АДРЕСАЦИЯ
В КОМПЬЮТЕРНЫХ СЕТЯХ

8.0.1.1

Для того, чтобы станции-абоненты могли организовать взаимодействие, им необходимо некоторым образом выделять друг друга среди других станций.

С целью идентификации станций им присваивают некоторые адреса. Таким образом, возникает *адресация* (addressing) в СПД.

8.0.1.2a

Как было сказано ранее, в форматах большинства пакетов присутствуют два адреса:

1. Адрес назначения (destination address).
2. Адрес источника (source address).

В процессе пересылки пакета между абонентами адресация играет ключевое значение.

Производительность СПД напрямую зависит от расположения адресов в пакете. Поэтому адреса «выносят» в самое начало пакета. Более того, поскольку с точки зрения доставки пакета адрес назначения является более важным (в СПД анализируется именно этот адрес), он как правило располагается раньше.

8.0.1.2b

Многие топологии предполагают возможность приема переданного одной из станций пакета всеми остальными станциями в пределах сегмента -- вне зависимости от того, какой из станций пакет был предназначен. Следует различать действия «принят станцией», «проанализирован станцией» и «обработан станцией». Факт приема станцией пакета подразумевает, что пакет будет проанализирован, но не подразумевает «полноценную» обработку. Именно сравнение считанного из принятого пакета адреса назначения со своим адресом, позволяет станции распознать пакет как «свой».

Считанный из пакета адрес источника позволяет станции (при необходимости) определить абонента, создавшего пакет.

8.0.2.1

Следует учитывать, что важное влияние на адресацию оказывает инкапсуляция. Адресация всегда «привязана» к некоторому протоколу, а протокол, в свою очередь, «привязан» к уровню модели OSI. Поэтому закономерно, что на каждом из уровней присутствует своя независимая система адресации.

Пакет, воспринятый как «свой» на одном из уровней, после его передачи на более высокий уровень, там вполне может быть «отвергнут». Кроме того, «окончательная» обработка не всегда происходит на прикладном уровне (классический пример: ретрансляция пакета между сегментами при маршрутизации).

8.0.2.2

В каждом пакете должны присутствовать по крайней мере адреса канального уровня.

В большинстве же практических реализаций семейств протоколов, кроме адресации на канальном уровне, предусмотрена адресация на сетевом (в связке с транспортным) и прикладном уровнях.

Допустимость повторения адресов на одном уровне вытекает из цели разработки определенного протокола.

8.0.2.3

Адреса канального уровня «зашиваются» в сетевое оборудование при его производстве и поэтому повторяться не должны. Они не предполагают возможность пользовательского вмешательства и их считают абсолютно уникальными. Часто (в том числе Cisco) такую адресацию называют *физической* (physical).

Адреса сетевого и прикладного уровней назначают пользователи. Часто (в том числе Cisco) такую адресацию называют *логической* (logical).

8.0.2.4

В нормальной ситуации, по крайней мере в течение сеанса взаимодействия, адреса разных уровней одной станции должны соответствовать друг другу. Поэтому возникает необходимость в служебных протоколах, отыскивающих эти соответствия.

8.0.2.5

Кроме всего прочего, даже на одном уровне модели OSI адресация может быть *иерархической* (hierarchical), то есть предполагать определенную структуризацию соответствующего адресного пространства.

Иерархичность выражается в количественном и качественном разделении адресов на типы.

8.0.2.6

Одним из примеров может служить связка адреса сетевого уровня с адресом транспортного уровня.

В рамках функционирования сетевой ОС можно выделить объекты:

1. *Сетевой процесс* (network process) -- представляет собой пару: процессор и выполняющаяся на нем сетевая (то есть использующая сетевые ресурсы) программа; причем, если меняется хотя бы один из этих компонентов, то получается новый процесс.

2. *Сетевой ресурс* (network resource) -- это любой компонент вычислительной системы, который может быть предоставлен в пользование сетевому процессу на определенное время.

Для того, чтобы взаимодействующие сетевые процессы могли найти друг друга, во всех реальных системах используется три уровня адресации:

1. Необходимо адресовать подсеть -- используется *адрес подсети* (subnet address).

2. Необходимо адресовать станцию в подсети -- используется *адрес станции* (node address).

3. Необходимо адресовать процесс в станции -- используется так называемый *адрес программного порта* (software port).

Под адрес порта, как правило, отведено два байта.

При назначении программных портов учитываются диапазоны, к которым они относятся.

Диапазоны программных портов применительно к семейству TCP/IP.

Port Number Range	Port Group
0 -- 1023	Well Known
1024 -- 49151	Registered
49152 -- 65535	Private and Dynamic

Так называемые хорошо известные порты предназначены для адресации основных сервисов в Internet.

Порты для дополнительных публичных сервисов нужно регистрировать.

Порты для приватных (и редких) сервисов регистрировать не нужно.

8.0.2.8

Для чего нужны динамические порты?

8.0.3.1

Специально для компьютерных сетей были разработаны четыре основных способа адресации, которые заключаются в применении адресов четырех базовых типов:

1. *Юникаст* (unicast) -- пакет с таковым адресом назначения должен быть обработан одной соответствующей станцией.

2. *Бродкаст* или, по-другому, *широковещательных* (brodcast) -- пакет с таковым адресом назначения должен быть обработан всеми станциями.

3. *Мультикаст* (multicast) -- пакет с таковым адресом назначения должен быть обработан несколькими станциями из множества.

4. *Эникаст* (anycast) -- пакет с таковым адресом назначения должен быть обработан одной станцией из множества.

По сути, мультикаст- и эникаст-адреса являются *групповыми идентификаторами* (Group IDs).

