Network Working Group
Request for Comments: 951

Bill Croft (Stanford University)
John Gilmore (Sun Microsystems)
September 1985

Протокол загрузки BOOTP

BOOTSTRAP PROTOCOL (BOOTP)

1. Статус документа

Этот RFC предлагает протокол для сообщества ARPA-Internet и служит приглашением к дискуссии в целях развития протокола. Распространение этого документа не ограничивается.

2. Обзор

В этом RFC описан протокол загрузки (BOOTP) на основе IP/UDP, позволяющий бездисковым клиентским машинам определить свой адрес IP, адрес серверного хоста и имя файла для загрузки в память и выполнения. Операцию загрузки можно мысленно разделить на две фазы. Данный RFC описывает первую фазу, которую можно назвать «определением адреса и выбором загрузочного файла». После того, как адрес и имя файла получены, управление передаётся на вторую фазу загрузки, где выполняется копирование загрузочного файла. Для переноса файла обычно применяется протокол TFTP [9], поскольку предполагается размещение программ обеих фаз в памяти PROM на клиенте. Однако BOOTP может работать и с другими протоколами типа SFTP [3] или FTP [6].

Предполагается, что программы в PROM клиента обеспечивают способ выполнения загрузки без привлечения пользователя. Такая загрузка будет происходить при включении питания в необслуживаемом режиме. Пользователю следует обеспечивать механизм, позволяющий вручную задать требуемый адрес и имя файла для загрузки без применения протокола BOOTP и перехода непосредственно в фазу переноса загрузочного файла. Если доступна энергонезависимая память, предлагается записывать в неё принятые по умолчанию настройки и обходить протокол BOOTP, если эти настройки не приведут к отказу при загрузке файла. Если кэшированная информация не даёт нужного результата, протоколу загрузки следует возвращаться в фазу 1 и использовать BOOTP.

Ниже приведено краткое описание этапов протокола.

1. Выполняется один обмен пакетами. Используются повторы по тайм-ауту, пока не будет получен отклик. В обоих направлениях используется общая схема полей. Используются поля с фиксированным, максимальным разумным размером для более простого определения и анализа.
2. Поле opcode может содержать два значения. Клиент передаёт широковещательный пакет bootrequest, сервер отвечает пакетом bootreply. Пакет bootrequest включает аппаратный адрес клиента и IP-адрес, если он известен.
3. Запрос может включать имя сервера, от которого клиент хочет получить отклик. Это позволяет клиенту выполнить загрузку с определённого хоста (например, при наличии множества версий загрузочного файла или размещении сервера в удалённой сети или домене). Клиент не взаимодействует со службами имён и доменов, эта функция передана серверу BOOTP.
4. Запрос может включать «базовое» (generic) имя файла для загрузки (например, unix или ethertip). Когда сервер передаёт bootreply, он заменяет это поле полным именем подходящего для загрузки файла. При определении этого имени сервер может обращаться к своей базе данных для сопоставления адреса клиента и запрошенного имени с конкретным загрузочным файлом, настроенным для данного клиента. Если в bootrequest строка filename пуста (null), сервер возвращает поле filename, указывающее «принятый по умолчанию» файл для данного клиента.
5. Для случаев когда клиент не знает своего адреса IP сервер должен иметь базу данных, отображающую аппаратный адрес на адрес IP. Этот адрес IP для клиента помещается в поле bootreply.
6. Некоторые топологии сетей (например, в Стэнфорде) могут быть организованы так, что в данном кабельном сегменте нет своего сервера TFTP (например, все хосты и шлюзы кабельного сегмента могут быть бездисковыми). При взаимодействии с соседними шлюзами протокол BOOTP может обеспечить загрузку серверов, расположенных в нескольких интервалах пересылки (hop) от клиента. Это описано ниже в разделе 8. Загрузка через шлюзы. Эта часть протокола не требует каких-либо специальных действий на стороне клиента. Реализация такой возможности не обязательна и может потребовать некоторого добавления кода на шлюзах и серверах.

