На прошлом занятии мы рассматривали принципы работы протокола DHCP и в некоторой степени затрагивали вопросы, связанные с полями заголовка DHCP. Сегодняшнее занятие как раз и будет посвящено изучению заголовка протокола DHCP. Теперь, после того, как Вы уже представляете себе основные задачи, решаемые с помощью DHCP и основные принципы его работы, мы можем рассмотреть формат заголовка протокола и уточнить все те детали взаимодействия с помощью DHCP, которые пока не затрагивали в прошлом уроке.

Итак, будем постепенно рассматривать заголовок протокола DHCP, поясняя каждое поле заголовка и задачи, которые с помощью этого поля решаются. Для начала отметим, что заголовок DHCP, как ряд заголовков ранее изученных нами протоколов содержит в себе стационарную часть с фиксированным форматом и опции, которые могут следовать после стационарной части заголовка. При этом стационарная часть заголовка DHCP составляет целых 59 (!) четырехбайтовых слов или 236 байт! Зачем DHCP такой большой заголовок станет ясно по ходу его изучения. Итак, начинаем рассматривать первое четырехбайтовое слово DHCP заголовка:

| op | | | |
|---|---|---|---|

Данное поле выполняет вполне понятную и привычную нам роль, аналог такого поля есть во многих сетевых протоколах, например в ARP. Поле показывает тип операции, значение 01 показывает, что данный пакет является запросом от клиента серверу (BOOTREQEST), значение 02 показывает, что данный пакет является ответом сервера (BOOTREPLY). Очень важно подчеркнуть, что данное поле принимает только ДВА значения и не предназначено для идентификации типа пакета (DHCPDISCOVER, DHCPOFFER, DHCPREQEST и т.д.) Для идентификации типа DHCP пакета применяется другой механизм, мы разберем его в свое время, для нас сейчас важно понимать, что сообщения следующих типов: DHCPDISCOVER, DHCPREQEST, DHCPDECLINE, DHCPRELEASE имеют значение данного поля, равное 01, а сообщения DHCPOFFER, DHCPACK, DHCPNACK – 02.

Переходим к рассмотрению второго поля заголовка:

| op | htype | | |
|---|---|---|---|

Поле htype или Hardware Type показывает, какая технология канального уровня используется сервером и клиентом и полностью аналогично соответствующему полю заголовка ARP. Особые комментарии очевидно не нужны, переходим к следующему полю:

| op | htype | hlen | |
|---|---|---|---|

Данное поле показывает длину аппаратного адреса, используемого DHCP клиентом. Как мы уже знаем, аппаратный адрес клиента является важным идентификатором для DHCP сервера, на основании которого принимается решение о предложении клиенту того или иного IP адреса, так что вполне очевидно, что MAC адрес клиента будет присутствовать в заголовке DHCP пакета. А так как формально говоря длина аппаратного адреса в разных канальных технологиях может быть отличной, то DHCP серверу придется считывать поле MAC адреса станции клиента переменной длины, и один из самых простых способов обеспечить нормальное чтение такого поля – просто заранее оговорить его длину. В целом снова налицо полная аналогия с протоколом ARP. Следующее поле называется hops:

| op | htype | hlen | hops |
|---|---|---|---|

Данное поле предназначено для использования в особых случаях – при маршрутизации DHCP сообщений. Мы пока не говорили о технологии маршрутизации сообщений протокола DHCP, более того, так как большинство пакетов DHCP отправляются широковещательно нам пока даже не понятно, каким образом такие пакеты можно вообще маршрутизировать. Позднее мы рассмотрим технологию маршрутизации сообщений DHCP, пока же скажем, что это поле клиент

всегда обязан установить в 0, а маршрутизаторы, перенаправляющие DHCP сообщения могут это поле использовать, мы к этому вопросу еще вернемся, пока мы рассматриваем работу протокола DHCP в одной сети и это поле нас не интересует. Следующее четырехбайтовое слово целиком состоит из единственного поля xid:

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |

Данное поле играет важную роль в DHCP взаимодействиях и служит для идентификации транзакции между клиентов и сервером. Напомним, что пакеты протокола DHCP очень часто отправляются широковещательно, и для того, чтобы клиенты и сервера могли отличать пакеты посланные им от чужих пакетов используется данное поле «идентификатор транзакции». Работает данное поле очень просто: клиент в своем первом пакете серверу устанавливает случайное число в это поле, сервер во всех своих пакетах данному клиенту в рамках данной процедуры взаимодействия будет цитировать значение этого поля, а клиент соответственно в рамках данного обмена пакетами с сервером все свои пакеты будет снабжать тем же значением поля xid. Таким образом участники DHCP транзакции могут четко отличать пакеты, поступившие в рамках данной транзакции от пакетов, принадлежащих другим транзакциям. Без подобного поля сервера и клиенты легко могли бы запутаться при осуществлении нескольких взаимодействий одновременно.

Следующее четырехбайтовое слово состоит из двух двухбайтовых полей, рассматриваем первое – secs:

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | | |

Данное поле заполняет клиент. Это поле имеет смысл только в пакетах типа DHCPDISCOVER ему клиенту рекомендовано указывать в этом поле, сколько секунд прошло от момента, когда клиент начал пытаться получить IP адрес до момента, когда был послан данный пакет DHCPDISCOVER. Т.е., например, клиент посылает первый пакет DHCPDISCOVER и заполняет поле secs = 0. Не получив подходящего (или вообще никакого) предложения, клиент посылает еще один пакет DHCPDISCOVER через, положим, 4 секунды, и заполняет поле secs = 4. Для чего это может быть полезно? Положим в нашей сети несколько DHCP серверов. Если они все вместе будут отвечать на каждый DHCPDISCOVER клиента, то будет порождаться лишний трафик, да еще и широковещательный. Вместо этого можно «назначить» один из DHCP серверов в сети «главным», т.е. позволить ему отвечать на DHCP пакеты с полем secs = 0, а прочие сервера сконфигурировать таким образом, чтобы они отвечали только на пакеты с некоторым значением поля secs, не менее заданного. Тогда на DHCPDISCOVER поступающие от клиентов будет отвечать только один DHCP сервер, а в случае выхода его из строя, когда значение поля secs в пакетах DHCPDISCOVER, посылаемых клиентами превысит заранее сконфигурированный порог, на такие пакеты смогут отвечать другие сервера, таким образом можно уменьшить широковещательный трафик в сети. Отмечаем, что мы пока еще серьезно не говорили о проектировании среды с многими DHCP серверами, поэтому детальное рассмотрение вопроса о конфигурировании нескольких DHCP серверов мы пока отложим. Переходим к следующему полю:

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | flags | |

Поле flags занимает в заголовке DHCP 16 бит, но в RFC2131 описано использование только первого (старшего) бита, остальные 15 биты называются MBZ (Must Be Zero, должны быть обнулены). Старший бит поля flags называется broadcast и нам необходимо разобрать, как он работает. Как мы видели в анализаторе протоколов, такие пакеты, как DHCPOFFER, DHCPACK, DHCPNACK посылаются сервером широковещательно, но мы говорили о том, что такие пакеты могут посылаться и направленно. Дело в том, что когда DHCP сервер предлагает клиенту IP адрес в пакете DHCPOFFER (каким полем это делается, мы разберемся позднее, они ниже в заголовке), клиенту УЖЕ разрешается принимать IP пакеты, предназначенные для этого IP адреса. Но некоторые стеки протоколов НЕ могут принимать пакетов, пока не получат адрес «окончательно» и в таком случае сервер посылает свои DHCP сообщения в рамках DHCP транзакции широковещательно. У клиента есть инструмент для того, чтобы показать серверу, готов ли он (клиент) принимать unicast IP пакеты до окончания процедуры назначения адреса – этот самый бит broadcast. Если клиент готов принимать unicast пакеты в этом случае, RFC2131 требует, чтобы клиент не устанавливал данный бит, в противном случае клиент должен его устанавливать. Если в пакете, полученном от клиента этот бит равен 1, сервер обязан отвечать на такой пакет широковещательно, в противном случае сервер обязан отвечать на такой пакет unicast пакетом. Во взаимодействиях, которые мы наблюдали на прошлых занятиях, клиент устанавливает данный бит и сервер отвечает широковещательно.

Переходим к следующему четырехбайтовому слову, оно состоит из единственного поля ciaddr.

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | flags | |
| ciaddr | | | |

Данное поле заполняет в своих пакетах только клиент (никогда не заполняется сервером) и только в том случае, если у клиента УЖЕ есть IP адрес, т.е. это поле должно быть равно 00 00 00 00 в пакетах DHCPDISCOVER, так как клиент, посылая такое сообщение не имеет адреса, но может быть заполнено в пакете DHCPREQEST если клиент продлевает аренду адреса, так же заполняется клиентом в пакетах DHCPDECLINE и DHCPRELEASE. Клиент имеет право заполнять это поле, только если может отвечать на ARP запросы по этому IP адресу.

Следующее четырехбайтовое слово снова содержит единственное четырехбайтовое поле – yiaddr.

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | flags | |
| ciaddr | | | |
| yiaddr | | | |

Данное поле может заполнять в своих пакетах только сервер и это поле и есть то место в пакете, где DHCP сервер предлагает или назначает адрес клиенту, разумеется, что это поле крайне важно. Сервер заполняет его в пакетах DHCPOFFER, предлагая клиенту указанный в поле IP адрес и в DHCPACK, назначая тем самым указанный в этом поле адрес клиенту. В пакете DHCPNACK данное поле не заполняется, так как по смыслу данный пакет есть не предложение/назначение адреса, а, напротив, запрещение пользоваться адресом.

И следующее четырехбайтовое слово снова содержит единственное четырехбайтовое поле – siaddr.

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | flags | |
| ciaddr | | | |
| yiaddr | | | |
| siaddr | | | |

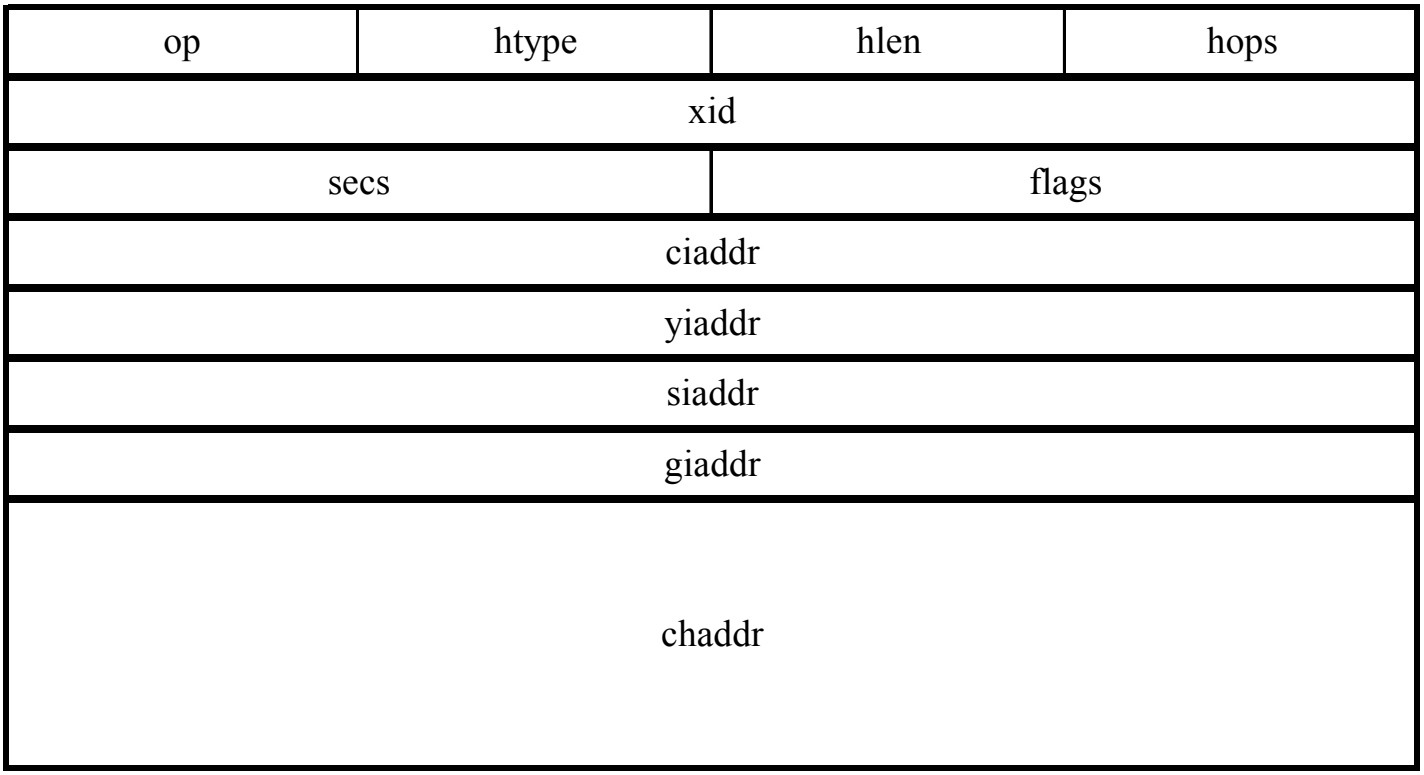
Данное поле заполняется только сервером и только в пакетах DHCPOFFER и DHCPACK, при чем заполняется опционально, т.е. сервер может данное поле и не заполнять. В этом поле сервер может указать клиенту IP адрес DHCP сервера, которым сможет воспользоваться клиент на следующем шаге процесса начальной загрузки. Более детально о возможных способах использования этого поля будет сказано ниже.