Специфика тех или иных типов накладывает ограничения на возможность использования адресов.

Бродкаст-, мультикаст- и эникаст-адреса не могут быть адресами источников, так как отдельно взятый пакет может сгенерировать только одна станция.

8.0.3.3

Особую проблему представляет собой межсегментная ретрансляция группового трафика (актуально для прикладного мультикаст-трафика). Проблема решается с помощью дополнительных служебных протоколов.

8.0.3.4

Наиболее сложной формой адресации является эникаст-адресация.

Очевидно, что каждый раз при приеме эникаст-пакета должен осуществляться выбор на основе какого-либо критерия. При этом адресуемые станции должны осуществлять выбор в пределах группы сами. Отправившая пакет станция не может принимать участие в алгоритме выбора, она уже сделала свой «выбор» записав в пакет в качестве адреса назначения эникаст-адрес. Выбор должен быть сделан заблаговременно, чтобы принимающая станция была готова к поступлению в группу пакета.

Примером критерия выбора может служить время задержки.

Выбор может осуществляться однократно либо периодически.

8.0.3.5

Придумайте алгоритм децентрализованного выбора в группе, если станции заранее ничего друг о друге не знают.

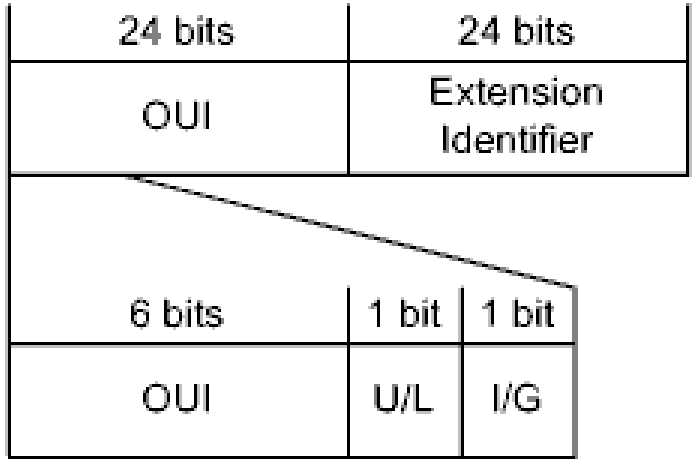
8.0.4.1

Почти все широко применяемые в настоящее время технологии ЛКС (например, Ethernet) разработаны IEEE, поэтому производители соответствующего сетевого оборудования соблюдают правила, сформулированные в стандартах этой организации.

Уникальность MAC-адресов контролирует IEEE RA (IEEE Registration Authority).

В стандартах IEEE определены три базовых формата MAC-адресов: MAC-48, EUI-48 и EUI-64, где EUI (Extended Unique Identifier) -- расширенный уникальный идентификатор.

MAC-48 можно считать синонимом EUI-48, хотя изначально это было более общее понятие.



00-16-41-57-7D-48

Поля:

OUI (Organizationally Unique Identifier) -- уникальный идентификатор организации.

U/L (Universal/Local) -- признак универсальности-локальности адреса.

I/G (Individual/Group) -- признак индивидуального-группового адреса.

Extension Identifier -- идентификатор-наполнитель.

Формат EUI-48

8.0.4.2b

OUIs выдают централизовано, уникальность оставшейся части должны обеспечивать сами организации (любым способом по своему усмотрению).

Время валидности адресов (время, которое нужно выдержать перед повторным присвоением того же адреса другому устройству) определено как 100 лет.

Иногда, при администрировании, возникает необходимость подменить адрес, «зашитый» в оборудование, на некий другой. Этот новый адрес называют *локальным административным адресом* (locally administered address). Его признаком является единичное значение бита U/L. Согласовывать значение остальных битов не требуется, но в пределах сегмента адрес не должен повторяться.

8.0.4.2c

Граница между OUI и Extension Identifier может проходить не только посредине адреса. В общем случае предусмотрены три варианта разрядности поля OUI:

1. MA-L (MAC Address -- Large) -- 24 бита (данная схема использовалась IEEE RA до 1 января 2014 г.)
2. MA-M (MAC Address -- Medium) -- 28 битов (схема доступна после 1 января 2014 г.)
3. MA-S (MAC Address -- Small) -- 36 битов (схема доступна после 1 января 2014 г.)

Иногда поле OUI рассматривают как CID (Company ID), что, по большому счету, то же самое -- зависит от комбинации значений битов U/L и I/G (рассматривают уже как биты X и M соответственно).

При так называемом каноническом представлении (canonical representation) MAC-адрес сдвигается в канал начиная со старших разрядов (на рисунке слева).

По правилам IEEE MAC-адреса записывают в следующей нотации:

XX-XX-XX-XX-XX-XX

Где X -- шестнадцатеричная цифра (верхний регистр).

Но очень часто используют альтернативные нотации.
Примеры:

`00-16-41-57-7D-48` -- IEEE
`00-16-41-57-7d-48`
`00:16:41:57:7D:48`
`00:16:41:57:7d:48`
`0016.4157.7d48` -- Cisco

8.0.4.4

Все юникаст-MAC-адреса должны иметь нулевое значение бита I/G.

Групповые MAC-адреса формируются по особым правилам, которые будут рассмотрены позже.

В качестве бродкаст-MAC-адреса принято использовать значение `FF-FF-FF-FF-FF-FF`.

8.0.4.5

Следует отметить, что EUI-64 может использоваться не только для адресации, а и для просто идентификации устройств.