3. Формат пакетов

Все числовые значения приводятся в десятичном формате, если явно не указано иное. Пакет BOOTP помещается в стандартную дейтаграмму IP [8] UDP [7]. Для простоты предполагается, что пакеты BOOTP никогда не фрагментируются. Все числовые значения представляются в «стандартном сетевом порядке байтов», т. е. сначала указывается старший байт.

В заголовке IP пакета bootrequest клиент указывает свой адрес IP в поле отправителя, если он известен или 0, если адрес не известен. Когда не известен IP-адрес сервера, в поле получателя указывается широковещательный адрес 255.255.255.255. Этот адрес является широковещательным для кабельного сегмента (номер сети не известен) [4].

Заголовок UDP содержит номера портов отправителя и получателя. Протокол BOOTP использует два зарезервированных номера портов - BOOTP client (68) и BOOTP server (67). Клиент передаёт запросы, указывая BOOTP server в качестве порта получателя, обычно эти запросы являются широковещательными. Сервер передаёт отклики в порт BOOTP client - в зависимости от возможностей ядра и драйвера на сервере это может быть широковещательный или индивидуальный пакет (см. раздел 4. Курица или яйцо?). Использование **двух** зарезервированных портов обусловлено необходимостью избежать «пробуждения» серверных демонов BOOTP, когда

отклик bootreply должен передаваться клиенту в широковещательном режиме. Поскольку сервер и другие хосты не прослушивают порт BOOTP client, все входящие широковещательные сообщения будут фильтроваться на уровне ядра. Мы не можем позволить клиенту просто выбрать случайный порт в качестве источника UDP, поскольку отклик сервера может быть широковещательным и выбранный случайно порт может вызывать конфликты с другими хостами, которые могут прослушивать этот порт.

В поле размера UDP указывается размер UDP с учётом BOOTP. В поле контрольной суммы UDP клиент (или сервер) при желании может указать 0 для снижения издержек реализации PROM. В разделе 7. Обработка пакетов используется метка [UDP checksum] для указания мест, где контрольная сумма может рассчитываться или проверяться.

Поле	***Размер***	***Описание***
op	1	Код операции или тип сообщения - 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Тип аппаратного адреса. См. раздел ARP в RFC Assigned Numbers[1]. 1 = 10mb ethernet.
hlen	1	Размер аппаратного адреса (например, 6 для Ethernet 10 Мбит/с)
hops	1	Клиент устанавливает 0. Может использоваться шлюзами при загрузке через них.
xid	4	Идентификатор транзакции в форме случайного значения, используемого для сопоставления откликов с запросом.
secs	2	Заполняется клиентом и указывает число секунд с начала попыток загрузки клиента.
	2	Не используется.
ciaddr	4	IP-адрес клиента. Если адрес известен, он указывается клиентом в bootrequest.
yiaddr	4	«Ваш» (клиента) адрес IP. Указывается сервером, если клиент не знает своего адреса (ciaddr=0)
siaddr	4	IP-адрес сервера. Возвращается сервером в bootreply.
giaddr	4	IP-адрес шлюза, используемый при загрузке через шлюз.
chaddr	16	Аппаратный адрес клиента. Указывается клиентом.
sname	64	Необязательное имя серверного хоста. Строка с null-символом в конце.
file	128	Имя загрузочного файла (строка с null-символом в конце). Имя generic или пустая (null) в bootrequest, полный путь к файлу в bootreply.
vend	64	Необязательные фирменные данные производителя. Например, это может быть тип оборудования или его номер в запросе, capability или идентификатор удалённого файла в отклике. Эта информация может быть отложена до третьей фазы (загрузки ядра).

4. Курица или яйцо?

Как сервер может передать дейтаграмму IP клиенту, который (ещё) не знает своего адреса IP? Всякий раз при отправке bootreply передающая машина выполняет перечисленные ниже операции.

1. Если клиент знает свой адрес IP (поле ciaddr отлично от 0), пакет можно передать как обычно, поскольку клиент будет отвечать на запросы ARP [5].
2. Если клиент ещё не знает своего адреса IP (ciaddr = 0), он не может ответить на запросы ARP, переданные отправителем bootreply. Здесь возможны два варианта, описанных ниже.
   a. Если ядро или драйвер отправителя имеет доработки, требуемые для создания записи в кэше ARP «вручную», он может создать такую запись, используя значения полей chaddr и yiaddr. Эта запись должна иметь конечный срок существования (тайм-аут), как и обычные записи ARP. Отправитель bootreply может сейчас просто передать пакет bootreply по IP-адресу клиента. В UNIX (4.2 BSD) это поддерживается.
   b. Если отправитель не имеет требуемых доработок, он может просто передать bootreply по широковещательному адресу IP для соответствующего интерфейса. Это лишь одна дополнительная широковещательная передача по сравнению с предыдущим случаем.