Следующее четырехбайтовое слово опять состоит из одного единственного поля – giaddr.

| op | htype | hlen | hops |
|---|---|---|---|
| xid | | | |
| secs | | flags | |
| ciaddr | | | |
| yiaddr | | | |
| siaddr | | | |
| giaddr | | | |

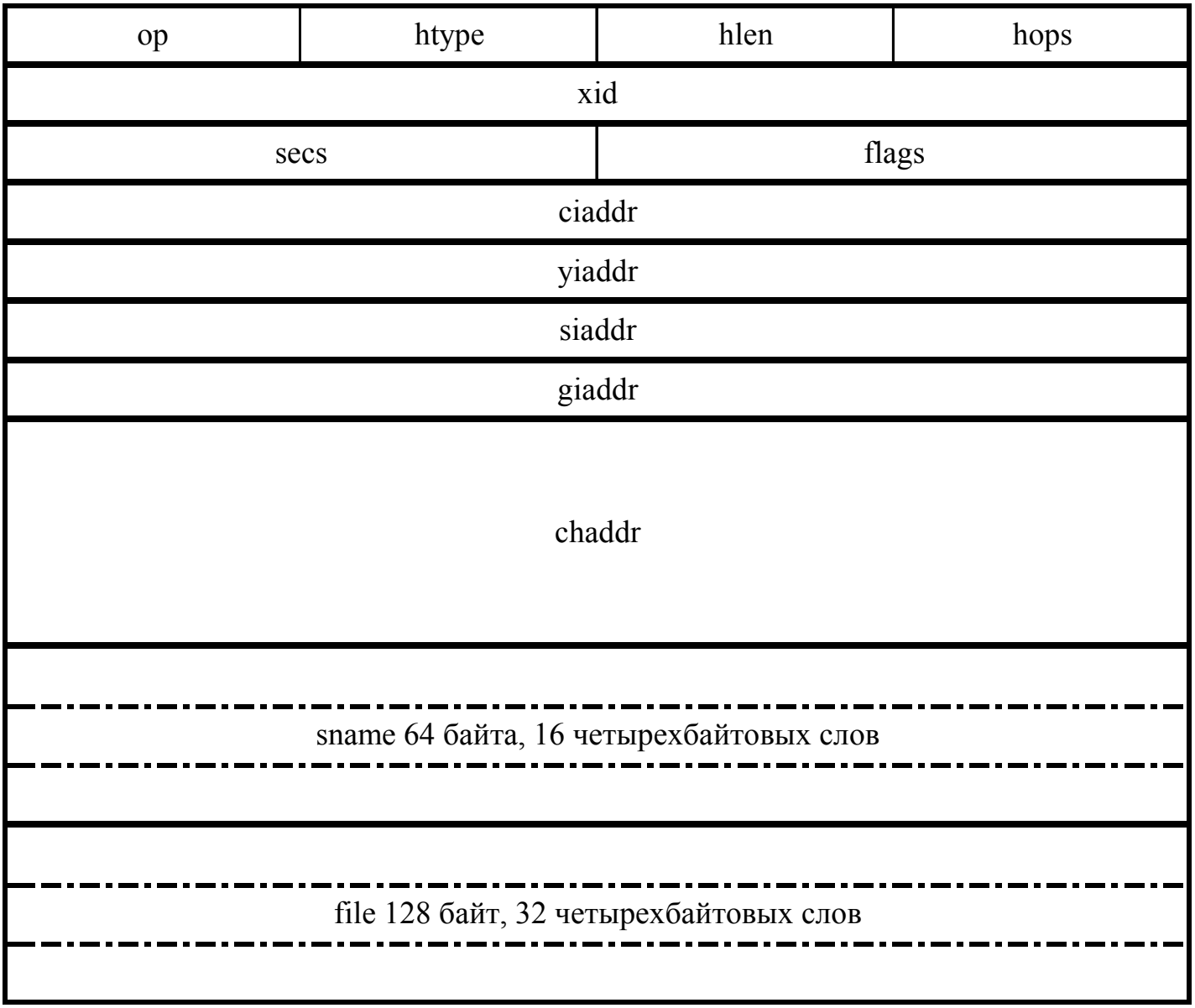
Данное поле не должны заполнять ни клиент, ни сервер, это поле заполняет маршрутизатор, перенаправляющий DHCP сообщения между сетями. Так как мы пока не рассматривали маршрутизацию DHCP сообщений, то смысл данного поля от нас пока скрыт, но когда мы дойдем до изучения маршрутизации сообщений DHCP, мы поймем, сколь важна роль поля giaddr в процессе перенаправления DHCP сообщений между маршрутизаторами.

Следующее поле заголовка DHCP занимает 16 байт, т.е. четыре четырехбайтовых слова и служит для идентификации клиента, посылающего запрос на DHCP сервер.




Клиент должен в этом поле указать свой MAC адрес, идентичный тому, который используется в клиентом в заголовках канального уровня. При этом на данное поле выделено 16 байт для того, чтобы при необходимости клиент, использующий более длинные адреса канального уровня, нежели 6 байт мог заполнить это поле своим канальным адресом. При этом выше мы видели поле hlen, которое показывает серверу, какой на самом деле длины канальный адрес клиента, так что клиент, указав в этом поле свой MAC адрес заполняет остальные байты данного поля нулями, а сервер считывает из этого поля только количество байт, указанное в поле hlen. При этом дублирование в заголовке прикладного протокола информации, которую сервер мог бы извлечь и из заголовка канального уровня, сделано для удобства работы DHCP сервера, которому гораздо проще анализировать MAC адрес клиента в заголовке DHCP, нежели получать сведения по интерфейсу от канального уровня. Отметим, что данный способ указания поля потенциально переменной длины достаточно необычен: например, мы знаем, что в протоколе ARP после поля, аналогичному hlen в заголовке следует поле с MAC адресом и длиной, равной указанной в поле hlen. В DHCP же длина поля с аппаратным адресом фиксирована и велика (16 байт), поле hlen лишь указывает, как много данных из поля фиксированной длины нужно интерпретировать. Впрочем, несложно понять, зачем в заголовке прикладного протокола делать поле «переменной» длины таким образом: это позволяет гарантировать выравнивание полей по четырехбайтовым границам всегда, в независимости от реально используемой длины канального адреса. В пакете DHCPDISCOVER клиент указывает свой канальный адрес в поле chaddr, в дальнейшем клиент указывает значение данного поля во всех своих пакетах, сервер в свою очередь цитирует значение данного поля во всех свои пакетах.

Следующее поле имеет длину 16 четырехбайтовых слов (64 байта) и называется sname. При этом имеет смысл сразу нарисовать и следующее поле file, длина которого составляет 32 четырехбайтовых слова (128 байт). Эти поля придется рисовать не в масштабе, слишком уж они длинные.




Перед тем, как рассмотреть предназначение этих полей, необходимо рассказать о том, что одной из задач, которая ставилась перед предшественником протокола DHCP, протоколом BOOTP (от которого DHCP фактически унаследовал формат пакета). Протокол BOOTP предназначался, в частности, для загрузки операционной системы узлами, лишенными жесткого диска или иного хранилища данных, на котором могла бы размещаться операционная система, так называемых бездисковых станций. Как происходила такая загрузка?

Предполагается, что в сети существует специальный файловый сервер, на котором размещен образ операционной системы, которую необходимо загрузить в память бездисковой станции. При старте станции, не имеющей операционной системы, очевидно, станции все равно необходимо какое-то программное обеспечение, которое будет управлять процессом получения станцией IP адреса, процессом скачивания с файлового сервера образа операционной системы и процессом загрузки полученного из сети образа в оперативную память. Вопрос: где на станции, не имеющей жесткого диска, разместить такое программное обеспечение? Помимо очевидного ответа – на дискете, хотелось бы найти более эффективные варианты, так как с одной стороны дискеты не надежны, с другой стороны раз уж станция бездисковая ☺, то хотелось бы получить решение и в том случае, если на клиентской станции нет и дисковода. Решение достаточно простое: на сетевом адаптере можно разместить микросхему постоянной памяти, в которую будет записано программное обеспечение DHCP клиента и файлового клиента, т.е. фактически BIOS компьютера при старте передает управление еще одному «BIOS», записанному в микросхеме сетевого адаптера. Сразу оговариваем, что не любой сетевой адаптер оборудован такой микросхемой с микропрограммной, готовой управлять загрузкой операционной системой по сети, так как наличие такой микросхемы удорожает адаптер, а использование данной технологии загрузки операционной системы не так распространено, чтобы встраивать микросхему в каждый сетевой адаптер. Данная

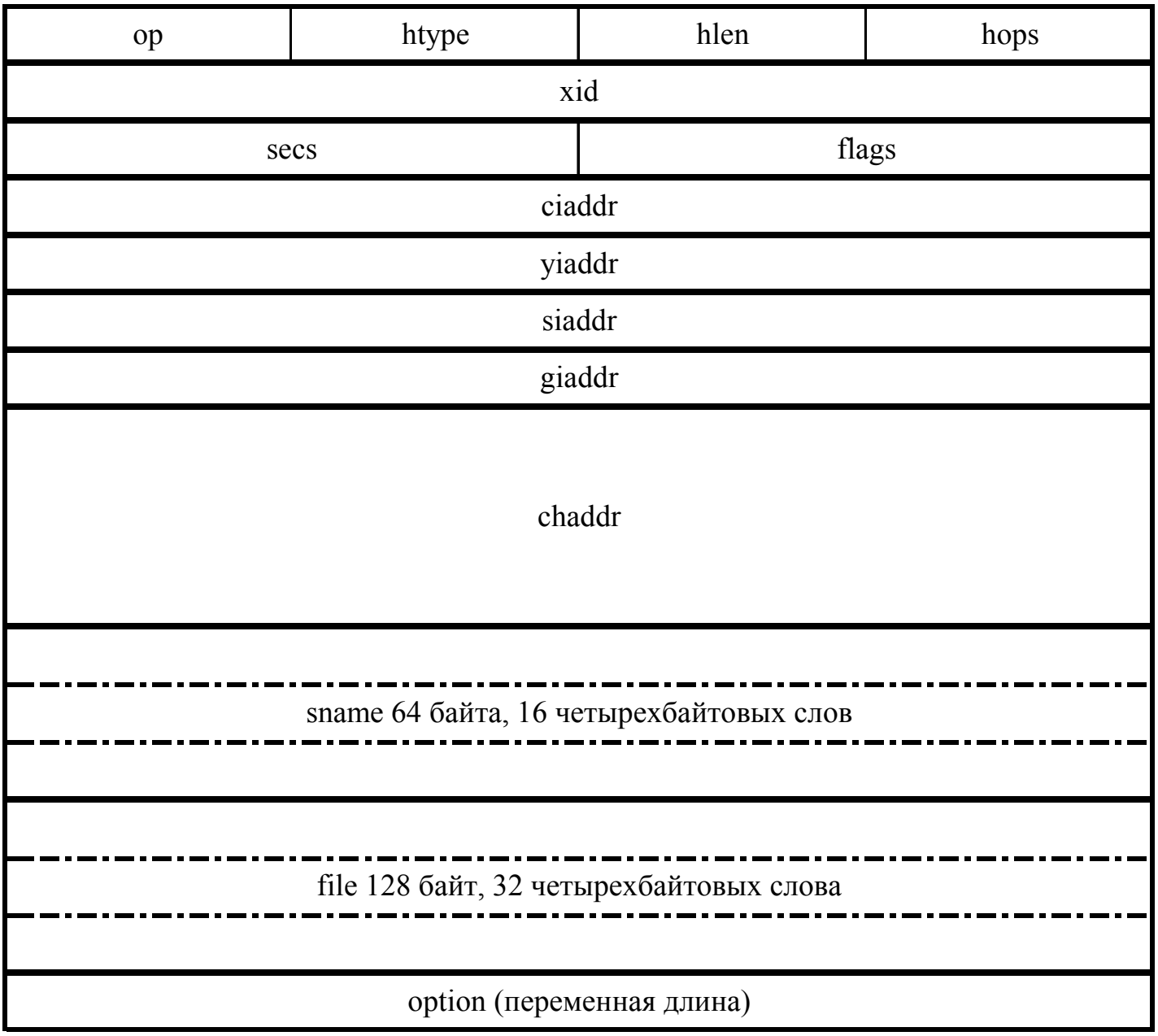
микросхема обычно называется Boot ROM, если в спецификации сетевого адаптера указано поддержка Boot ROM, это значит данный сетевой адаптер может использоваться для загрузки операционной системы по сети. Так же важно отметить, что сетевые адаптеры, встроенные на материнские платы очень часто поддерживают Boot ROM, так как для того, чтобы это реализовать нет необходимости установки дополнительной (удорожающей изделие) микросхемы – в микросхеме системного BIOS обычно достаточно места для того, чтобы разместить там необходимо программное обеспечение Boot ROM. Итак, в Boot ROM необходимо разместить DHCP клиента и клиента файловой службы (для скачивания образа оперативной памяти с файлового сервера). В стеке TCP/IP существует популярная файловая служба FTP, которую мы вскоре будем изучать, но для получения образа оперативной памяти бездисковая станция не использует протокол FTP, так как это весьма сложный протокол, к тому же работающий поверх TCP, а как мы знаем TCP сам по себе весьма сложен и его реализация в относительно небольшом (по объему хранимой информации) Boot ROM не желательна. В стеке TCP/IP есть специальный простой файловый протокол TFTP, который работает поверх UDP и весьма прост в реализации.