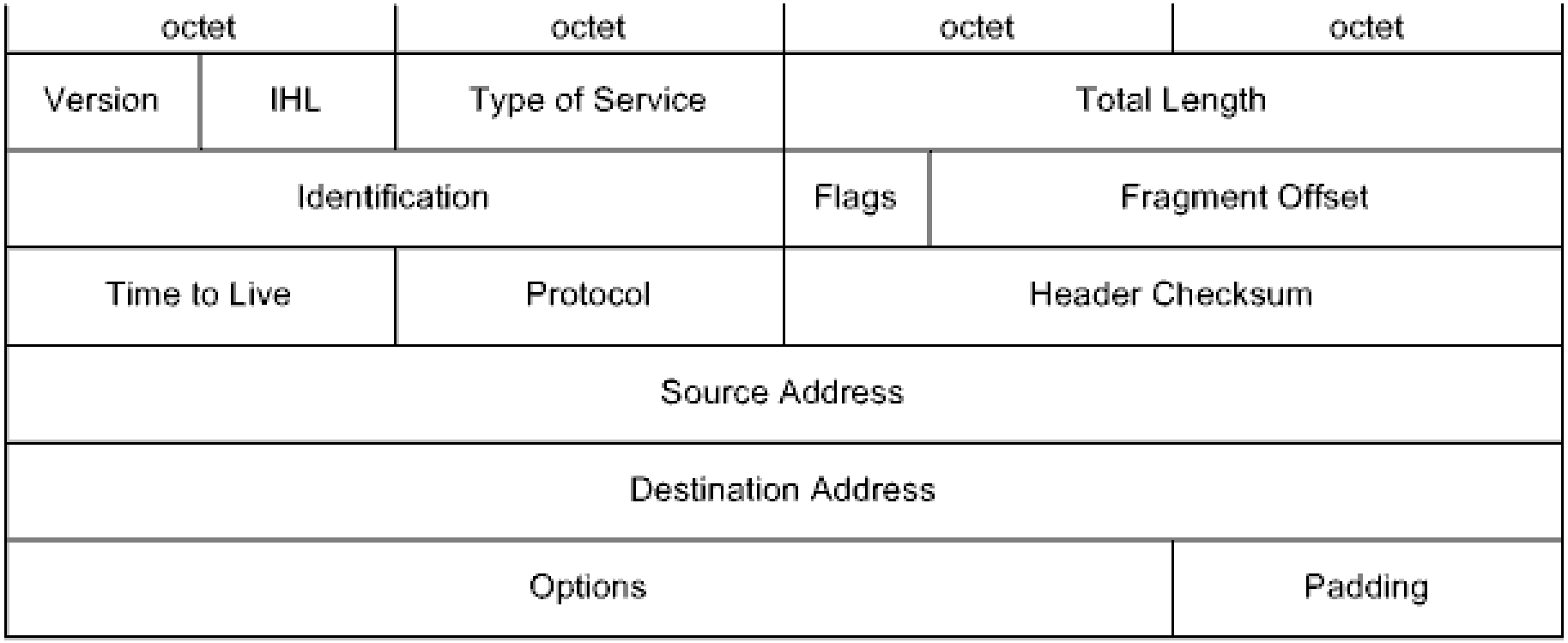
Примеры технологий с применением EUI-48: Ethernet, Wi-Fi, Token Ring.

Примеры технологий с применением EUI-64: IPv6, FireWire.

8.0.5.1a

В семействе TCP/IP за адресацию на сетевом уровне отвечает протокол IP.

Заголовок протокола IPv4 (версии 4) имеет фиксированную структуру.



Формат заголовка IPv4

8.0.5.1b

Поля:

Version -- версия (значение равно 4).

IHL (Internet Header Length) -- длина заголовка (в 32-ухбитных словах, минимальное значение равно 5).

Type of Service -- тип сервиса (связано с QoS).

Total Length -- общая длина данных (в байтах, не может превышать 65535 байтов).

Flags -- флаги.

Fragment Offset -- смещение текущего фрагмента (в 64-ехбитных словах, смещение первого фрагмента равно нулю).

Time to Live -- «время жизни» (при каждой ретрансляции уменьшается, когда становится равным нулю пакет уничтожается).

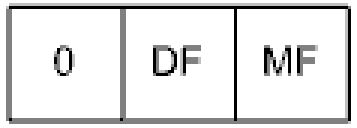
Protocol -- протокол (инкапсулируемый в поле данных).

Header Checksum -- контрольная сумма заголовка.

Source Address -- адрес источника.

Destination Address -- адрес назначения.

Options -- опции (например, связанные с безопасностью, размер вариативен).



Флаги:

DF (Don’t Fragment): 0 -- пакет фрагментирован, 1 -- пакет нефрагментирован.