5. Клиент, использующий ARP

Клиентский модуль PROM должен включать простую реализацию ARP, например, кэш адресов, содержащий единственную запись. Это позволит выполнить загрузку, включающую только фазу 2 (TFTP), если клиент знает свой адрес IP и имя файла для загрузки.

Клиенту, ожидающему получения отклика TFTP или BOOTP, следует быть готовым ответить на запрос ARP для отображения своего адреса IP на аппаратный адрес (если оно известно).

Поскольку bootreply будет включать (с аппаратной инкапсуляцией) аппаратный адрес отправителя (сервер или шлюз), клиент **может** быть в состоянии избежать отправки запроса ARP для IP-адреса сервера или шлюза, используемого в следующей фазе TFTP. Однако это следует считать особым случаем, поскольку желательно сохранить возможность загрузки с использованием лишь второй фазы, как описано выше.

6. Сравнение с RARP

Предложенный ранее протокол RARP[2] [1] был предназначен для обеспечения клиенту возможности определить свой адрес IP по известному аппаратному адресу. Однако протокол RARP имел недостаток, связанный с тем, что он работал на канальном уровне (а не IP/UDP). Это означает, что протокол RARP может быть реализован лишь на хостах со специально измененным ядром или драйвером, обеспечивающим доступ к необработанным (raw) пакетам. По причине наличия множества сетевых ядер, поддерживаемых разными организациями, протокол загрузки, не требующий изменения ядра получает явное преимущество.

BOOTP обеспечивает функцию поиска адреса IP по аппаратному адресу в дополнение к другим полезным функциям, описанным выше.

[1] В RFC 3232 документ Assigned Numbers был отменен, данные сейчас доступны по ссылке. *Прим. перев.*
[2] Reverse Address Resolution Protocol - протокол обратного преобразования адресов.

7. Обработка пакетов

7.1. Передача от клиента

Перед созданием пакет рекомендуется целиком очистить буфер, заполнив его нулями - это переведёт все поля в принятые по умолчанию состояния. Затем клиент создаёт пакет, заполняя описанные ниже поля.

В качестве IP-адреса получателя устанавливается значение 255.255.255.255 (широковещательный адрес) или IP-адрес сервера (если он известен). В полях IP-адреса отправителя и ciaddr устанавливается IP-клиента, если он известен, или 0 в противном случае. В заголовке UDP устанавливается подобающее поле размера, в качестве порта отправителя указывается BOOTP client, получателя - BOOTP server.

В поле op указывается 1 (BOOTREQUEST), в htype - тип аппаратного адреса из раздела ARP в RFC Assigned Numbers, а в hlen - размер аппаратного адреса (например, 6 для Ethernet 10 Мбит/с).

В поле xid помещается случайное значение идентификатора транзакции, secs указывает число секунд, прошедших с начала процесса загрузки клиента. Это значение позволит серверу узнать, как долго клиент предпринимает попытки. По мере роста этого значения некоторые серверы начинают проявлять «большую симпатию» к клиенту, который ещё не обслужен. Если у клиента нет подходящих часов, он может указать приблизительную оценку, используя циклический таймер. Возможна также установка в этом поле фиксированного значения (например, 100 секунд).

Если клиент знает свой адрес IP он указывает его в поле ciaddr и поле source address заголовка IP. А В поле chaddr указывается аппаратный адрес клиента.

Если клиент хочет загружаться с конкретного сервера, он может указать его имя (строка с null-символом в конце) в поле sname. Указанное имя должно быть разрешённым именем или псевдонимом желаемого хоста. Клиент имеет несколько вариантов заполнения поля file. Пустое поле говорит о том, что клиент хочет загрузить файл, принятый по умолчанию для его машины. Этот вариант может также говорить о том, что клиент лишь хочет узнать адреса IP (клиента, сервера, шлюза) и не думает об именах файлов.

