УДК 004.4; 004.6; 004.7

Способ управления трафиком в BitTorrent-сетях с помощью протокола DHT

В.Е. Рабинович, А.А. Шестаков

Протокол BitTorrent занимает лидирующее место среди протоколов передачи файлов. Основная цель данного протокола – раздать файл как можно большему количеству участников таким образом, чтобы во время распространения доступность файла не уменьшалась. При создании этот протокол рассматривался как централизованный, то есть имел центральный узел (трекер), который хранит и распространяет список участников раздачи. Будучи владельцем трекера, можно управлять распределением трафика внутри BitTorrent-сети, всего лишь изменяя список получателей.

Со временем встал вопрос о децентрализации протокола. Этому способствовали несколько факторов: неспособность трекеров обслуживать огромное количество участников, большое количество независимых трекеров, претензии со стороны правообладателей за распространение контента, защищённого авторскими правами. Решением стало появление дополнения к протоколу BitTorrent [5], описывающее распространение списка участников посредством протокола DHT. В результате исчезла возможность влиять на распределение трафика или управлять списком участников.

В данной статье предложен способ, с помощью которого можно влиять на трафик сети с протоколом BitTorrent и, пусть не в полной мере, а лишь частично, управлять процессом распространения файла. Это может быть полезно как интернет-провайдерам для перераспределения потоков данных внутри своей сети, так и правообладателям для ограничения распространения файла.

Ключевые слова: вычислительная система, вычислительная сеть, распределённые системы, сервер, клиент, протокол, HTTP, BitTorrent, трекер, DHT.

1. Введение

BitTorrent – это протокол распространения файлов. В протоколе реализована схема взаимодействия в сети, отличная от традиционной модели клиент – сервер. Основная идея протокола BitTorrent – способствовать распространению популярных файлов, не нагружая источник излишним трафиком и избавляя от возможных отказов в случае превышения возможностей сервера. BitTorrent сети являются самомасштабируемыми в том смысле, что скорость передачи файлов потребителям возрастает с их ростом. Такой эффект достигается за счёт того, что участник (узел сети), скачивающий файл, сам участвует в его распространении, причём таким образом, что части файлов, имеющих самую низкую доступность, распространяются в первую очередь.

BitTorrent является одним из самых распространённых протоколов для передачи больших файлов и, по оценкам, он занимает от 43% до 70% всего интернет трафика (в зависимости от географического положения) по данным на февраль 2009 [8].

BitTorrent-приложения являются большой проблемой для сетей интернет-провайдеров. Так как провайдер обычно выкупает канал связи у магистрального провайдера верхнего уровня, то трафик между провайдером и внешним миром – достаточно дорог. К сожалению, спецификация протокола BitTorrent не подразумевает выбор участников с учётом топологии

сети и пропускной способности каналов связи провайдера и инициирует передачу данных между случайно выбранными участниками по всему миру. Если в случае использования централизованной передачи посредством трекера провайдер может влиять на распространение файла [7], то при использовании DHT-протокола такой возможности нет.

Ни для кого не секрет, что «пиратство» на сегодняшний день является большой проблемой для правообладателей. Огромное количество контента, защищённого авторским правом, распространяется посредством BitTorrent-сетей благодаря «пиратам». Но если в случае использования централизованной передачи правообладателям достаточно лишь закрыть раздачу на трекере (центральном узле), то в случае использования протокола DHT это невозможно; мало того, невозможно даже определить инициатора раздачи этого файла.

Одним из решений перечисленных проблем может быть непосредственное влияние на список участников путём внедрения собственных узлов в DHT-сеть. В данной статье предложен способ, с помощью которого можно влиять на трафик сети с протоколом BitTorrent и управлять процессом распространения файла.

2. Протокол BitTorrent

Особенности протокола BitTorrent

BitTorrent – пиринговый (одноранговый, децентрализованный) протокол передачи данных, разработанный для эффективного распространения файлов большому количеству пользователей. Основная идея заключается в использовании исходящих ресурсов (полосы пропускания) всех участников, скачивающих файл.

Для файла, подлежащего распространению, необходимо создать торрент-файл (файл с метаданными, который содержит информацию о файле, контрольные суммы и адрес трекера, обслуживающего файл) и запустить для него трекер. Трекер – это приложение, которое хранит и распространяет информацию обо всех участниках, скачивающих файл. Клиент BitTorrent, скачивающий файл, соединяется с трекером и получает информацию обо всех участниках, скачивающих этот файл [3].