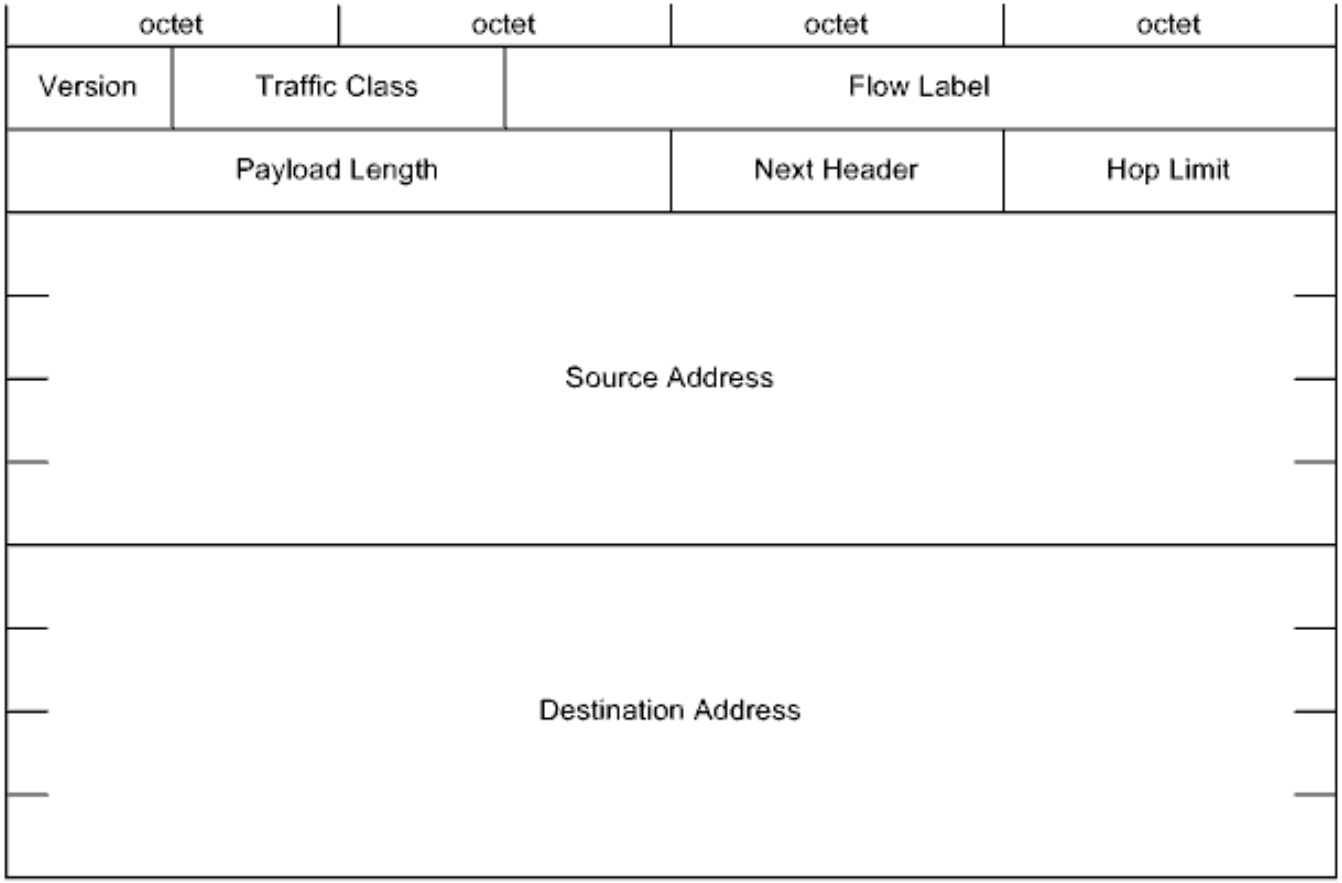
MF (More Fragments): 0 -- текущий фрагмент является последним, 1 -- текущий фрагмент не является последним.

Поле Flags

8.0.5.3a

Заголовок протокола IPv6 имеет «гибкую» структуру.

Заголовки «каскадируются» -- сколько заголовков нужно, столько и вставляется.



Формат заголовка IPv6

8.0.5.3b

Новые поля:

Traffic Class -- класс трафика (связано с QoS).

Flow Label -- метка потока (связано с QoS).

Payload Length -- длина полезной нагрузки (в байтах, аналог поля Total Length).

Next Header -- селектор следующего заголовка (в том числе, аналог поля Protocol).

Hop Limit -- ограничитель числа «прыжков» (аналог поля Time to Live).

8.0.5.4

Полноценная реализация IPv6 должна поддерживать следующие заголовки:

1. IPv6 header -- собственно IPv6-заголовок.
2. Hop-by-Hop Options header -- заголовок опций ретрансляции.
3. Destination Options header -- заголовок предназначенных станции назначения опций.
4. Routing header -- маршрутизационный заголовок.
5. Fragment header -- заголовок франмента.
6. Authentication header -- заголовок протокола AH (связано с защитой информации).
7. Encapsulating Security Payload header -- заголовок протокола ESP (связано с защитой информации).

+8. Upper-layer header -- заголовок протокола вышестоящего уровня.

Подробно IPv4- и IPv6-адресация будет рассмотрена в дальнейшем.

8.0.5.5

Охарактеризуйте протокол IP, исходя из уже полученных знаний.