Итак, подводим итог по вопросу о методе загрузке бездисковой станции под управлением микросхемы Boot ROM: в микросхеме Boot ROM размещен DHCP клиент и TFTP клиент, после передачи управления содержимому Boot ROM начинает работать DHCP клиент, который получает от доступного DHCP сервера IP и прочие необходимые параметры, например маску подсети. Кроме этого DHCP сервер может предоставить DHCP клиенту дополнительно (опционально) имя FTFP сервера, у которого клиент может получить образ операционной системы и имя того файла, который должен скачать клиент с TFTP сервера. Именно для решения задачи о передаче DHCP клиенту имени сервера TFTP и имени файла, который клиент должен получить, в протоколе DHCP и предусмотрены те два поля, о которых идет речь: поле sname предназначено для передачи имени TFTP сервера, а поле file – для передачи имени файла, который должен скачать клиент с указанного сервера. Эти два поля имеют, как мы видим фиксированную длину, а передаваемые с их помощью сведения, очевидно, могут иметь переменную длину, данная проблема решается очень просто: в конце имени сервера и имени файла устанавливается байт 00, который, очевидно, не может встретиться ни в имени файла, ни в имени сервера, все остальные байты после данного терминирующего байта не интерпретируются клиентом и заполняются нулями (говорят, что содержимое этих полей null-terminated string). Рассмотренное выше поле siaddr так же используется в данной технологии и используется для указания сервером IP адреса TFTP сервера, если оно известно серверу.

Важно отметить, что сегодня ценность такой технологии значительно уменьшилась по сравнению с прежними временами, связано это с ростом объема, занимаемого операционными системами. Действительно, лет 15 назад можно было загрузить на рабочую станцию DOS, занимающий меньше мегабайта, а сегодня операционная система, например Windows 2000 Professional, занимает сотни мегабайт, с одной стороны загружать ее по сети при включении станции крайне долго и приводит к серьезным нагрузкам на пропускную способность сети (а если станций много…), а с другой стороны для того, чтобы загруженная таким образом система работала, на рабочих станциях должен быть установлен крайне большой объем оперативной памяти. Однако данная технология может быть полезна даже не столько для удаленной ЗАГРУЗКИ операционной системы, сколько для организации ИНСТАЛЛЯЦИИ операционной системы по сети. Действительно, мы знаем, что при инсталляции Windows 2000 необходимо запустить на станции небольшую (4 дискеты) мини операционную систему, которая собственно и проводит процедуру инсталляции Windows 2000. Эту мини операционную систему можно запустить с установочного CD, с комплекта установочных дискет, наконец, разместить на жестком диске станции, запустив из существующей операционной системы winnt.exe/winnt32.exe. С помощью DHCP и FFTP можно провести установку Windows 2000 на компьютер, у которого нет привода CD-ROM, дисковода, старой операционной системы: загрузившись с помощью Boot ROM и получив IP адрес, имя TFTP сервера и имя файла образа памяти, который содержит в себе мини ОС, проводящую инсталляцию с помощью DHCP, затем загрузив в память соответствующий образ с помощью TFTP и передав ему управление, можно начать процедуру инсталляции Windows 2000, при этом дистрибутивные файлы могут быть получены мини ОС по сети уже с помощью более совершенного файлового протокола, поддержка которого будет реализована в самой мини ОС. Windows 2000 Server поддерживает такую технологию удаленной инсталляции Windows 2000 Professional, для этого в Windows 2000 Server есть специальная служба RIS (Remote Installation Service), которую мы будем изучить в свое время в

соответствующем курсе. Однако уже сегодня мы разобрали, изучая стандартный протокол DHCP стека TCP/IP все сетевые принципы, лежащие в основе работы службы RIS.

Продолжаем изучение заголовка протокола DHCP. Мы закончили рассмотрение стационарной части заголовка DHCP, после стационарной части заголовка следует поле опций переменной длины, таким образом можно наконец изобразить полный формат заголовка DHCP:




Обсудим теперь принципы форматирования поля «опции», а затем перейдем к изучению конкретных опций DHCP.

Если в заголовке DHCP будут присутствовать опции (по стандарту есть опции, которые обязательно должны присутствовать в заголовке), то первые четыре байта поля опций должны принимать фиксированное значение, так называемое «магическое число» (magic cookie), равное 63 82 53 63 в шестнадцатеричной записи. После этого поля следуют непосредственно опции. Формат опций DHCP напоминает формат опций IP и TCP, но с некоторым отличием. Как и в случае IP/TCP опций, опции DHCP начинаются с однобайтового поля code, показывающего, по сути, тип опции. Снова таки существует два вида опций: состоящие только из поля code и имеющее тело (фиксированной или переменной длины). Опций, состоящих только из поля code всего две, остальные опции имеют поле length и поле с данными самой опции. В отличие от опций протоколов IP и TCP, поле length в опциях DHCP показывает длину ТЕЛА опции, т.е. не учитывает байт code и байт length. В остальном принцип формирования опций DHCP похож на соответствующий механизм протоколов IP/TCP, т.е. в произвольном порядке могут следовать различные опции,

каждая характеризуется типом длинной и телом, если получатель пакета не понимает некоторой опции, то он может пропустить ее, так как знает ее длину и перейти к анализу следующей опции.

Сначала рассмотрим две опции, формат которых предполагает использование только поля code. Первая опция имеет поле code равным 0 и называется pad (заполнитель). Данная опция используется для выравнивания и может быть применена в заголовке DHCP пакета несколько раз. Вторая опция имеет поле code равным 255 (FF) и показывает окончание опций DHCP пакета. Данная опция обязательно должна фигурировать после всех опций пакета, после нее допустимо применение только опции pad. Зачем то нужно: в отличие от заголовков IP и TCP, где длина поля опций оговаривается в стационарной части заголовка, в заголовке DHCP длина не оговаривается, поэтому совершенно необходимо указывать окончание писка опций.

Впредь договоримся записывать значение поля code в десятичной форме, как это принято в RFC2132. Ясно, что перечисленные опции являются исключительно служебными, теперь перейдем к значимым опциям DHCP.

Для начала необходимо провести условную классификацию используемых в DHCP опций. Разделим все опции DHCP на два больших класса:

- Опции, передающие клиенту конфигурационные параметры стека TCP/IP, т.е., собственно говоря и выполняющие прямую задачу протокола DHCP
- Опции, необходимые для правильного обмена данными между DHCP клиентом и DHCP сервером, т.е. направленные на обеспечение функционирования САМОГО протокола DHCP. Такие опции RFC2132 называет DHCP Extension.

Начнем анализ опций с опций второго типа, так как их понимание поможет Вам разобраться в тонкостях работы самого протокола DHCP, уточнив тем самым знания, полученные на предыдущем занятии. Для начала рассмотрим опцию 53, которая называется DHCP Message Type. Как мы знаем в стационарной части заголовка DHCP присутствует поле op, принимающее только два значения – запрос/ответ. Данное поле заимствовано протоколом DHCP у протокола BOOTP (как и вся стационарная часть заголовка), но в протоколе DHCP используется большее количество типов сообщений, и тип сообщения как раз и передается с помощью опции 53. Приводим формат опции и возможные значения из RFC:

```
 Code   Len  Type
+-----+-----+-----+
|  53 |  1  | 1-8|
+-----+-----+-----+
```

Ясно, что с помощью одного байта можно перенумеровать 255 различных типов DHCP сообщений, а нам таких типов известно всего 7, то тело опции имеет длину 1 байт, о чем и говорит поле length. Еще раз подчеркнем, что поле length показывает длину не всей опции, а лишь данных, в IP или TCP заголовке в таком случае поле length было бы равно 3.

Возможные значения:

```
Value   Message Type
-----   ------------
1       DHCPDISCOVER
2       DHCPOFFER
3       DHCPREQUEST
4       DHCPDECLINE
5       DHCPACK
6       DHCPNAK
7       DHCPRELEASE
8       DHCPINFORM
```

Заметим, что есть еще один тип сообщения, о котором мы не говорили ранее – DHCPINFORM, данное сообщение появилось впервые в данном RFC и используется станцией клиентом в том случае, если у станции уже есть настроенный IP адрес (статически), но станция желает получить у DHCP сервера дополнительные конфигурационные параметры, в ответ на такое сообщение сервер должен ответить DHCPACK с передачей данных параметров.

Следующая опция нам тоже уже знакома, с помощью данной опции клиент может в пакете DHCPDISCOVER попросить сервер выдать ему определенный IP адрес, опция называется

Requested IP Address и значение поля code равно 50. Очевидно, что длина опции фиксирована и равна 4 байта, показываем формат опции из RFC.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  50 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

Отмечаем, что данная опция может фигурировать в пакетах только типа DHCPDISCOVER и только в том случае, когда клиент знает, каким адресом он пользовался ранее.

Рассматриваем следующую опцию – Client-Identifier. Данная опция предназначена для того, чтобы клиент уникально идентифицировал себя для DHCP сервера, который в свою очередь использует сведения из данной опции как уникальный идентификатор, к которому сервер привязывает выдачу того или иного IP адреса. Формат данной опции по RFC2132:

```
Code   Len   Type  Client-Identifier
+-----+-----+-----+-----+-----+---
|  61 |  n  |  t1 |  i1 |  i2 | ...
+-----+-----+-----+-----+-----+---
```

Станция может указать в качестве своего уникального идентификатора свой аппаратный адрес, в таком случае поле type (третий байт) должно совпадать с полем htype заголовка DHCP. Кроме этого станция может указать произвольный идентификатор, в таком случае поле type в опции должно быть равно нулю. Необходимо, чтобы все DHCP клиенты в сети использовали различные значения данной опции. Отметим, что чаще всего клиент указываем в данной опции свой MAC адрес, поэтому длина данной опции составит 7 байт (байт type и 6 байт MAC адреса). При этом длина опции не может быть менее двух (байт type и байт идентификатора). Подчеркиваем, что данная опция крайне важна, так как на основе передаваемых в ней данных сервер производит назначение IP адреса клиенту.

Еще одна важнейшая опция – Server-Identifier. Данная опция, содержащая в себе IP адрес DHCP сервера, обязательно должна включатся сервером в пакет DHCPOFFER для того, чтобы клиент знал, какой сервер сделал ему предложение. В свою очередь клиент обязан включать данную опцию (цитируя, разумеется, то, что передал сервер) в пакет DHCPREQEST для того, чтобы сервера правильно понимали, кому из них послан данный DHCPREQEST (который посылается широковещательно на втором и третьем уровнях, и следовательно адреса получателя в заголовках второго и третьего уровня не могут служить серверу (серверам) для выяснения того, кому послан данный пакет DHCPREQEST. Кроме того, информацию, полученную из данной опции клиент должен использовать в том случае, если хочет послать DHCP серверу unicast сообщение. Показываем формат данной опции:

```
Code   Len          Address
 +-----+-----+-----+-----+-----+-----+
 |  54 |  4  |  a1 |  a2 |  a3 |  a4 |
 +-----+-----+-----+-----+-----+-----+
```

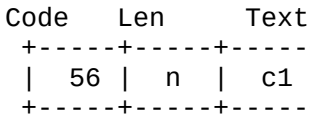
Очевидно, что так как данная опция передает один IP адрес ее длина фиксирована и равна четырем.