Для передачи данных между участниками используется всего 5 активных соединений. Первоначально все соединения между участниками заблокированы. Каждые 10 секунд клиент оценивает, каким участника он будет отправлять данные. Четыре из них выбираются на основе алгоритма «око за око». Его суть заключается в выборе таких участников, которые смогут передать ему данные с максимальной скоростью. Последнее, пятое соединение, выбирается каждые 30 секунд на основе алгоритма «оптимистической разблокировки», суть которого заключается в случайном выборе из всех остальных участников, то есть делается оптимистичный прогноз о скорости передачи выбранного участника. Она позволяет новым участникам начать передачу, а также найти других участников, которые могут иметь лучшую полосу пропускания. При этом благодаря алгоритму «око за око» трафик смещается в сторону наиболее быстрых участников. Таким образом поддерживается ограничение в 5 конкурентных передач. Данная цифра описана в спецификации протокола, но может отличаться в конкретных реализациях клиента. В любом случае выбор активных соединений ограничивается первоначальным списком соседей, полученным на этапе запроса к трекеру.

Участники обмениваются списком блоков файла каждый раз, когда им необходимо скачать новые блоки. В результате этих обменов все участники имеют актуальную информацию о данных, имеющихся у остальных. После проведения разблокировки, клиент запрашивает у разблокированного участника информацию об имеющихся у него блоках и выбирает блок для скачивания, используя принцип «наименее распространённого», то есть выбирает блок, который есть у наименьшего числа участников. Заметим, что участник знает только о блоках файла, распространённых среди участников, которые получил от трекера.

Трекер

BitTorrent-трекер — веб-сервер, осуществляющий координацию участников BitTorrent-сети. Основная функция BitTorrent-трекера — обработка запросов клиентов. Участник, заинтересованный в скачивании файла, должен в первую очередь соединиться с трекером (создать анонс-запрос), обслуживающим этот файл, чтобы присоединиться к BitTorrent-сети этого файла. Каждый участник подключается к трекеру и в ответ получает *C* (по умолчанию 35) случайных участников из всех, подключившихся к трекеру. Если файл обслуживается более чем одним трекером, клиент перемешивает полученный список и выбирает из них определённое количество. Затем участник инициирует соединения со всем остальными и начинает скачивание. Естественно, участники могут покидать сеть в любое время; таким образом, если количество активных участников снизилось до определённого значения (по умолчанию 20), то необходимо снова соединиться с трекером, чтобы получить новый список.

Трекер лишь «связывает» участников друг с другом, но напрямую не участвует в обмене файлов. Более того, трекер не имеет никакой информации о содержимом этих файлов, поскольку клиенты сообщают ему лишь сервисную информацию, такую как хеш-сумма файла, размер и его имя.

Трекер считается «слабым» местом системы BitTorrent, поскольку при его отключении новые клиенты просто не могут «найти» друг друга. При этом уже участвующие в раздаче клиенты могут некоторое время продолжать обмен, постепенно теряя тех, кто отключился или у кого поменялся IP-адрес.

3. Протокол DHT

Особенности протокола DHT

DHT (Distributed Hash Table — «распределённая хеш-таблица») — это класс децентрализованных распределённых систем, которые обеспечивают поисковый сервис, похожий по принципу работы на таблицу хешей, и имеют структуру ассоциативного массива. Каждый узел DHT-сети может рационально искать значение, ассоциированное с заданным именем. Ответственность за поддержку связи между именем и значением распределяется между узлами таким образом, что изменение набора узлов не ведёт к потере информации. Это позволяет легко масштабировать DHT и постоянно отслеживать добавление/удаление узлов и ошибки в их работе.

Сеть DHT — это инфраструктура, которая может быть использована для построения многих комплексных сервисов. Например: распределённые файловые системы, пиринговое распространение файлов и системы распространения контента, кооперативный web-кэш, многоадресная доставка (multicast), доставка к ближайшим узлам (anycast), сервис доменных имен и система мгновенных сообщений.

