то это последний фрагмент.
- Identification (32 бита): Число, идентифицирующее оригинальный пакет.

Фрагментация IPv6

- IPv6-пакеты не фрагментируются маршрутизаторами. Пакеты больше MTU сетевого подключения уничтожаются и отправителю посылается сообщение Packet too Big (ICMPv6 тип 2), то же происходит в IPv4, если установлен бит Don’t Fragment.
- Конечные IPv6-узлы выполнят Path MTU discovery для определения максимального размера отправляемых пакетов, и протокол более высокого уровня ограничит размер пакета.
- Если протокол более высокого уровня не в состоянии это сделать, отправитель использует расширенный заголовок Fragment для выполнения фрагментации IPv6-пакетов.
- Все протоколы, передающие через себя IPv6-пакеты, должны иметь MTU равный или больший 1280 октетов. Протоколы, не способные передать пакет длиной 1280 октетов одним блоком, должны произвести фрагментацию и сборку самостоятельно, не затрагивая уровень IPv6.

Фрагментирование IPv6

- Пакет, содержащий фрагмент оригинального (большого) пакета, состоит из двух частей:
  - нефрагментируемая часть оригинального пакета, одинаковая для всех фрагментов;
  - фрагментируемая часть, идентифицируемая по смещению фрагмента.
- Нефрагментируемая часть пакета состоит из фиксированного заголовка и расширенных заголовков оригинального пакета (опционально).
- Значение поля Next Header последнего заголовка нефрагментируемой части должно быть равным 44, обозначающее, что следующим заголовком будет Fragment.
- В заголовке Fragment поле Next Header должно быть равно типу первого заголовка фрагментируемой части. После заголовка Fragment следует фрагмент оригинального пакета.
- Размер каждого фрагмента фрагментируемой части должен быть кратен 8, исключение составляет последний фрагмент.

Сборка фрагментов IPv6

- Принимающий узел, собрав все фрагменты, отбрасывает расширенный заголовок Fragments и размещает фрагменты по смещениям, указанным в поле Fragment Offset, умноженным на 8.
- Пакеты, содержащие фрагменты, не обязаны приходить в правильном порядке, они будут переставлены принимающим узлом, если потребуется.
- Если спустя 60 секунд после получения первого фрагмента были собраны не все фрагменты, сборка оригинального пакета отменяется и все полученные фрагменты отбрасываются. Если при этом получен первый фрагмент (с полем Fragmant Offset равным нулю), то отправителю фрагментированного пакета посылается сообщение Fragment Reassembly Time Exceeded (ICMPv6 тип 3 код 1).
- Максимальный размер оригинального пакета не должен превышать 65 535 октетов, а если после сборки оригинальный пакет оказывается больше, он должен быть отброшен.

Полезные данные IPv6

- За фиксированным и расширенными заголовками находится данные протокола транспортного уровня. Поле Next Header последнего IPv6-заголовка указывает тип полезных данных пакета.
- Поле фиксированного заголовка Payload Length имеет размер 16 бит, максимально возможный размер полезных данных и расширенных заголовков равен 65535 октетам. Максимальный размер фрейма многих протоколов канального уровня значительно меньше.
- IPv6-пакет может нести больше данных с помощью опции jumbo payload в расширенном заголовке Hop-By-Hop Options. Опция позволяет обмениваться пакетами с размером полезных данных на 1 байт меньшим чем 4 ГиБ ($2^{32} - 1 = 4294967295$ байт). Пакет с таким содержимым называют джамбограммой.
- TCP и UDP имеют поля длины ≤ 16 битами, для поддержки джамбограмм требуются модифицированные протоколы транспортного уровня. Джамбограммы могут работать на подключениях с MTU, большим чем 65583 октетов (более 65 535 октетов для полезных данных, 40 октетов для фиксированного заголовка и 8 октетов для расширенного заголовка Hop-By-Hop Options).

Адресные нотации IPv6

- Адреса IPv6 отображаются как восемь групп по четыре шестнадцатеричных символа, разделённых двоеточием:
  - 2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d
- Если одна или более групп подряд равны 0000, то они могут быть опущены и заменены на двойное двоеточие (::).
  - 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 две разделённые нулевые группы из-за возникновения неоднозначности.
- При использовании IPv6-адреса в URL необходимо заключать адрес в квадратные скобки:
  - http://[2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d]/
- Если необходимо указать порт, то он пишется после скобок:
  - http://[2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d]:8080/

Уровни иерархии и зарезервированные адреса IPv6

Префикс	Идентификатор провайдера	Идентификатор абонента	Идентификатор подсети	Идентификатор узла

IPv6 адрес	Длина префикса (биты)	Описание	Заметки
::	128	—	см. 0.0.0.0 в IPv4
::1	128	loopback адрес	см. 127.0.0.1 в IPv4
::xx.xx.xx.xx	96	встроенный IPv4	Нижние 32 бита это адрес IPv4. Также называется *IPv4 совместимым IPv6 адресом*. Устарел и больше не используется.
::ffff: xx.xx.xx.xx	96	Адрес IPv4, отображенный на IPv6	Нижние 32 бита это адрес IPv4. Для хостов, не поддерживающих IPv6.
2001:db8::	32	Документирование	Зарезервирован для примеров в документации в RFC 3849
fe80:: — febf::	10	link-local	Аналог 169.254.0.0/16 в IPv4
fec0:: — feff::	10	site-local	Помечен как устаревший в RFC 3879 (Аналог внутренних сетей 10.0.0.0; 172.16.0.0; 192.168.0.0)
fc00::	7	Unique Local Unicast	Пришёл на смену Site-Local RFC 4193
ffxx::	8	multicast	