8.0.6.1

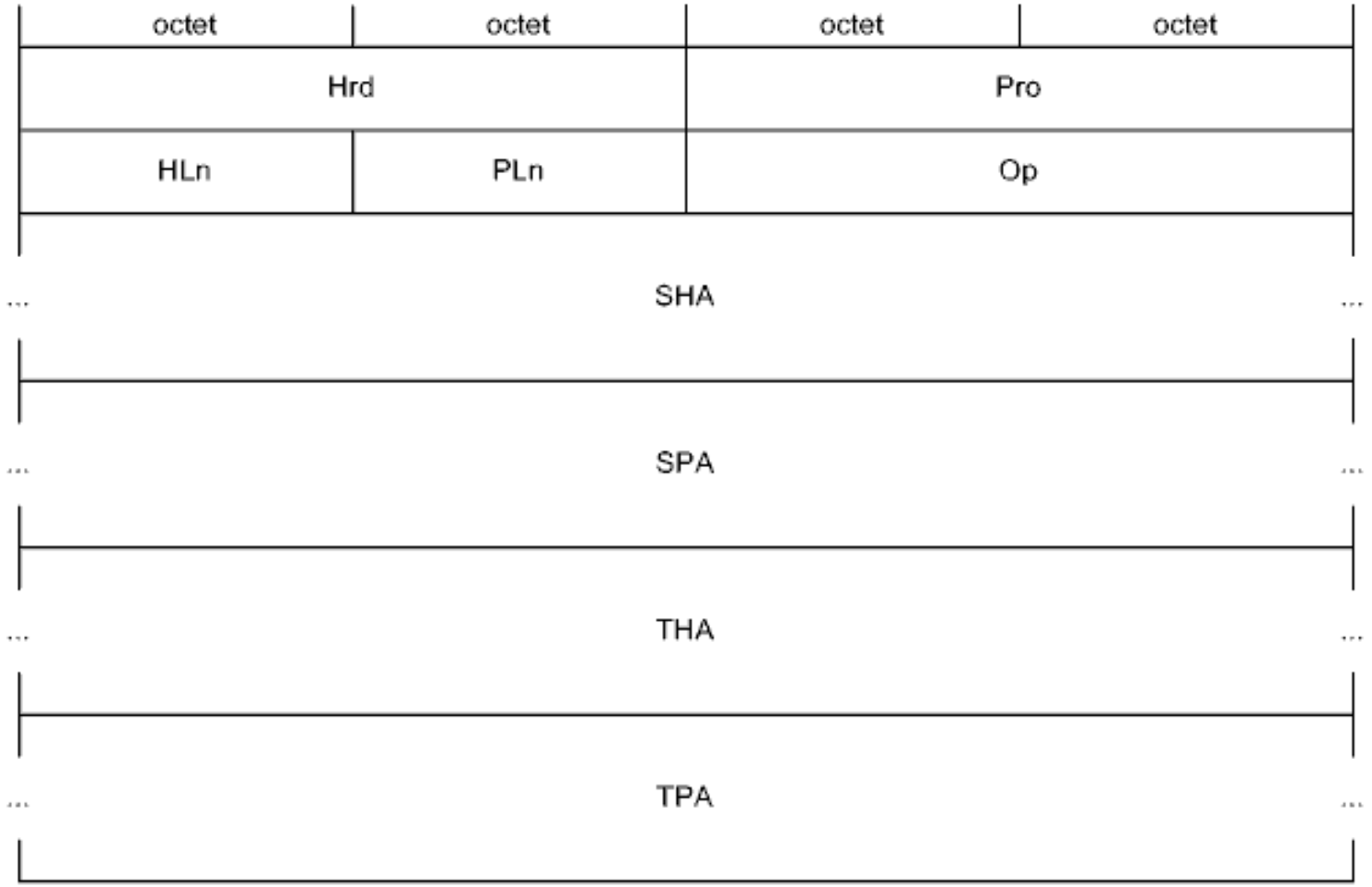
Группа протоколов под названием ARPs (Address Resolution Protocols) предназначена для восстановления соответствий между MAC-адресами и IP-адресами.

Под прямым преобразованием, собственно ARP, понимают нахождение MAC-адреса по IP-адресу.

Обратное преобразование выполняется по протоколу RARP (Reverse ARP).

Существует еще InARP (Inverse ARP) и некоторые другие расширения, которые, как и практическое применение ARP, будут рассмотрены в дальнейшем.

8.0.6.2a



Формат пакета ARP

8.0.6.2b

Поля:

Hrd (Hardware) -- тип оборудования (1 -- Ethernet).

Pro (Protocol) -- протокол (800h -- IP).

HLn (Hardware address Length) -- длина аппаратного (физического) адреса (в байтах, 6 -- Ethernet).

PLn (Protocol address Length) -- длина протокольного (логического) адреса (в байтах, 4 -- IP).

5. Op (Opcode) -- код операции: 1 -- Request -- запрос, 2 -- Reply -- ответ (и некоторые другие).

6. SHA (Sender Hardware Address) -- аппаратный адрес запрашивающей станции.

7. SPA (Sender Protocol Address) -- протокольный адрес запрашивающей станции.

8. THA (Target Hardware Address) -- аппаратный адрес запрашиваемой станции.

9. TPA (Target Protocol Address) -- протокольный адрес запрашиваемой станции.

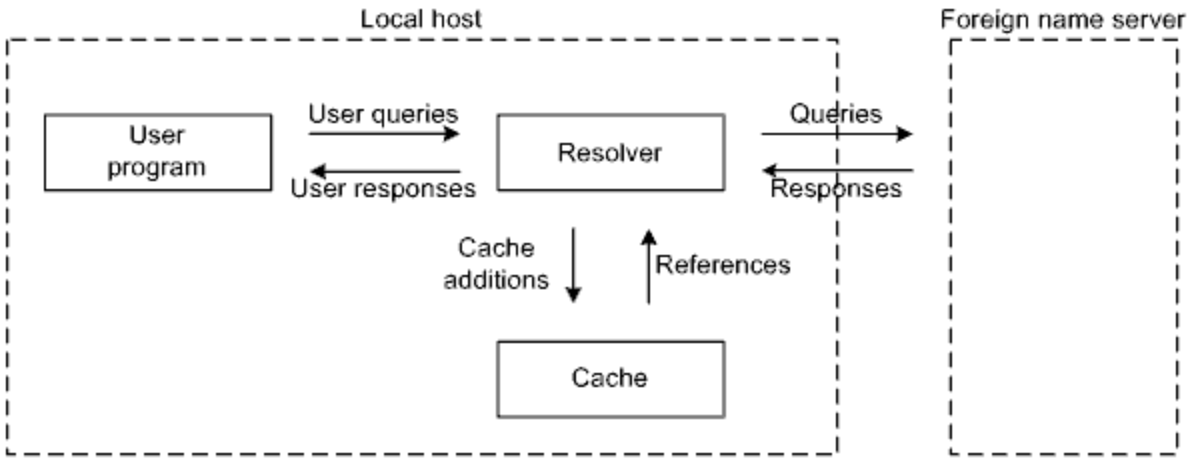
8.0.7.1

Протокол системы DNS (Domain Name System) (основные RFCs -- 1034 и 1035) предназначен для восстановления соответствий между IP-адресами и адресами прикладного уровня.