В поле можно также указать «базовое» имя типа unix или gateway, что будет говорить о желании клиента загрузить указанную программу, настроенную для его машины. Наконец, это поле может указывать полный путь к загрузочному файлу.

В поле vend клиент может указать связанные с конкретным производителем строки или структуры. Например, это может быть тип оборудования или серийный номер устройства. Однако работа сервера BOOTP не должна **зависеть** от наличия такой информации.

При использовании поля vend рекомендуется помещать в его начало 4-байтовое значение magic number. Это позволит серверу определить тип информации в поле. Значения могут назначаться с использованием обычного процесса magic number - вы просто берете некое значение и оно становится «магическим» Для bootreply и bootrequest могут применяться разные значения, что позволит клиенту выполнять специальные действия с полученной в отклике информацией.

[UDP checksum]

7.2. Стратегия повторов для клиента

Если в течение некоторого времени не было получено ответа, клиенту следует повторить запрос. Интервал ожидания следует выбирать осторожно, чтобы не вызвать перегрузки сети. Предположим, что в кабельном сегменте имеется 100 машин, который разом включились после сбоя по питанию. Простой повтор запросов каждые 4 секунды приведёт к перегрузке сети.

В качестве варианта стратегии можно рассмотреть экспоненциальное увеличение интервала ожидания подобно тому, как это делается при конфликтах (коллизиях) в сети Ethernet. Примером может служит передача первого пакета в момент 0:00, второго - в :04, затем :08, :16, :32, :64. Следует также вносить случайную величину при выборе каждого интервала, подобно спецификации Ethernet, начиная с некой «маски» к которой применяется операция «логическое И» со случайным значением для первого повтора. При каждом следующем повторе размер маски увеличивается на 1 бит. Это будет приводить к удвоению средней задержки при каждом следующем повторе.

После того, как «средний» интервал повтора достигнет 60, дальнейшее увеличение не имеет смысла и вносятся лишь случайные изменения интервала.

Перед каждым повтором клиенту следует обновлять значение поля secs. [UDP checksum]

7.3. Получение сервером сообщения BOOTREQUEST

[UDP checksum] Если в качестве порта получателя UDP указано не BOOTP server, пакет отбрасывается.

Если имя сервера (sname) пусто (не указан конкретный сервер) или указанное имя соответствует реальному имени или псевдониму сервера, обработка пакета продолжается.

Если заданное имя не соответствует имени получившего пакет сервера, возможно несколько вариантов.

1. Можно просто отбросить пакет.
2. Если поиск указанного в sname имени показывает, что оно находится в том же кабельном сегменте, пакет отбрасывается.
3. Если имя sname относится к другой сети, можно переслать пакет по соответствующему адресу. Для этого проверяется значение поля giaddr (адрес шлюза). Если giaddr = 0, в него помещается адрес данного сервера или шлюза, который ведёт в нужную сеть. После этого пакет пересылается.

Если IP-адрес клиента (ciaddr) имеет значение 0, клиент не знает своего адреса IP. Предпринимается попытка найти аппаратный адрес клиента (chaddr, hlen, htype) в базе данных сервера. Если адреса не найдено, пакет отбрасывается. В противном случае сервер знает IP-адрес клиента и указывает его в поле yiaddr (ваш адрес IP).

Далее проверяется имя загрузочного файла (file). Поле будет пустым (null), если клиенту не нужен загрузочный файл или он хочет использовать принятый по умолчанию файл. Если поле не пусто, его значение используется для поиска в базе данных вместе с IP-адресом клиента. Если это используемый по умолчанию или базовый файл (возможно индексированный по адресу клиента) или полностью заданный путь, соответствующий реальному файлу, значение поля file заменяется полным путём к выбранному загрузочному файлу. Если поле не пусто, но соответствия не найдено, это говорит о том, что клиент запрашивает отсутствующий файл (возможно он имеется на другом сервере BOOTP) и пакет отбрасывается.