Еще одна опция – Maximum DHCP Message Size. С помощью этой опции клиент в сообщениях DHCPDISCOVER или DHCPREQEST (но не DHCPDECLINE) может при желании сообщить серверу какого максимального размера DHCP сообщения может принять клиент. Минимальная допустимая длина – 576 байт. Формат опции:

```
 Code   Len     Length
+-----+-----+-----+-----+
|  57 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

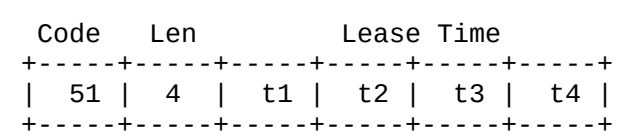
Длина опции – два байта, сами два байта тела опции рассматриваются как целое беззнаковое число, показывающее максимально длину DHCP сообщения в байтах.

Следующая опции – Message. С помощью этого сообщения сервер в сообщении DHCPDECLINE или клиент в сообщении DHCPDECLINE могут передать текстовое сообщение, поясняющее ошибку, данное сообщение программное обеспечение клиента или сервера должно вывести на доступное устройство вывода. Формат опции:



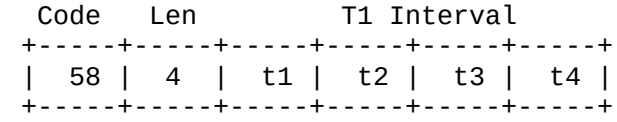

Длина данной опции переменна и зависит от того, как много текста необходимо передать, минимальная допустимая длина опции – 1 байт.

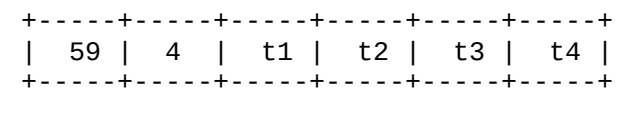
Далее напоминаем, что мы говорили ранее о сроке, на который выдается IP адрес сервером DHCP – использование конечного срока аренды позволяет серверу вернуть адреса скоп если клиент неожиданно, без посылки DHCPRELEASE, отключается. Каким образом клиенту передается время, на которое он получает адрес? С помощью опции с полем code = 51, данная опция называется IP Address Lease Time. Формат данной опции:




Данное время передается с помощью 32 битового целого беззнакового числа, выраженного в секундах. Так как синхронизация часов между клиентом и сервером вообще говоря отсутствует, данное время является относительным – сервер говорит клиенту, на какое количество секунд он выдает клиенту IP адрес в аренду. Использование значения FF FF FF FF означает неограниченное время аренды, но пользоваться таким стилем выделение адресов не рекомендуется, так как в таком случае могут возникать проблемы с доступным пулом адресов сервера (причины детально рассматривались на прошлом занятии). Использование 32 бит для передачи времени аренды позволяет выделить время в диапазоне от 1 секунды до примерно 136 лет (+ возможность выделения адреса без ограничений времени), что полностью удовлетворяет потребности протокола DHCP.

Для того, чтобы в случае преждевременного отключения клиента (без посылки DHCPRELEASE) сервер мог скорее освободить занимаемый клиентом адрес, клиенту сообщается помимо аренды еще два времени, через которые клиент должен подтвердить использование им адреса. Обычно первое из этих времен равно половине времени аренды, а второе равно 87.5 % времени аренды. Эти два времени передаются клиенту в пакетах DHCPOFFER и DHCPACK и называются Renewal Time (T1) и Rebinding Time (T2). Для передачи данных параметров используются еще две опции, с кодами 58 и 59, приводим формат этих опций:




Отмечаем, что данные опции точно так же используют относительные 32 битные времена, выраженные в секундах, как и опция с кодом 51.

Рассматриваем еще одну опцию, относящуюся к типу DHCP Extension. Данная опция позволяет клиент правильно интерпретировать DHCP сообщения в следующем случае: если рассмотренные выше поля sname и file не могут быть использованы для выполнения своей задачи (например, вследствие недостаточности длины соответствующих полей), то с помощью специальных опций можно перенести передачу имени сервера и имени файла в область опций DHCP. Очевидно, опции, передающие имя файла и имя сервера не должны относится к тому типу сообщений, которые мы сейчас рассматриваем, так как эти опции в чистом виде предназначены для

передачи клиенту конфигурационных параметров, но необходимо как то показать клиенту, что значения полей sname и file перегружаются с помощью соответствующих опций, а для этого используется еще одна опция, сообщающая клиенту о том, что поля sname и file перегружены, а эта опция как раз относится к типу DHCP Extension, так как предназначена для того, чтобы обеспечить функционирование САМОГО протокола DHCP. Данная опция использует поле code, равное 52 и называется Option Overload. Записываем формат данной опции:

| Code | Len | Value |
|---|---|---|
| 52 | 1 | 1/2/3 |

Значение поля Value, равное 1 означает, что поле file перегружается с помощью соответствующей опции, значение поля Value, равное 2 означает, что поле sname перегружается с помощью соответствующей опции, значение поля Value, равное 3 означает, что и поле file и поле sname перегружаются с помощью соответствующих опций.

Далее рассматриваем крайне важную опцию, которая называется Parameter Request List. Код этой опции – 55. Необходимо хорошо понять смысл этой опции: DHCP сервер может передать клиенту множество конфигурационных параметров, при этом далеко не каждый DHCP клиент готов принять все или многие из этих параметров даже если они будут переданы клиенту от сервера. Для того чтобы клиент и сервер могли договориться о том, что сервер не передает клиенту тех параметров, которые клиент все равное не сможет принять (для экономии пропускной способности линий связи, процессорных мощностей как клиента так и сервера), клиент в своих пакетах DHCPDISCOVER и DHCPOFFER использует данную опцию, в теле данных которой клиент последовательно перечисляет коды тех опций, которые он может принять от сервера. Если сервер может предложить клиенту опцию, которую включил в свой Parameter Request List клиент, то эта опция предлагается, если же сервер не может предложить опции, которую клиент готов принять, равно как и в случае, если, сервер готов предложить клиенту опцию, которую он не готов принимать, то в таких случаях сервер не включает такую опцию в свои пакеты DHCPOFFER и DHCPACK. Важно отметить, что клиент не должен включать в свой Parameter Request List опции того типа, которые мы сейчас рассматриваем (т.е. DHCP Extensions), в список Parameter Request List включаются только те опции, которые предназначены для конфигурирования стека TCP/IP на стороне клиента.

Так же отмечаем, что существует еще одна опция DHCP Extension, которая называется Vendor Class Identifier, но об этой опции мы будем говорить позднее.

Итак, подводим итог по изученным DHCP Extension, еще раз напоминаем все опции этого типа, напоминаем, что эти опции не являются конфигурационными параметрами стека, которые сервер передает клиенту, а предназначены для организации функционирования самого протокола DHCP.

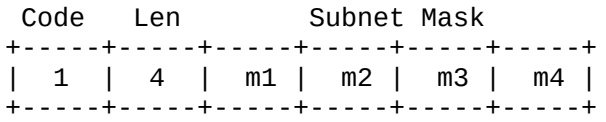
Теперь переходим к рассмотрению опций первого типа, т.е. опций, которые позволяют передать клиенту те или иные параметры стека TCP/IP. Опции такого типа можно разделить на несколько подтипов, будем рассматривать эти подтипы и соответствующие опции.

Разумеется рассмотрение всех опций не должно быть проведено по следующим причинам:

- Их слишком много, а при необходимости можно воспользоваться RFC как справочником.
- Многие опции не представляют сегодня интереса, так как касаются мало используемых технологий
- Многие опции пока не понятны Вам, так как те технологии, которые конфигурируются с помощью этих опций еще не изучены

Вывод: мы будем рассматривать только часть опций, те, которые Вы сегодня можете хотя бы отчасти понять. Начнем рассмотрение с самых первых опций, которые унаследованы протоколом DHCP от своего предшественника – протокола BOOTP. Эти опции принято называть BOOTP Vendor Extension или DHCP Vendor Extension (отмечаем, что опции Pad Option и End Option так же относятся к этому подтипу).

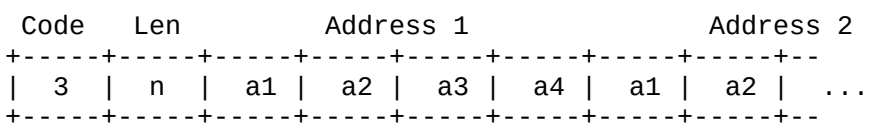
Рассматриваем сначала опцию с кодом 1 – Маска подсети. Смысл данной опции очевиден – с ее помощью узлу передается маска подсети, которая будет использоваться вместе с IP адресом, который передан узлу в стационарной части заголовка DHCP. Формат опции:




Очевидно, что длина опции всегда составляет 4 байта, любой современный клиент, разумеется, понимает такую опцию. Более того, при конфигурировании сервера мы еще не рассматривали настройку опций, но при этом оснастка управления DHCP сервером Microsoft «заставила» нас задать данную опцию так, как будто это обязательное свойство скопа адресов.

Отмечаем, что DHCP может быть применен не только для настройки параметров стека TCP/IP, в частности с помощью опции к кодом 2 можно сконфигурировать часовой пояс, в котором находится клиент (если клиент сам не имеет такой информации). Для этого с помощью этой опции можно передать клиенту сдвижку времени в секундах относительно UTC выраженную 32 битным числом, так что длина опции будет всегда равна 4. Формат записывать не стоит, эта опция не слишком важна, отметим лишь, что RFC2132 является источником справочных сведений об опциях DHCP и при необходимости с его помощью можно уточнить формат любой опции.

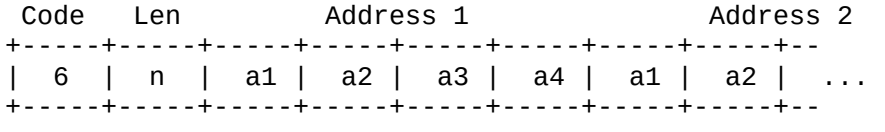
Опция с кодом 3 крайне важна, отмечаем, что с ее помощью узлу можно передать несколько адресов маршрутизаторов, которые узел сможет использовать в качестве шлюзов по умолчанию для отправки IP пакетов в удаленные сети. Формат опции:




С помощью опции можно последовательно передать адреса нескольких шлюзов, причем сервер должен перечислять их в том порядке, в котором их должен предпочитать клиент, ясно, что длина данной опции кратна четырем и не менее четырех. Обычно эту опцию все клиенты могут принимать от сервера.

С помощью опций с кодами 4 и 5 DHCP сервер может передать клиенту соответственно адреса серверов времени стандарта RFC868 (Time Protocol, очень примитивный, сегодня не используется) и адреса серверов имен, работающих в рамках технологии, описанной в IEN 116, которая сегодня так же не применяется.

Опция с кодом 6 очень важна и применяется для передачи клиенту адресов серверов DNS имен, как мы уже говорили вкратце, именно эта технология символьных имен крайне активно используется сегодня в стеке TCP/IP. DNS – это следующая служба (и прикладной протокол), которую мы будем изучать в данном курсе, пока же отметим, что с помощью данной опции клиенту можно передать IP адреса нескольких серверов DNS имен в порядке предпочтения использования клиентом. Формат опции:

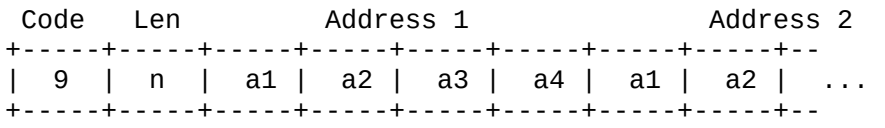



Ясно, что длина опции кратна 4 и не менее 4. Отметим, что вследствие важности технологии DNS практически любой DHCP клиент готов принимать от DHCP сервера данную опцию.

Опция 7 предназначена для сообщения клиенту адреса сервера, на который клиент с помощью специального протокола, не описанного в RFC может хранить логи.

С помощью опции с кодом 8 можно сообщить клиенту адреса quotd серверов.

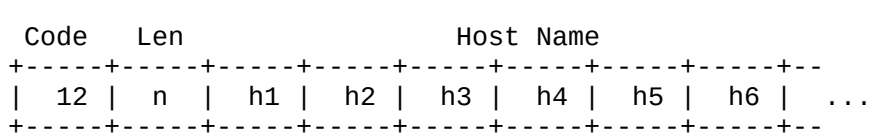
Опция с кодом 9 позволяет серверу сообщить клиенту адреса серверов печати, работающих с помощью протокола LPR. Данный протокол предназначен для отправки заданий на печать и управления сетевыми принтерами в стеке TCP/IP и будет изучен нами в данном курсе. Формат опции:



Длина опции кратна четырем, минимальная длина опции четыре, под адресами понимаются IP адреса серверов печати.

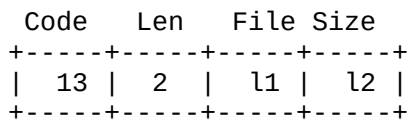
С помощью DHCP опций с кодами 10 и 11 клиенту можно предоставить адреса серверов определенных служб, которые сегодня практически не используются.

Рассматриваем DHCP опцию с кодом 12. С помощью данной опции станции можно передать имя узла (неструктурированное плоское имя, или полное структурированное доменное имя узла). Формат доменных имен мы будем изучать при рассмотрение системы имен DNS. Формат опции:



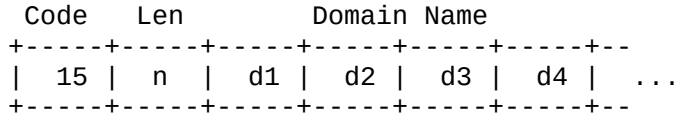

Длина опции произвольная, но не меньше одного байта, тело опции – текстовое имя узла, набор знаков будет изучен нами при изучении системы доменных имен DNS.

Рассматриваем следующую опцию – Boot File Size Option. С помощью данной опции узлу можно сообщить размер файла образа памяти, который узел будет от TFTP сервера. Длина файла задается двумя байтами и выражена в количестве 512 байтовых блоков. Формат опции:




С помощью опции с кодом 14 узлу можно сообщить полное имя файла, в который узел должен записать отчет в случае краха системы (dump file). Данная опция используется редко и останавливаться на ней нет необходимости, просто упоминаем.

Рассматриваем опцию с кодом 15 – с ее помощью узлу можно передать его доменное имя – часть структурированного имени узла в рамках системы доменных имен DNS. Еще раз отмечаем, что система доменных имен DNS будет нами рассмотрена в ближайшее время. Формат опции:

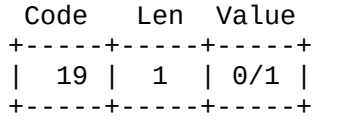



Длина опции не менее одного байта, тело опции – строка с именем домена имен.

Опция с кодом 16 позволяет указать клиенту адрес сервера, на котором клиент может разместить свой файл виртуальной памяти, данная опция особого интереса для нас не представляет. Опции с кодами 17 и 18 рассматривать так же нет особого смысла.

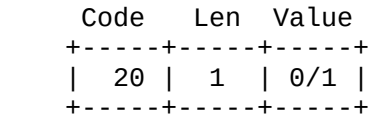
На этом рассмотрение опций, относящихся к подтипу DHCP Vendor Extension окончено. Рассматриваем следующий подтип – параметры связанные с уровнем IP для узла (IP Layer Parameters per Host. Все данные параметры будут использоваться не только тем интерфейсом, который получил DHCP опцию, но узлом как таковым.

Рассматриваем опцию с кодом 19. Данная опция может разрешить или запретить узлу выполнять перенаправление IP пакетов, т.е. работать в роли маршрутизатора IP пакетов. Формат опции:



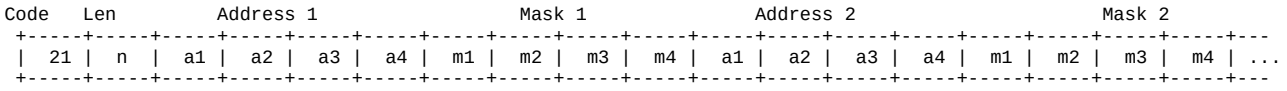

Значение поля данных равное нулю означает, что узел не должен перенаправлять IP пакетов, значение 1, напротив, позволяет узлу перенаправлять IP пакеты при необходимости.

Опция с кодом 20 разрешает или запрещает узлу перенаправлять IP пакеты с маршрутом, указанным источником. Формат опции:



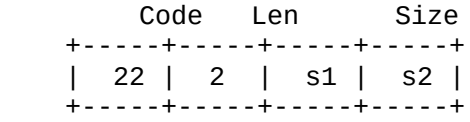

Интерпретация значения поля данных такая же, как и в предыдущей опции.

С помощью опции с кодом 21 можно сконфигурировать пары адрес сети/маска, в которые узел может перенаправлять пакеты с маршрутизацией от источника. Все пакеты с маршрутизацией от источника, у которых адрес следующего маршрутизатора не принадлежит перечисленным в опции парам адрес/маска должны быть отброшены узлом. Формат опции:

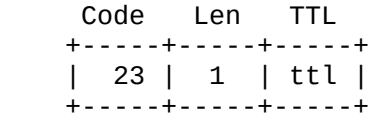



Длина опции должна быть кратна восьми байтам, минимальная длина опции – 8 байт.

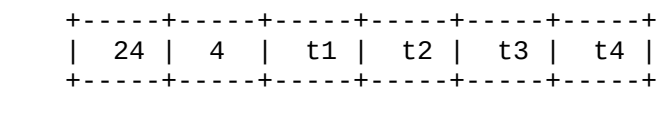
Опция с кодом 22 сообщает узлу, IP пакеты какого минимального размера узел должен быть готов принимать целиком или фрагментированными, поле данных указывает данную величину в байтах. Минимальное значение, передаваемое узлу не должно быть меньше 576 байт (вспомнить, почему). Формат опции:



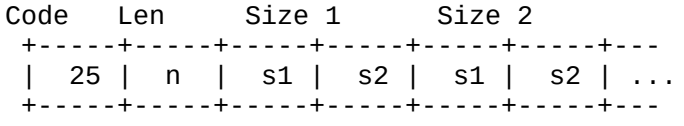

С помощью опции с кодом 23 можно передать узлу значение TTL, которое узел будет по умолчанию устанавливать во всех передаваемых IP пакетах. Формат опции:




Если узел готов выполнять перед коммуникациями с удаленными узлами процедуру определения Path MTU, то с помощью опции с кодом 24 узлу можно передать таймаут, в течение которого будут устаревать изученные узлом MTU. Формат опции:




Значение поля Timeout задается в секундах. При этом с помощью DHCP опции с кодом 25 узлу можно сообщить дискретный список MTU, которые узел будет пробовать использовать в процессе изучения MTU некоторого пути, формат опции:




Каждое возможное значение MTU записывается в виде 2 байтового числа и выражено в байтах, минимально возможное значение, которое сервер может передавать ограничено 68 байтами.

На этом рассмотрение опций, относящихся к подтипу IP Layer Parameters per Host окончено. Рассматриваем следующий подтип – параметры связанные с уровнем IP для интерфейса (IP Layer Parameters per Interface). Все эти параметры будут использоваться только тем интерфейсом, который получил DHCP сообщение с данной опцией, но не другими интерфейсами узла.

С помощью опции с кодом 26 можно указать интерфейсу его MTU в байтах. Минимальное допустимое значение равно 68 байт, формат опции:

```
Code   Len      MTU
   +-----+-----+-----+-----+
   |  26 |  2  |  m1 |  m2 |
   +-----+-----+-----+-----+
```

С помощью опции с кодом 27 клиенту можно сообщить, должен ли он считать, что все сети составной сети, членом которой является клиент имеют тот же MTU, что и MTU самого клиента, или в сети могут существовать сети с меньшим MTU. Во втором случае клиент либо будет выполнять технику Path MTU Discovery, либо не снабжать свои пакеты флагом DF, это остается на усмотрение клиента. Формат опции:

```
    Code   Len  Value
   +-----+-----+-----+
   |  27 |  1  | 0/1 |
   +-----+-----+-----+
```

Значение в поле данных, равное 0 означает, что в сети, возможно, есть сети с меньшим MTU, значение 1 означает, что в се сети имеют тот же MTU, что и интерфейс клиента, который получил DHCP сообщение с данной опцией. Еще раз отметим, что поведение клиента при этом в RFC не регламентировано, его лишь снабжают информацией.

С помощью опции с кодом 28 клиенту можно сообщить широковещательный адрес его подсети, впрочем, как мы знаем, современные клиенту в этом вряд ли нуждаются, так как могут рассчитать такой адрес сами на основании знания IP адреса и маски подсети, так что такая опция сегодня вряд ли может представлять интерес. Формат опции:

```
    Code   Len     Broadcast Address
   +-----+-----+-----+-----+-----+-----+
   |  28 |  4  |  b1 |  b2 |  b3 |  b4 |
   +-----+-----+-----+-----+-----+-----+
```

Длина опции, очевидно, всегда равна 4 байта, данные – широковещательный адрес в подсети клиента.

С помощью опций с кодами 29 и 30 можно сообщить узлу, в какой мере он должен поддерживать работу с ICMP сообщениями типов 17/18, т.е. запрос маски подсети, ответ на запрос маски подсети. С помощью опции 29 конфигурируется, должен ли клиент посылать такие запросы, а с помощью опции с кодом 30 конфигурируется, должен ли клиент отвечать на подобные запросы. Напоминаем, что технология запросов/ответов масок подсетей фактически нужна была для работы в паре с протоколом RARP, и поэтому сегодня данные сообщения практически не используются как раз вследствие популярности протокола DHCP, так что особой практической ценности данные опции не представляют. Формат опций:

```
    Code   Len  Value
   +-----+-----+-----+
   |  29 |  1  | 0/1 |
   +-----+-----+-----+
```

```
    Code   Len  Value
   +-----+-----+-----+
   |  30 |  1  | 0/1 |
   +-----+-----+-----+
```

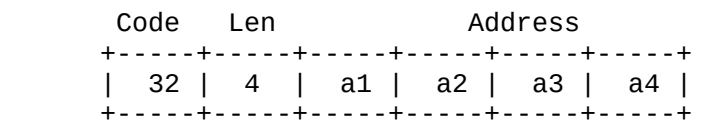
Значение 1 в обоих случаях означает поддержку данной технологии, значение 0 – отсутствие поддержки.

Следующие две опции конфигурируют клиента на использование ICMP сообщений типа 9/10 – объявления маршрутизаторов/обнаружение маршрутизаторов. С помощью опции с кодом 31 можно включить использование клиентом технологии поиска маршрутизаторов, формат опции:

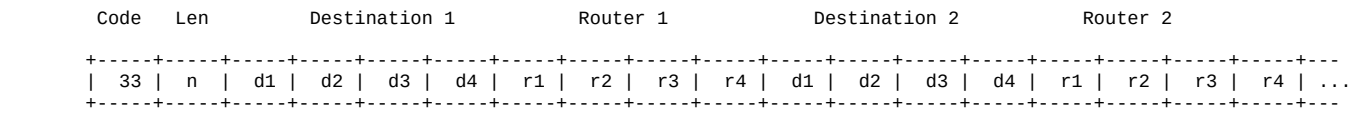
```
    Code   Len  Value
   +-----+-----+-----+
   |  31 |  1  | 0/1 |
   +-----+-----+-----+
```

Значение равное 0 запрещает клиенту пользоваться данной технологий, значение равное 1 включает у клиента использование технологии поиска маршрутизаторов. Важно отметить, что DHCP сам по себе предоставляет механизм конфигурирования маршрутизаторов по умолчанию на узлах, но техника IRDP имеет перед конфигурированием маршрутизаторов посредством DHCP серьезное преимущество – конфигурирование узлов происходит динамически без привлечения администратора, которому в случае с конфигурированием маршрутизаторов по умолчанию с помощью протокола DHCP придется переконфигурировать соответствующую опцию в скопе.

DHCP опция с кодом 32 позволяет сообщить клиенту о том, к какому IP адресу должен посылать клиент свои сообщения ICMP type 10. Ценность данной опции сомнительна, так как такое сообщение можно просто послать на адрес 255.255.255.255 и гарантировано достичь цели, с другой стороны мы знаем, что технология IRDP позволяет использовать в пакетах ICMP type 10 групповые адреса получателей, что снизит нагрузку на вычислительные мощности всех узлов сети, не являющихся маршрутизаторами, объявляющими о себе. И на тот случай, если клиент не знает соответствующего группового адреса, на который необходимо слать сообщения ICMP type 10, то он может получить его с помощью DHCP. Формат опции:




И, наконец, последняя опция в данном подтипе – Static Route Option. С помощью данной опции клиенту можно передать статические записи, которые клиент должен включить в свою таблицу маршрутизации, формат опции:

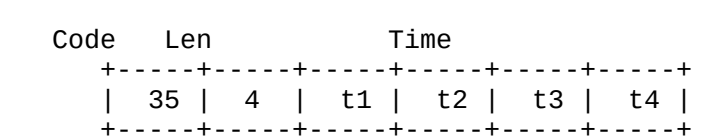



Как видим, каждый маршрут содержит в себе две записи – получатель и следующий маршрутизатор. Видно, что отсутствует маска подсети, что ограничивает данную технологию либо созданием маршрутов в классовые сети, либо к узлам, что не слишком функционально. Минимальная длина опции 8 байт, длина опции кратна 8 байтам. Адрес 0.0.0.0 не может быть использован в качестве адреса получателя (ну конечно и маршрутизатора), для указания маршрута по умолчанию есть другая DHCP опция.

На этом рассмотрение опций, относящихся к подтипу IP Layer Parameters per Interface окончено. Рассматриваем следующий подтип – параметры связанные с канальным уровнем для интерфейса (Link Layer Parameters per Interface), как ясно из названия да и по смыслу функций канального уровня, данные параметры применимы только к тому интерфейсу, который получил данную опцию в DHCP сообщении.

Опция с кодом 34 указывает интерфейсу на специфический способ формирования формата кадров канального уровня, называемый инкапсуляция трейлеров, которую мы не рассматривали, так как он не имеет никакого практического значения сегодня, детально рассматривать данную опцию не имеет смысла.

Опция с кодом 35 позволяет передать узлу таймаут записей в кэше ARP, таймаут выражен в секундах, формат опции:




Опция с кодом 36 позволяет сконфигурировать узлу, какой формат кадров Ethernet он должен использовать по умолчанию, допустимо использование либо кадров Ethernet II либо кадров 802.3. Формат опции:

```
 Code   Len  Value
+-----+-----+-----+
|  36 |  1  | 0/1 |
+-----+-----+-----+
```

Значение поля Value, равное 1 означает использование узлом кадров формата IEEE 802.3, значение 0 требует использования кадров формата Ethernet II.

Следующий подтип DHCP опций – параметры связанные с протоколов TCP.

Опция с кодом 37 передает узлу значение TTL, которое необходимо использовать в заголовке IP пакета при отправке TCP сегментов. Формат данной опции:

```
 Code   Len   TTL
+-----+-----+-----+
|  37 |  1  |  n  |
+-----+-----+-----+
```

Опция с кодом 38 управляет посылкой клиентом TCP сегментов keep alive. Четырехбайтовое значение, передаваемое в качестве данных данной опцией клиент должен использовать как таймер keep alive в секундах во всех TCP соединениях. Если значение поля данных равно нулю, это значит, что TCP не должен использовать технику TCP keep alive, полагаясь на то, что подобные функции будут реализованы в прикладных протоколах и приложениях. Формат опции:

```
Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  38 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

Опция с кодом 39 конфигурирует, должен ли TCP при посылке пактов TCP keep alive передавать «лишний» байт информации. Формат опции:

```
 Code   Len  Value
+-----+-----+-----+
|  39 |  1  | 0/1 |
+-----+-----+-----+
```

Значение, равное 1 заставляет клиента использовать лишний байт, значение равное 0 означает, что использовать лишний байт не следует.

И, наконец, последний подтип DHCP сообщений, которые нам необходимо рассмотреть, это параметры, конфигурирующий прикладные протоколы и службы на узле (Application and Service Parameters). Так как мы еще не рассматривали никакие прикладные протоколы и службы (в отличие от канального, сетевого и транспортного уровней OSI), то большинство параметров не будут прозрачны и понятны до конца. Пока просто перечислим эти опции, затем, говоря о прикладных протоколах, необходимо не забывать упоминать, какие DHCP опции можно передать узлам в связи с тем или иным прикладным протоколом или службой.

Опции с кодами 40 и 41 конфигурируют клиентов службы NIS (Network Information Service), опции 64 и 65 конфигурируют клиентов службы NIS+.

Опция с кодом 42 сообщает клиенту адрес сервера времени NTP (Network Time Protocol), с которым клиент может синхронизировать точное время (сдвижка относительно UTC задается опцией с кодом 2). Данную технологию синхронизации времени использует, например, операционная система Windows XP. Формат опции:

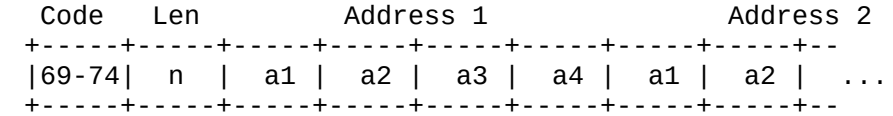
```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  42 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

В поле данных передаются IP адреса серверов NTP в порядке предпочтения клиентом. Пока пропускаем опцию с кодом 43.

Опции с кодами 44, 45, 46, 47 конфигурируют клиентов, поддерживающих работу стека протоколов NetBIOS/SMB поверх TCP/IP. При изучении такого взаимодействия мы будем изучать

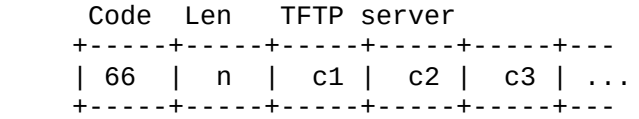
ряд параметров, управляющих работой NetBIOS/SMB поверх TCP/IP, и тогда вспомним, что ряд этих параметров можно сконфигурировать с помощью DHCP.

С помощью опций с кодами 48 и 49 можно передать клиентам параметры настройки X Window. С помощью опций с кодами 69-74 можно передать клиентам адреса доступных для них серверов важных служб стека TCP/IP: SMTP, POP3, NNTP, WWW, Finger, IRC. Формат всех этих опций идентичен:

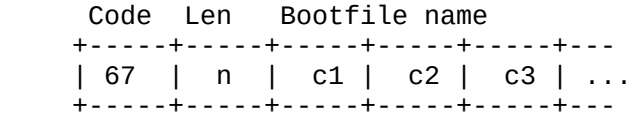



Опции с кодами 75 и 76 служат для конфигурирования клиентов для работы со службой каталогов StreetTalk.

И наконец, рассмотрим еще две опции, которые нами ранее уже упоминались. Мы говорили о том, что поля sname и file стационарной части заголовка DHCP могут быть перегружены с помощью опции с кодом 52, в таком случае указание клиенту имени загрузочного файла и имени TFTP сервера делается с помощью двух опций, с кодами 66 и 67. Рассмотрим опцию с кодом 66, она используется для передачи клиенту имени TFTP сервера, ее формат:




Минимальная длина по TFTP Server равна единице. Опция с кодом 67 служит для указания имени загрузочного файла и ее формат:




Теперь будем подводить итоги по изученному материалу. Самым главным в этом подведении итогов должно стать структурирование представления информации об. Так как опций много и запомнить формат каждой невозможно, да и не нужно, КРАЙНЕ важно, чтобы Вы хорошо разобрались в СТРУКТУРЕ предоставленного материала, а справочные сведения о формате опций всегда доступны в RFC2132.

Итак, пробуем подвести итоги, классифицируем опции:

- Опции, предназначенные для организации обмена данными между DHCP клиентом и DHCP сервером, эти опции необходимы самому протоколу DHCP. Номера 50 – 61 (не рассмотрена опция 60), ну и еще сюда можно отнести опции 0 и 255, хотя формально исторически они относятся к следующему типу
- Опции, предназначенные для конфигурирования параметров стека TCP/IP на клиентах, номера 1 – 49, 64 – 76. Отмечаем, что опция с кодом 43 нами пока не рассмотрена.
  - Опции, специфицированные еще для протокола BOOTP, еще не разделяемые на подтипы, номера 1 – 18.
  - Опции, связанные с протоколом IP для хоста, номера 19 – 25.
  - Опции, связанные с протоколом IP для интерфейса, номера 26 – 33.
  - Опции, связанные с канальным уровнем, номера 34 – 36.
  - Опции, связанные с протоколом TCP для интерфейса, номера 37 – 39.
  - Опции, связанные с приложениями и службами, номера 40 – 49, 64 – 65, 68 – 76.
  - Опции, перегружающие поля заголовка DHCP, номера 66, 67.
- Опции с кодами 62, 64 не определены.
- Опции с кодами 128 – 254 зарезервированы
- Опции с кодами 77 – 127 пока не определены.

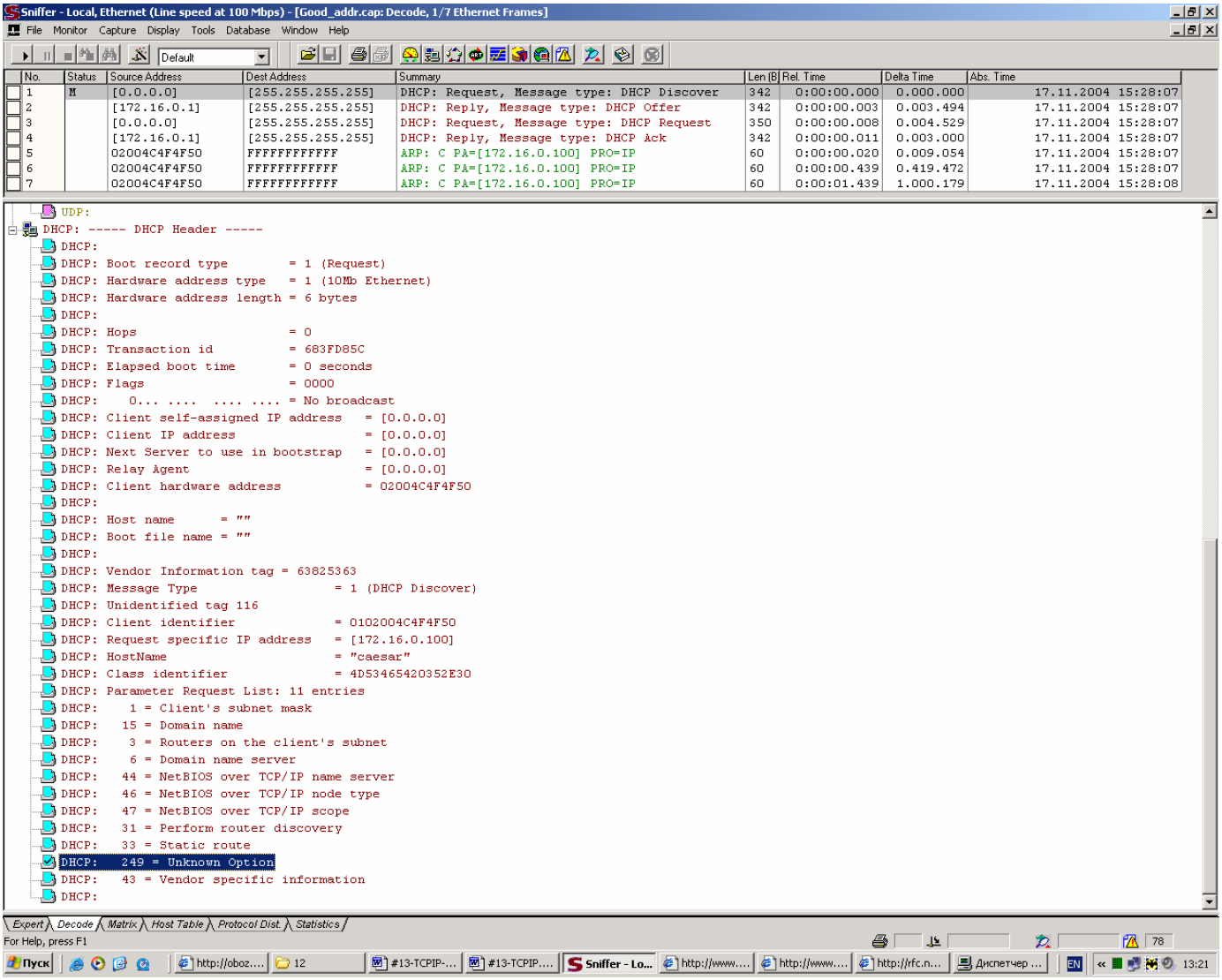
Такой анализ поможет Вам четко представить себе объем изученного материала и структурировать восприятие материала, что, в конечном счете, крайне положительно сказывается

на понимании. Так же отметим, что опции с кодами 43 и 60 мы пока не рассмотрели, но рассмотрим позднее.

Так же важно провести анализ опций вот с какой точки зрения: необходимо разделить опции на те, которые конфигурируют внутренние параметры стека (таймаут ARP кэша, таймаут TCP keep alive etc), т.е. параметры, которые скорее всего так или иначе реализованы в любом стеке, и могут быть перегружены с помощью DHCP или наоборот, проигнорированы и на те параметры, которые передают стеку такие параметры, которые иначе стеку взять неоткуда: маска, шлюз и так далее, т.е. большинство опций. Поэтому опции первого типа клиенты часто могут не поддерживать, так как имеют соответствующие параметры настроенными локально, а опции второго типа стек обычно готов получать с помощью DHCP. Так же важно отметить, что опции, описывающие адреса доступных служб различных типов (WWW, SMTP, POP3 и т.д.) часто не поддерживаются клиентом вследствие того, что в операционной системы обычно предполагается конфигурирование клиентов для работы с этими службами (например, почтовой программы), но не самого стека.

Теперь перейдем к закреплению рассмотренного материала на практике.

Для начала рассматриваем стационарную часть заголовка DHCP, анализируя использование полей заголовка. Воспользуемся примерами трафика из прошлого урока. Сперва рассмотрим обмен между клиентом и сервером в том случае, если на сервере не сконфигурировано никаких опций, т.е. пока будем изучать использование опций типа DHCP Extension. Рассматриваем трафик из файла good_addr.cap. Для начала анализируем пакет DHCPDISCOVER.




Рассмотрите поочередно все поля заголовка DHCP во всех пакетах. Убедитесь, что поле OP принимает только два значения, следовательно, тип пакета оно не показывает. Убедитесь, что поле xid во всех остальных пакетах (DHCPOFFER, DHCPREQEST, DHCPACK) в рамках данной транзакции сохраняется. Подчеркиваем, что не смотря на то, что клиент НЕ устанавливает флаг Broadcast, сервер тем не менее отвечает широковещательно, что вообще говоря противоречит требованиям RFC. Отмечаем, что поле ciaddr не заполнено клиентом, что означает, что адреса он

пока не имеет. Поле yiaddr не заполнено в пакете DHCPDISCOVER. Сразу убедитесь, что это поле заполнено в пакете DHCPOFFER и тем самым сервер делает клиенту предложение IP адреса. Отмечаем, что поля sname и file пусты, хотя и занимают в заголовке 64 и 128 байт соответственно.

Подводим итог – стационарная часть заголовка достаточно проста и позволяет серверу лишь назначить клиенту IP адрес и имя файла для загрузки. Переходим к опциям. Рассматриваем опции, используемые клиентом в пакете DHCPDISCOVER. Первая опция показывает тип пакета, показываем в анализаторе значение опции 35 01 01, что означает: опция Тип пакета, длина данных 1 байт, значение 01, т.е. DHCPDISCOVER. Следующую опцию анализатор пакетов отказался нам пояснить, как видим он заявляет об опции 116. Сразу отмечаем, что если приемная сторона (а анализатор пакетов тоже в некотором роде приемная сторона) не может понять некоторую опцию, он может ее просто пропустить. Действительно, формат опций универсален, второй байт опции означает длину данных (в данном случае 1 байт), так что приемная сторона просто пропускает три байта: 74 01 01 и может анализировать данные дальше. Рассматривая вчера RFC2132 мы не нашли там описания данной опции, она появилась позже и описана в RFC2563. С помощью данной опции сервер может указать клиенту, что в случае, если сервер не может предоставить клиенту IP адреса клиент должен взять себе адрес сам из известного нам диапазона 169.254.0.0/16. Формат опции:

```
 Code   Len   Value
+-----+-----+-----+
| 116 |  1  | 0/1 |
+-----+-----+-----+
```

Работает опция следующим образом: клиент отправляет DHCPDISCOVER с установленной опцией, значение 0 означает, что клиент не поддерживает автоконфигурирование, единица означает, что автоконфигурирование поддерживается. Клиент заявляет об этом серверу. Если сервер не может предоставить клиенту IP адрес в пакете DHCPOFFER, т.е. заполняет поле yiaddr нулями, то клиент, в соответствии с этой же опцией в ответе сервера назначает либо не назначает себе автоматически адрес. Итак, в нашем случае клиент заявляет серверу, что в случае отсутствия возможности предоставления адреса клиент готов использовать автоконфигурирование.

Следующая опция – идентификатор клиента, напоминаем, что ее типовая длина 7 байт: тип аппаратного адреса (в нашем случае – 01) и собственно аппаратный адрес – 6 байт. Этой опции соответствуют байты 3d 07 01 02 00 4c 4f 4f 50. Следующая очень важная опция – IP адрес, который просит себе клиент на основании предыдущего опыта, байты 32 04 ac 10 00 64. Следующая опция – имя хоста сервера, отметим, что анализатор не совсем корректно отображает, какие именно байты принадлежат данной опции, не показывая байты кода опции и длины, а показывая только байты имени хоста клиента. Это опции соответствуют байты 0c 06 63 61 65 73 61 72. Следующая опция номер 60, идентификатор класса, мы ее пока не рассматривали, пропускаем пока и на этот раз. Следующая важнейшая опция – Parameter Request List, с помощью которой клиент заявляет серверу, какие именно опции клиент готов получать от сервера. Как видно из данной опции, клиент готов получить: маску подсети (1), доменное имя (15), адреса шлюзов по умолчанию (3), адреса DNS серверов (6), три параметра, касающиеся работы стека NetBIOS/SMB поверх TCP/IP (44, 46, 47), использование технологии IRDP (31), статические маршруты (33), опцию с кодом 43 (мы ее пока не рассматривали, не рассматриваем и сейчас) и неизвестную ни анализатору ни нам опцию с кодом 249.

В RFC3442 (Dec 2002) описана новая опция, которая заменяет собой опцию с кодом 33, т.е. переопределяет правила передачи клиенту статических маршрутов, ее код 121. Вспомним главный недостаток опции с кодом 33 – невозможность передать в маршруте маску подсети, что сводит на нет полезность данной опции в современной сети. Предложение в RFC3442 избавлено от этого недостатка, формат опции:

```
Code Len Destination 1    Router 1    Destination 2       Router 2
 +-----+---+----+-----+----+----+----+----+----+----+-----+----+----+----+----+----+
 | 121 | n | d1 | ... | dN | r1 | r2 | r3 | r4 | d1 | ... | dN | r1 | r2 | r3 | r4 |
 +-----+---+----+-----+----+----+----+----+----+----+-----+----+----+----+----+----+
```

Данная опция использует сжатый формат представления пары сеть получателя/маска, используя вместо 8 байт для передачи этой пары переменное число байт, от одного до пяти. Формат поля Destination имеет следующий вид: первый его байт показывает сколько бит в маске отвечают

за номер сети (в десятичной форме), а далее перечисляются все ненулевые байты номера сети. В RFC приведены примеры, имеет смысл рассмотреть их сейчас:

```
  Subnet number    Subnet mask      Destination descriptor
   0                0                0
   10.0.0.0         255.0.0.0        8.10
   10.0.0.0         255.255.255.0    24.10.0.0
   10.17.0.0        255.255.0.0      16.10.17
   10.27.129.0      255.255.255.0    24.10.27.129
   10.229.0.128     255.255.255.128  25.10.229.0.128
   10.198.122.47    255.255.255.255  32.10.198.122.47
```

Возникает резонный вопрос: при чем же здесь опция с кодом 249☺? Microsoft в статье, опубликованной по адресу

http://support.microsoft.com/default.aspx?scid=kb%3BEN-US%3B121005

разъясняет, со ссылкой на draft

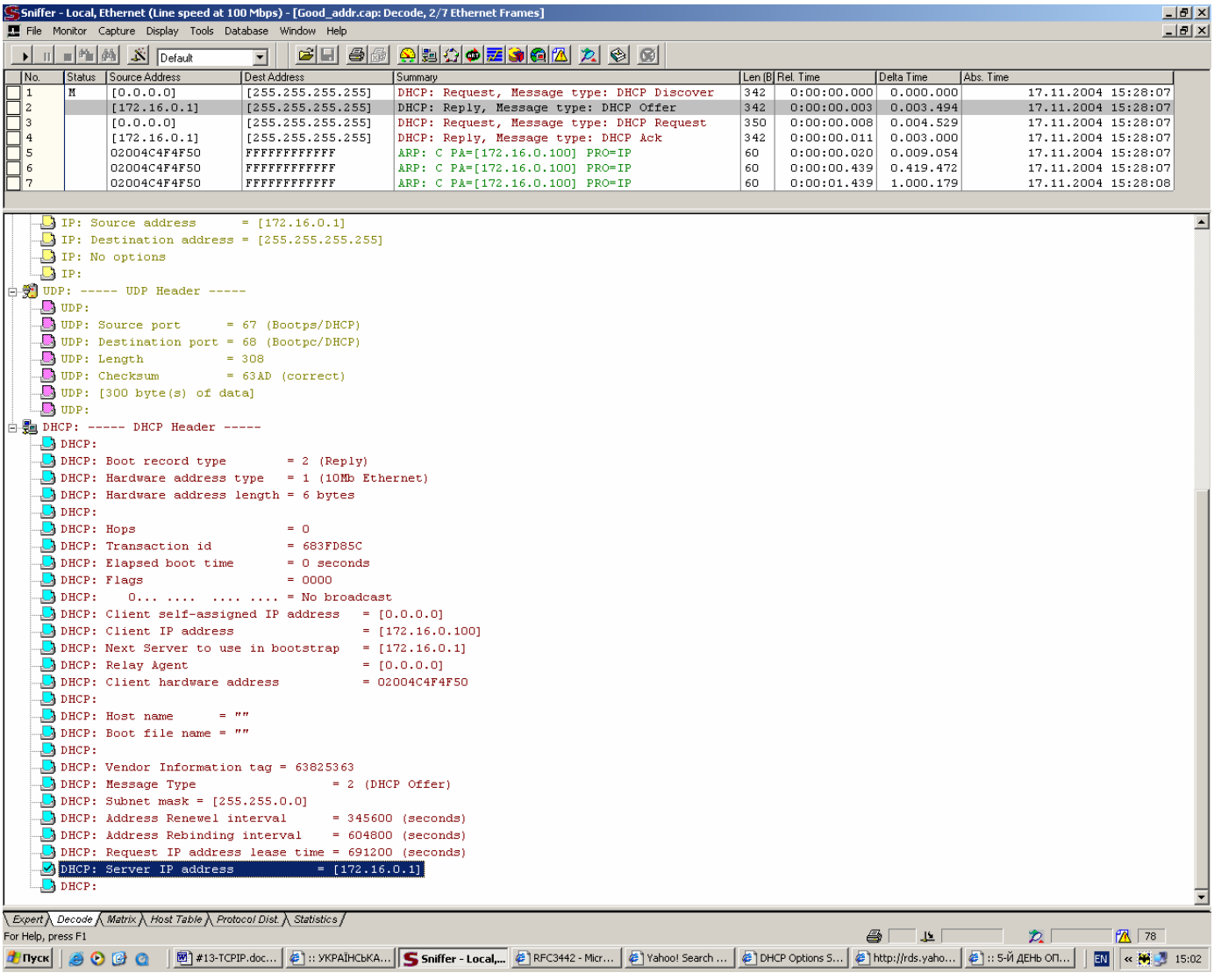
http://www.ietf.org/internet-drafts/draft-ietf-dhc-csr-06.txt (там не доступен, доступен по адресу:

http://gnu.kookel.org/ftp/ftp.isc.org/dhcp/draft-ietf-dhc-csr-06.txt)

что использует эту возможность (тогда еще данный документ имел статус draft и номер 121 опции не был присвоен) с кодом 249 (!!!)

Заканчиваем список опций как и положено байтом 255, сигнализирующим конец списка опций, после чего несколько байтов заполнителя 00 для выравнивания.

Теперь посмотрим, какие же опции использует сервер в своем пакете DHCPOFFER:

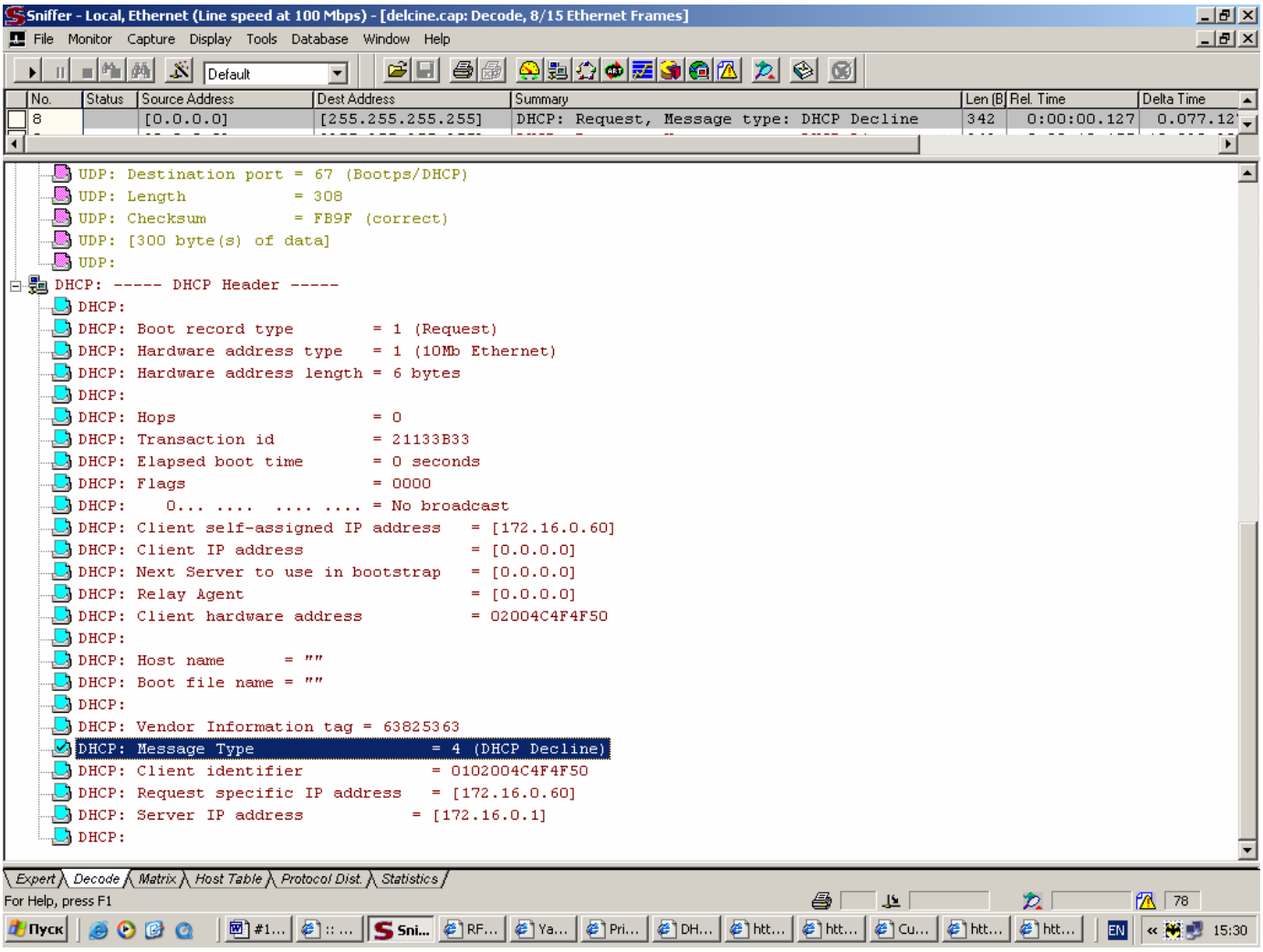



Как видим сервер передает клиенту тип сообщения DHCPOFFER, передает три времени, о которых мы говорили на прошлом занятии, свой IP адрес и маску подсети. Ясно, что только

последняя опция – есть опция, конфигурирующая стек на стороне клиента, остальные опции являются частью DHCP Extension. Это вполне ясно - мы ведь пока и не настраивали скопу ни одной опции, а маска подсети, как уже говорилось в DHCP сервер Microsoft должна быть настроена обязательно при создании скопа.

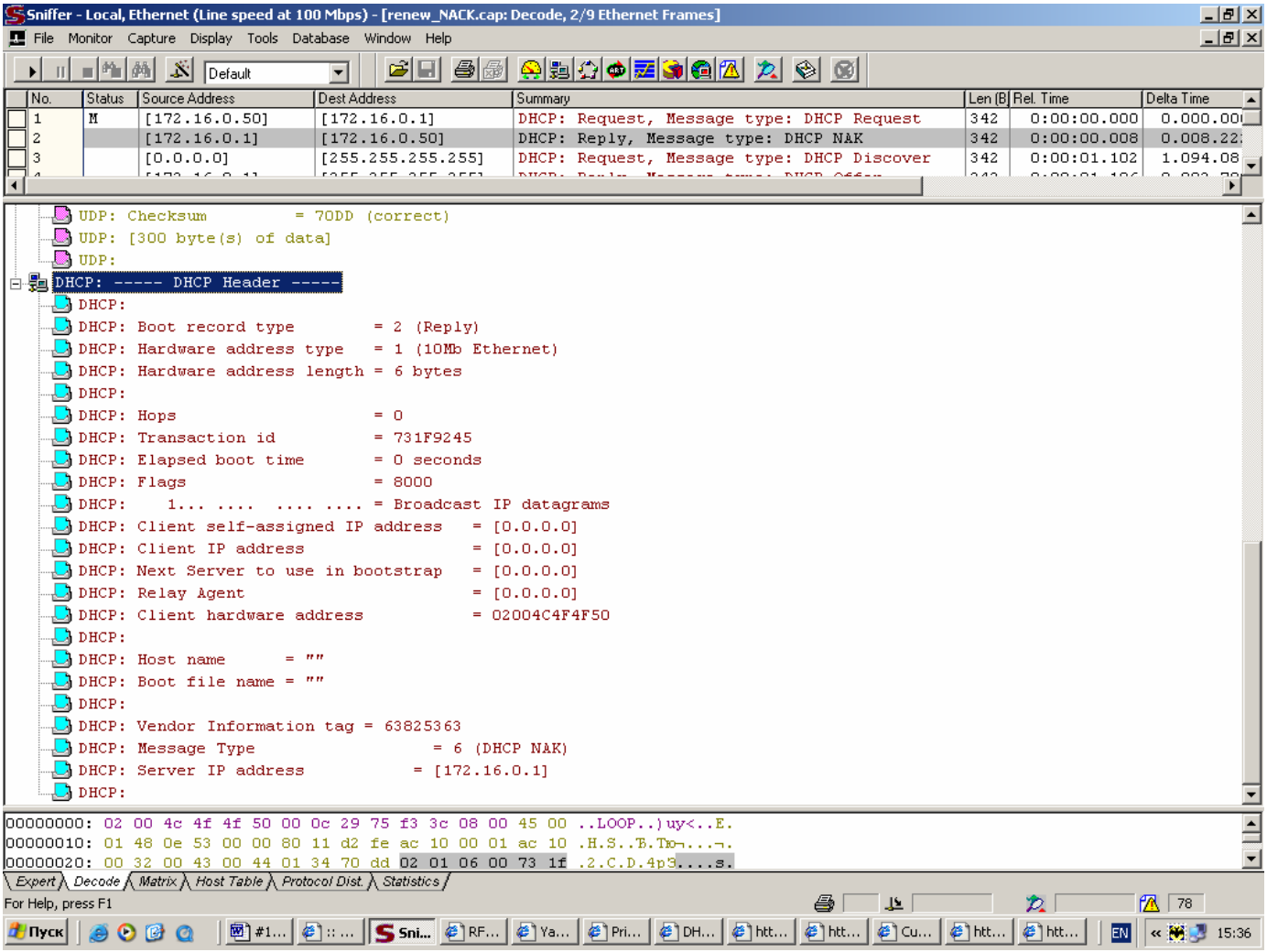
Показываем, что набор опций пакета DHCPREQUEST идентичен набору опций в DHCPDISCOVER за одним исключением – в пакете DHCPREQUEST присутствует еще одна опция с кодом 81, которую снова не может распознать наш анализатор. Данная опция не будет нами сейчас рассмотрена детально, так как она связано с тонкостями работы службы DNS, а взаимодействие между службами DHCP и DNS мы будем рассматривать когда будем изучать службу DNS. Пока скажем, что с помощью этой опции клиент может попросить у сервера записать сведения о нем в базу данных DNS сервера, более детально об этом речь пойдет позднее. Сравниваем опции в пакете DHCPOFFER и DHCPREQUEST и убеждаемся, что опции идентичны за исключением все той же опции с кодом 81.

Итак, мы рассмотрели типовые служебные опции, которые фигурируют в пакетах типа DHCPDISCOVER, DHCPOFFER, DHCPREQUEST и DHCPACK. Сразу покажем служебные опции пакетов прочих типов. Открываем файл decline.cap из прошлого модуля и демонстрируем особенности формирования данного пакета.



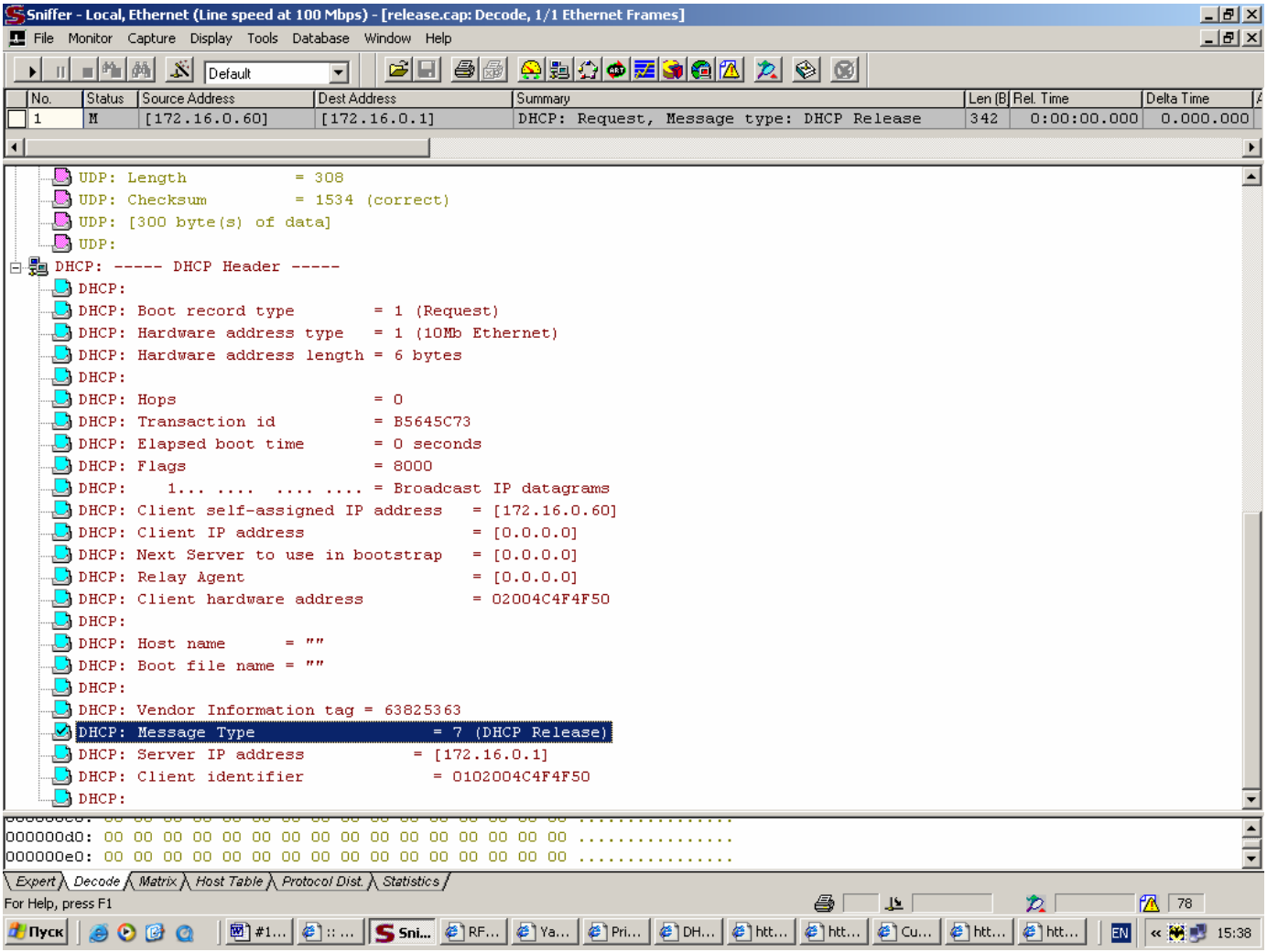