8.0.7.2

Следует отметить, что под *доменом* (domain, иногда cloud) в СПД обобщенно понимают совокупность устройств, работающих в рамках некоторых единых правил.

8.0.7.3



Структура системы DNS

8.0.7.4

Система DNS соответствует клиент-серверной модели и включает три основных компонента:

1. Адресное пространство доменных названий (domain name space) и записи о ресурсах -- RRs (Resource Records).
2. Серверы названий (name servers).
3. Программы, отвечающие на запросы клиентов (resolvers).

Каждый из этих компонентов «видит» систему DNS по-своему.

8.0.7.5a

Адресное пространство доменных названий имеет иерархическую древовидную структуру.

Каждый узел дерева на некотором уровне иерархии обозначают *DNS-меткой* (DNS label) длиной от 0 до 63 байтов (должна начинаться с буквы и состоять из комбинации букв любого регистра, цифр и символа –). Метка нулевой длины зарезервирована и является корнем дерева. При присоединении станции к определенному домену ей так же присваивают метку.

Доменное название строится из меток -- в соответствии с путем к корневой метке. Полная длина не может превышать 255 байтов.

Доменное название может относиться как к отдельно взятой станции, так и к некоторой ветви дерева, то есть к DNS-домену (DNS domain).

Доменное название может быть как абсолютным (absolute), то есть содержащим всю цепочку меток от станции до корневой метки, так и относительным (relative), то есть содержащим только часть меток.

Внутреннее представление метки: один байт, в котором указана длина метки, за которым следуют собственно байты метки. При интерпретации меток регистр букв не учитывается.

8.0.7.5b

Согласно принятой нотации записи доменных названий метки разделяют точками и корневая метка является крайней справа.

8.0.7.6

Напишите пример цепи из доменных названий.

Изначально, когда сеть Internet была сосредоточена на территории США, базовым критерием структуризации доменных названий Internet-сайтов являлось целевое использование. Были зарегистрированы следующие домены первого уровня -- TLDs (Top Level Domains): `.arpa` (ARPANET), `.com` (commerce), `.edu` (education), `.gov` (government), `.int` (international), `.mil` (military), `.net` (network), и `.org` (organization).

В дальнейшем, по мере расширения Internet, широкое распространение получили национальные TLDs, например, `.BY`.

С недавнего времени основной упор сделан на продвижение национальных языков (в качестве альтернативы английскому языку). Зарегистрированы дополнительные национальные TLDs, например, `.БЕЛ`.

Четыре TLDs зарезервированы для специального использования: `.example`, `.invalid`, `.localhost`, `.test`.

8.0.7.8

Серверы названий удерживают БД с записями о ресурсах.

Серверы названий делят на:

1. *Авторитетные* (authoritative, master) -- являются первоисточниками информации о некоторых частях системы DNS, называемых *зонами* (zones).

2. *Вспомогательные* (non-authoritative, slave) -- работающие на основании сведений от авторитетных серверов.

Таким образом, серверы так же образуют иерархию -- вплоть до наличия корневых серверов.

8.0.7.9

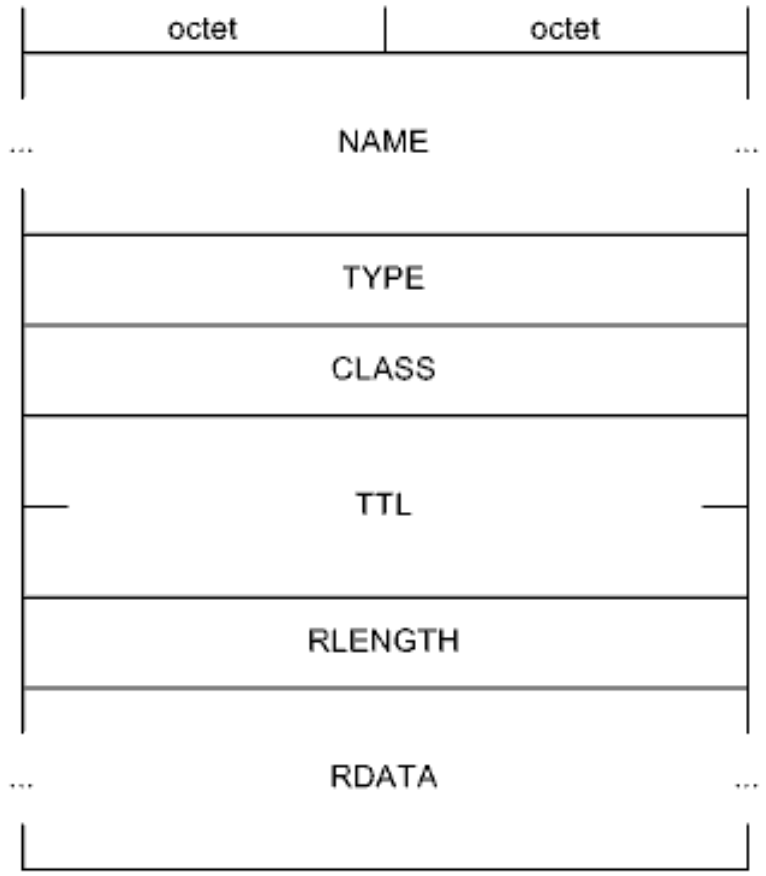
Под прямым преобразованием понимают нахождение IP-адреса по доменному названию.

Возможно и обратное преобразование.

8.0.7.10

Каждой входящей в систему DNS станции (как и каждому домену) соответствует некоторое количество RRs.