Далее нужно проверить поле vend, содержащие определяемые производителем данные и при наличии в нем распознаваемой информации следует предпринять заданные клиентом действия, а результат поместить в поле vend пакета с откликом. Например, рабочая станция-клиент может предоставить ключ аутентификации и получить от сервера возможность доступа к удалённому файлу или предоставить набор параметров конфигурации, которые могут быть переданы операционной системе, которая вскоре будет загружена.

IP-адрес сервера помещается в поле siaddr, а в поле op указывается BOOTREPLY. Для порта получателя UDP задаётся значение BOOTP client. Если адрес клиента в ciaddr отличается от 0, пакет передаётся по этому адресу. В противном случае, если адрес шлюза giaddr отличается от 0, устанавливается порт получателя UDP BOOTP server и пакет передаётся по адресу giaddr. В остальных случаях клиент находится в одном кабельном сегменте с сервером, но ещё не знает своего адреса IP, поэтому используется метод, описанный в разделе 4. Курица или яйцо? Если этот метод применяется при наличии на передающем хосте множества интерфейсов, используется поле yiaddr (IP-адрес клиента) при выборе сети (интерфейса) для передачи пакета. [UDP checksum]

7.4. Получение сервером или шлюзом сообщения BOOTREPLY

[UDP checksum] Если yiaddr (IP-адрес клиента) указывает на один из подключённых кабельных сегментов, используется один из методов разделе 4 для пересылки пакета клиенту. Следует убедиться, что поле порта получателя указано значение UDP BOOTP client.

7.5. Приём сообщения клиентом

Не забыть обработать запросы ARP для своего адреса IP (если он известен). [UDP checksum] Клиенту следует отбрасывать входящие пакеты IP/UDP, адресованные в другой (не BOOTP client) порт, не являющиеся BOOTREPLY, не соответствующие IP-адресу клиента (если он известен) или с несоответствующим идентификатором транзакции. В остальных случаях пакет содержит нужный отклик. В поле yiaddr будет указан IP-адрес клиента, который мог быть не известен до этого. В поле file указано имя файла для запроса TFTP read. Адрес сервера указан в поле siaddr. Если поле giaddr (адрес шлюза) отлично от 0, пакеты следует пересылать по этому адресу, чтобы они попали на сервер.

8. Загрузка через шлюзы

Эта часть протокола является не обязательной и требует некого дополнительного кода на серверах и шлюзах, но позволяет организовать загрузку через шлюз. Это полезно прежде всего в ситуациях, когда в качестве шлюзов используются бездисковые машины. Шлюзы с диском (например, UNIX-машина) могут поддерживать свои серверы BOOTP/TFTP.

Шлюз, прослушивающий широковещательные запросы BOOTREQUEST, может принять решение о пересылке (возможно широковещательной) таких пакетов при необходимости. Например, конфигурация шлюза может включать список других сетей или хостов для получения копий широковещательных запросов BOOTREQUEST. Даже при наличии поля hops, дурным тоном будет глобальная широковещательная пересылка запросов, поскольку это почти всегда приводит к «петле широковещания».

Пересылка может происходить сразу же или по достижении полем secs (число секунд с начала попыток загрузки) некого порога.

Если шлюз решает переслать запрос, ему следует проверить поле giaddr (IP-адрес шлюза). При нулевом значении поля в него следует поместить свой адрес IP (в сторону принявшего пакет кабельного сегмента). Может также использоваться поле hops для управления дальнейшей пересылкой. Значение этого поля увеличивается при каждой пересылке. Например, если поле имеет значение 3, следует подумать об отбрасывании пакета.

[UDP checksum]

Здесь мы рекомендуем размещать эту специальную функцию пересылки на шлюзах. Однако это не всегда нужно. Пока в одной сети с загружаемым клиентом есть какой-либо агент пересылки BOOTP, этот агент может переслать пакеты при необходимости. Таким образом, эти услуги могут быть совмещены со шлюзом или реализованы отдельно.

Если агент пересылки не совмещён со шлюзом, этот агент может сэкономить свою работу, помещая широковещательный адрес принявшего bootrequest интерфейса в поле giaddr. В результате отклик будет пересылаться с использованием обычных шлюзов без участия агента. Недостатком такого подхода является невозможность использовать методы без широковещания, описанные в разделе 4, для пересылки откликов, что увеличивает нагрузку на каждый хост в кабельном сегменте клиента.