Для начала отметим, что наш клиент заполнил поле ciaddr, хотя по сути адреса у него пока нет, кроме того, пакет посылается широковещательно на канальном уровне и ограниченно широковещательно на сетевом уровне. Установлены служебные опции: клиент, разумеется, показывает тип сообщения DHCPDECLINE, свой идентификатор, повторяет в опции свой неудачный IP адрес и использует опцию адреса сервера для того, чтобы указать, какому именно DHCP серверу посылается данный пакет.

Рассмотрим особенности формирования пакета DHCPNACK. Открываем файл renew_nack.cap из прошлого урока и анализируем опции:



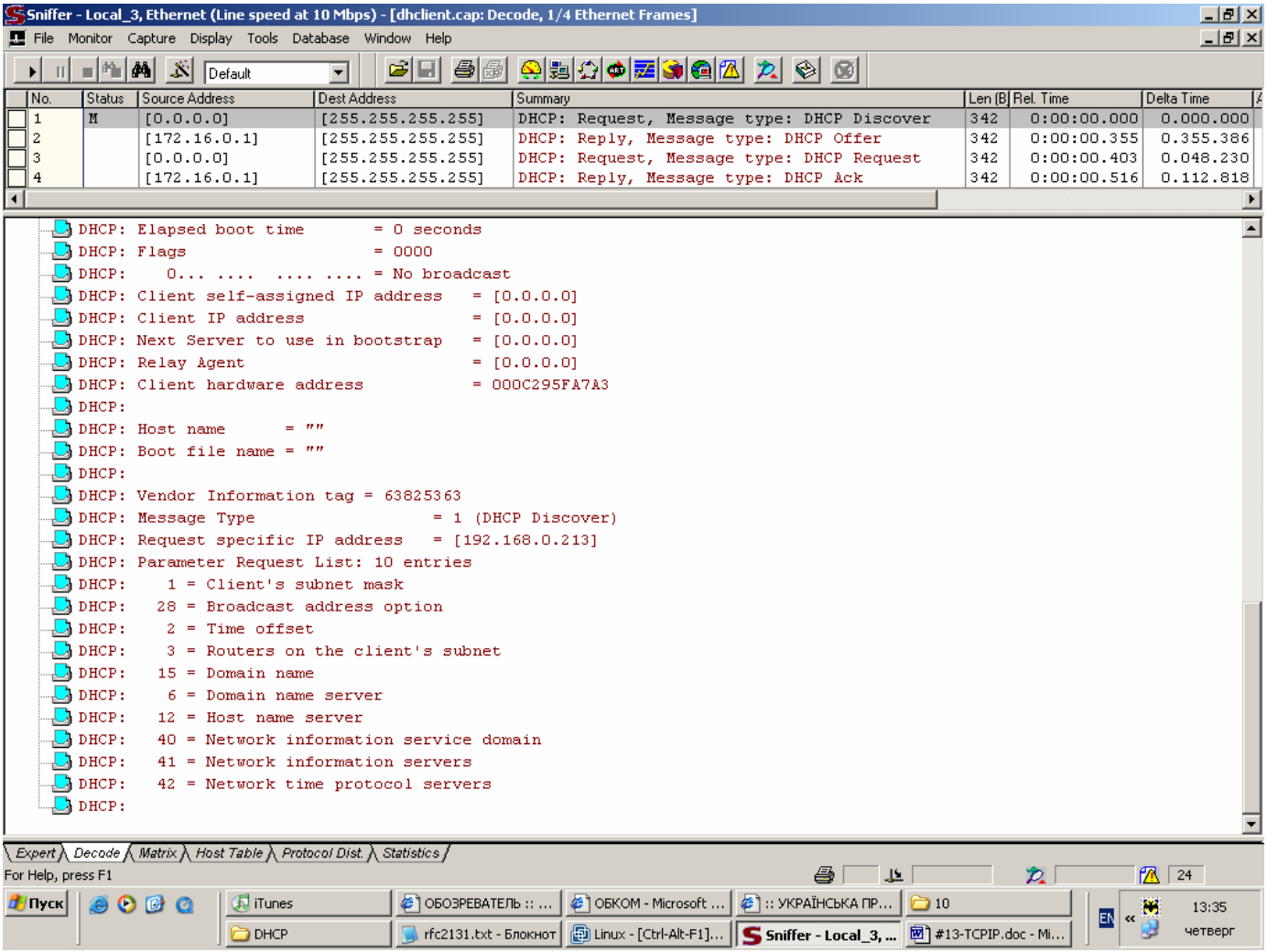

Видно, что в данном пакете сервер не предлагает клиенту никакого адреса, сообщает тип пакета и показывает свой IP адрес. Больше сервер ничего не посылает.

Наконец показываем особенности формирования пакета DHCPRELEASE. Открываем файл release.cap из прошлого урока:




Отмечаем, что клиент пока еще заполняет поле ciaddr, в поле опций указывает тип пакета, свой идентификатор и IP сервера, при этом пакет посылается направленно.

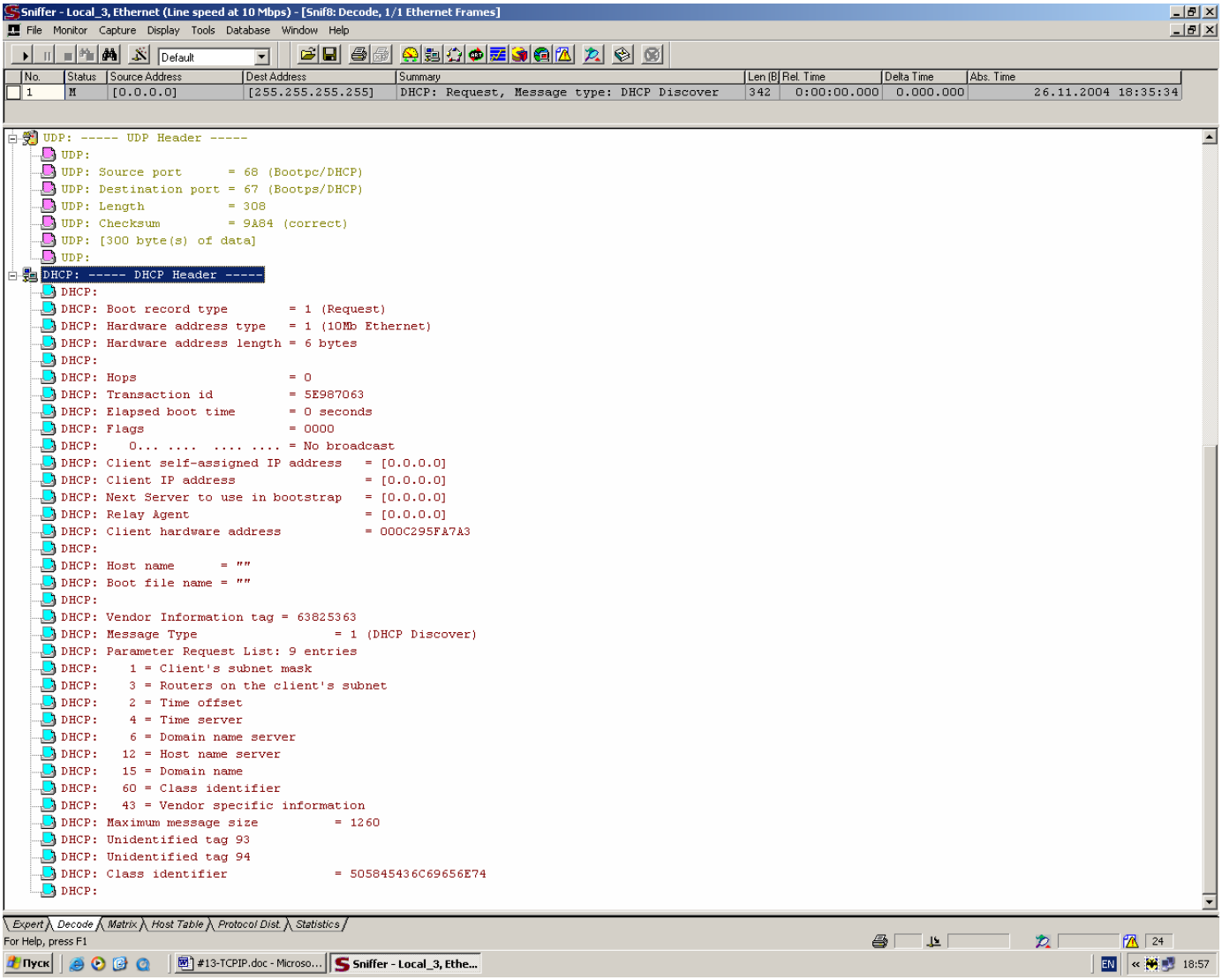
Проанализируем поведение клиента под управлением Microsoft Windows 2000 Pro. Как видим, данный клиент готов получать только ограниченное количество опций. Данный клиент не готов получать опции, конфигурирующие канальный уровень, TCP, большинство прикладных протоколов. Другие клиенты могут быть готовы получать от сервера больше опций, имеет смысл рассмотреть, как заполняют опцию Parameter Request List клиенты, встроенные в другие операционные системы. Для начала рассмотрим поведение DHCP клиента в операционной системе Linux. Рассмотрим клиента dhclient:




Видно, что Parameter Request List данного клиента достаточно бедный – клиент, например, вообще не может получать от DHCP сервера статических маршрутов.

Теперь рассмотрим поведение DHCP клиента, встроенного в Boot ROM сетевого адаптера (по сути, не имеет значения, о каком адаптере идет речь – главное – показать отличия).

Рассматриваем список опций в файле pxe.cap. Видно, что данный клиент тоже не умеет получать статических маршрутов, но с другой стороны от него это и не требуется, так как данный клиент не предназначен для обеспечения полноценного конфигурирования узла и предназначен лишь для того, чтобы обеспечить только начальный старт бездискового узла или узла без операционной системы с целью инсталляции операционной системы. При этом поддерживается две опции с кодами 93 и 94, специфичные именно для таких клиентов.




Итак, в сегодняшнем уроке мы рассмотрели заголовок пакета DHCP и опции, которые могут использоваться протоколом DHCP как для решения собственных задач, так и для конфигурирования клиентов, кроме того мы рассмотрели примеры, демонстрирующие функционирование полей заголовка DHCP и опций типа DHCP Extension. На следующем занятии мы рассмотрим конфигурирование DHCP сервера с точки зрения конфигурирования опций.