8.0.7.11a



Формат DNS RR

8.0.7.11b

Поля:

1. NAME -- доменное название (к которому относится RR, целевое при поиске).
2. TYPE -- тип.
3. CLASS -- класс (семейство протоколов).
4. TTL (Time To Live) -- «время жизни» (то есть время валидности RR, в секундах).
5. RLENGTH (Resource LENGTH) -- длина данных ресурса.
6. RDATA (Resource DATA) -- данные ресурса (зависят от типа и класса).

8.0.7.12a

Основные типы RRs:

1. A (A host address) -- IP-адрес хоста.

2. NS (Name Server) -- авторитетный сервер названий домена.

5. CNAME (Canonical NAME) -- каноническое доменное название (станции либо домена, для псевдонима).

6. SOA (Start Of a zone of Authority) -- оригинальные параметры зоны (сервер с изначальным описанием зоны, контактное лицо, время валидности и другие).

10. NULL -- нулевая запись (произвольная информация).

12. PTR (PoinTeR) -- указатель -- доменное название станции (при обратных преобразованиях).

13. HINFO (Host INFO) -- информация о станции (процессор и ОС).

15. MX (Mail eXchange) -- доменное название почтового сервера в домене (включая приоритет, этот тип используется и вместо нескольких отмененных типов).

16. TXT (TeXT strings) -- текстовые строки (либо строка).

28. AAAA (--) -- IPv6-адрес хоста (RFC 3596).

33. SRV (SeRVer selection) -- описание сервиса (любого дополнительного сетевого сервиса на станции, например, файлового) (RFC 2782).

И некоторые другие.

Классы RRs:

1. IN -- Internet.
2. CS -- CSNET (устарел и аннулирован).
3. CH -- Chaosnet (устарел).
4. HS -- Hesiod (для БД, очень редкий).

Остальные значения классов зарезервированы.

Примеры значений RRs класса IN:

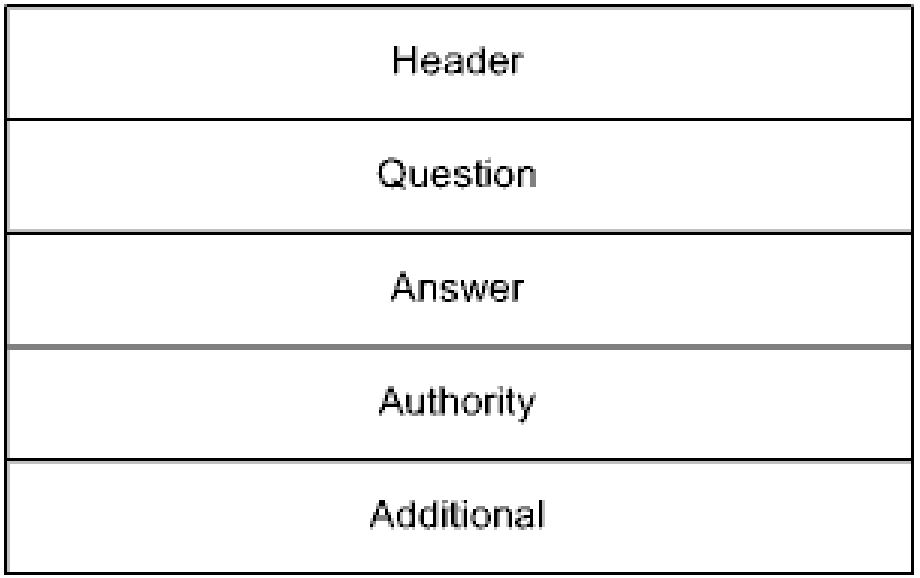
A: `192.168.11.1.`

CNAME: `5-508-fileserv.bsuir.by.`

MX: `10 mail.bsuir.by.`

NS: `proxy1.bsuir.by.`

PTR: `5-508-fileserv.bsuir.by.`

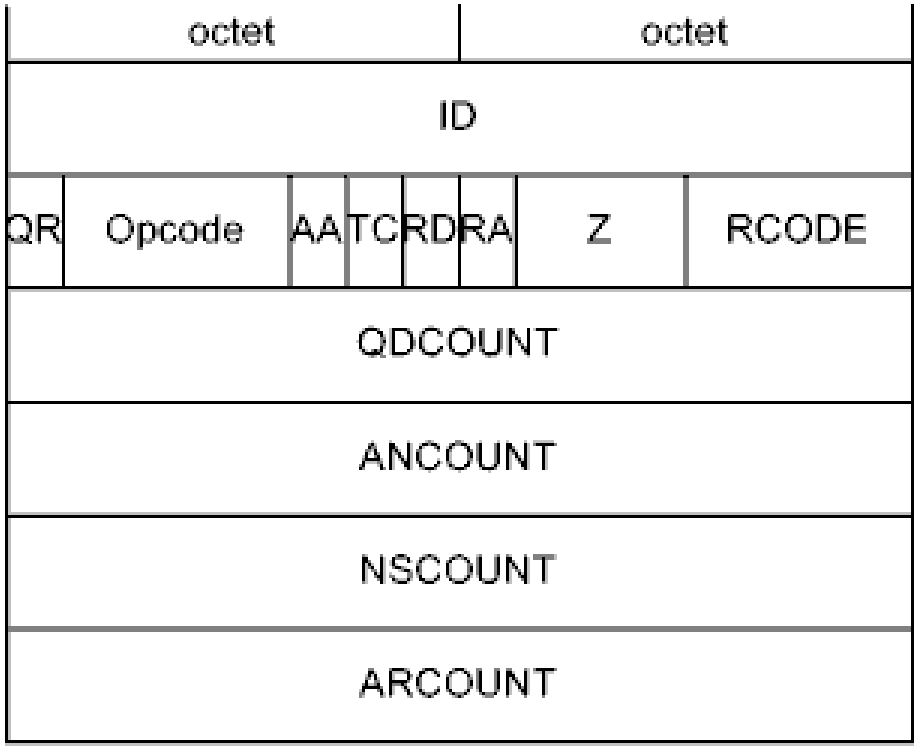


Поля:
1. Header -- заголовок.
2. Question -- вопрос.
3. Answer -- ответ.
4. Authority – авторитетный ответ.
5. Additional -- дополнение.