9. Пример базы данных сервера BOOTP

В качестве предложения здесь показан образец текстового файла базы данных, который может использовать программа сервера BOOTP. База данных имеет два раздела, между которыми размещается строка, начинающаяся символом процента в позиции 1. Первый раздел содержит принятый по умолчанию каталог и отображения базовых имён на каталог/путь к файлу. Первым базовым именем в этом разделе служит используемый по умолчанию файл, который возвращается, если bootrequest содержит пустую (null) строку file.

Вторая часть отображает типы аппаратных адресов и сами адреса (addresstype/address) на адреса IP (ipaddress). Можно также переписать используемое по умолчанию базовое имя, представив относящееся к ipaddress имя genericname. Элемент suffix также является необязательным - если он указан, любые базовые имена, указанные клиентом, будут доступны при добавлении suffix после pathname, подходящего для данного базового имени. Если файл

не найден, будет предпринята попытка использовать pathname. Эта опция suffix позволяет организовать целый набор базовых имён без особых усилий. Ниже показан общий формат, поля разделены одним или несколькими символами пробелов или табуляции, пустые поля в конце могут быть опущены, пустые строки и строки, начинающиеся с символа # будут игнорироваться.

```
# комментарий

homedirectory
genericname1    pathname1
genericname2    pathname2
...

% конец базовых имён, начало отображения адресов

hostname1 hardwaretype hardwareaddr1 ipaddr1 genericname suffix
hostname2 hardwaretype hardwareaddr2 ipaddr2 genericname suffix
...
```

Ниже приведён конкретный пример. Отметим, что значение hardwaretype совпадает с указанным в разделе ARP документа Assigned Numbers. Значения hardwaretype и ipaddr указаны в десятичном формате, hardwareaddr - в шестнадцатеричном.

```
# последнее обновление от smith

/usr/boot
vmunix          vmunix
tip             ethertip
watch           /usr/diag/etherwatch
gate            gate.

% конец базовых имён, начало отображения адресов

hamilton        1 02.60.8c.06.34.98     36.19.0.5
burr            1 02.60.8c.34.11.78     36.44.0.12
101-gateway     1 02.60.8c.23.ab.35     36.44.0.32      gate 101
mjh-gateway     1 02.60.8c.12.32.bc     36.42.0.64      gate mjh
welch-tipa      1 02.60.8c.22.65.32     36.47.0.14      tip
welch-tipb      1 02.60.8c.12.15.c8     36.46.0.12      tip
```

Если в приведённом выше примере mjh-gateway выполняет загрузку по умолчанию, он получит файл /usr/boot/gate.mjh.

10. Благодарности

Спасибо Ross Finlayson и другим авторам RFC, описывающих загрузку по протоколу TFTP [2] с помощью RARP [1].

Спасибо за предшествующую работу и комментарии Noel Chiappa, Bob Lyon, Jeff Mogul, Mark Lewis и David Plummer.

Литература

1. Ross Finlayson, Timothy Mann, Jeffrey Mogul, Marvin Theimer. A Reverse Address Resolution Protocol. RFC 903, NIC, June, 1984.
2. Ross Finlayson. Bootstrap Loading using TFTP. RFC 906, NIC, June, 1984.
3. Mark Lottor. Simple File Transfer Protocol. RFC 913, NIC, September, 1984.
4. Jeffrey Mogul. Broadcasting Internet Packets. RFC 919, NIC, October, 1984.
5. David Plummer. An Ethernet Address Resolution Protocol. RFC 826, NIC, September, 1982.
6. Jon Postel. File Transfer Protocol. RFC 765[1], NIC, June, 1980.
7. Jon Postel. User Datagram Protocol. RFC 768, NIC, August, 1980.
8. Jon Postel. Internet Protocol. RFC 791, NIC, September, 1981.
9. K. R. Sollins, Noel Chiappa. The TFTP Protocol. RFC 783[2], NIC, June, 1981.

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Этот документ был заменён RFC 959. *Прим. перев.*
[2]Этот документ был заменён RFC 1350. *Прим. перев.*