Заголовок присутствует всегда, остальные поля вариативны.

Формат сообщения DNS

8.0.7.16a



Формат заголовка сообщения DNS

8.0.7.16b

Поля:

1. ID (IDentifier) -- идентификатор (программы, сгенерировавшей запрос).

2. QR (Query/Responce) -- флаг запроса-ответа: 0 -- Query -- запрос, 1 -- Responce -- ответ.

3. OPCODE (OPeration CODE) -- код операции (запроса): 0 -- QUERY (standard QUERY) -- стандартный запрос (о прямом преобразовании), 1 -- IQUERY (Inverse QUERY) -- запрос об обратном преобразовании (RFC 3425 отменен, альтернатива -- использование PTR RR), 2 -- STATUS (server STATUS request) -- запрос состояния сервера, 4 -- NOTIFY -- уведомление (об изменениях в БД о зоне) (RFC 1996), 5 -- UPDATE -- обновление (динамическое обновление БД о зоне) (RFC 2136), 6 -- DSO (DNS Stateful Operations) -- стабильные DNS-операции (альтернативный унифицированный синтаксис) (RFC 8490), остальные значения зарезервированы.

4. AA (Authoritative Answer) -- флаг авторитетного ответа.

5. TC (TrunCation) -- флаг «усеченности» сообщения (при слишком длинном сообщении).

6. RD (Recursion Desired) -- флаг желательной рекурсии (при обработке запроса).

8.0.7.16c

7. RA (Recursion Available) -- флаг поддержки рекурсии.

8. Z (Zero) -- нулевые биты (зарезервировано).

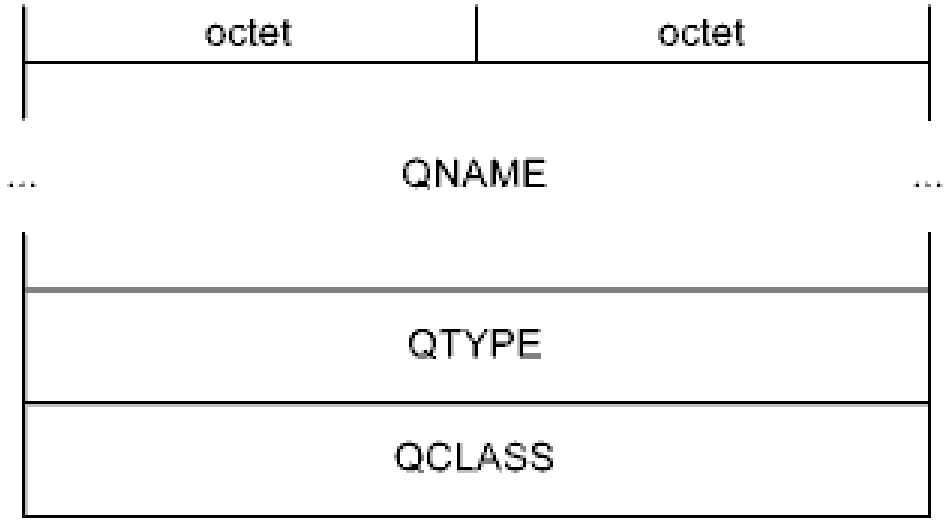
9. RCODE (Response CODE) -- код ответа: 0 -- NoError (No Error) -- ошибок нет, 1 -- FormErr (Format Error) -- ошибка в формате, 2 -- ServFail (Server Failure) -- сбой сервера, 3 -- NXDomain (Non-eXistent Domain Name) -- доменное название не существует, 4 -- NotImp (Not Implemented) -- запрос не поддерживается, 5 -- Refused (Query Refused) -- запрос отклонен, остальные значения относятся к расширениям DNS (RFC 2136, RFC 2845, RFC 2930, RFC 4635, RFC 6672, RFC 6891, RFC 7873, RFC 8490) и зарезервированы.

10. QDCOUNT (Query DNS COUNT) -- количество элементов (RRs) в поле Question (обычно один).

11. ANCOUNT (ANswer COUNT) -- количество элементов (RRs) в поле Answer.

12. NSCOUNT (Name Server COUNT) -- количество элементов (RRs) в поле Authority.

13. ARCOUNT (Additional Records COUNT) -- количество элементов (RRs) в поле Additional.



Поля:

1. QNAME (Query NAME) -- доменное название в запросе.
2. QTYPE -- (Query TYPE) -- тип запроса.
3. QCLASS (Query CLASS) -- класс запроса.

Формат элемента поля Question

Множество значений QTYPE является расширением множества значений TYPE. Основные из новых типов:

251. IXFR (Incremental zone i.e. X transFeR) -- запрос текущих изменений в БД о зоне (от вспомогательного сервера авторитетному, по одноименному протоколу) (RFC 1995).

252. AXFR (Authoritative zone i.e. X transFeR) -- запрос полной БД о зоне (по одноименному протоколу) (+RFC 5936).

255. * -- запрос всех RRs.

Множество значений QCLASS является расширением множества значений CLASS. Новый класс:

255. * -- любой класс.

Практическое применение DNS будет рассмотрено в дальнейшем.