Изыскания в области DHT изначально были мотивированы, в частности, пиринговыми системами, такими как I2P, Napster, Gnutella, Freenet, которые использовали распределённые в интернете ресурсы для создания одного единственного приложения.

DHT-сеть характеризуется следующими свойствами:

- децентрализация: связь между узлами поддерживается без единого координационного узла;
- масштабируемость: система одинаково эффективно функционирует при тысячах или миллионах узлов;
- отказоустойчивость: система одинаково надёжна с узлами постоянно подключающимися, отключающимися и выдающими ошибки.

Ключевая методика создания DHT-сети заключается в том, что любой узел должен скоординироваться только с несколькими узлами в системе (как правило, их число $O(\log n)$, где n — общее количество участников). При этом при изменении числа участников сети каждый узел должен проделать ограниченный объём работы для актуализации таблиц маршрутизации внутри сети.

Структура DHT-сети может быть разбита на несколько основных компонентов [6]:

— в качестве основы используется **пространство ключей**, такое как, например, набор 160-битных строк (количество бит может варьироваться);

— **схема разбиения пространства ключей** распределяет ключи среди участвующих узлов;

— **транспортная сеть**, обеспечивающая соединение между узлами и позволяющая найти владельца любого ключа из пространства ключей.

Типичное использование протокола DHT для хранения и выдачи информации происходит следующим образом. Чтобы сохранить файл с данным именем и информацией в DHT-сети, узлом определяется хеш-сумма от имени файла по алгоритму SHA1, из которого формируется 160-битный ключ k, после чего формируется сообщение и посылается любому участвующему узлу в сети DHT. Послание идёт от одного узла к другому через транспортную сеть до тех пор, пока оно не достигнет единственного узла, ответственного за ключ k, в соответствии со схемой разбиения, где и будет храниться пара ключ – значение. Любой другой клиент может получить содержимое файла, сформировав ключ тем же алгоритмом из имени требуемого файла и отправив в сеть сообщение с запросом данных, связанных с этим ключом. Сообщение снова пройдёт через транспортную сеть к узлу, ответственному за ключ, который отправит ответ с затребованными данными.

Различные реализации DHT-протокола используют разные варианты постоянного хеширования для отображения ключей в узлы. Этот подход включает в себя функцию, которая определяет абстрактное понятие расстояния между ключами, которое никак не зависит от географического расстояния и топологии транспортной сети. Каждый узел представляет собой единичный ключ, называемый идентификатором (ID), и узел с ID j владеет всеми ключами, для которых j – самый ближайший к ID.

Совместное использование протоколов DHT и BitTorrent

DHT в BitTorrent фактически выполняют основную функцию BitTorrent-трекера — помогают участникам обмена узнать друг о друге. Основными функциями DHT являются:

- помощь участникам в поиске друг друга;
- снижение нагрузки на трекер;
- поддержка обмена в периоды недоступности трекера;
- обмен без участия трекера.

Реализация распределённой сети в BitTorrent-клиентах основана на варианте DHT, называемом Kademlia, с использованием протокола UDP. Клиенты BitTorrent «слушают» тот же номер порта UDP, который они используют для входящих TCP-соединений. Каждый подключённый клиент является в сети DHT отдельным узлом. У него есть свой уникальный ID (идентификатор), случайно выбираемый из того же 160-битного пространства, что и хеш-функции торрент-файлов.

Каждый узел хранит таблицу маршрутизации, содержащую контактную информацию о многих «ближайших» к нему узлах и о нескольких более далёких. «Близость» двух узлов вычисляется из «сходства» их ID и не имеет никакого отношения к их географической близости.

4. Механизм работы протокола DHT в BitTorrent-сетях

Идентификатор узла и таблица маршрутизации

При подключении узла к сети DHT формируется его уникальный идентификатор (ID). Он представляет из себя 160-битную строку, так же как и хеш файла, и выбирается случайным образом. Для определения схожести ID с хешем или с другим ID вычисляется расстояние между ними. Каждый узел имеет информацию об узлах, ID которых наиболее схожи с его собственным, при этом чем больше расстояние, тем меньше таких узлов хранится в таблице маршрутизации.

В Kademlia расстояние рассчитывается как операция XOR между ключами (идентификаторы узлов или хеш-суммы файлов) и результат операции преобразуется в беззнаковое целое. D(A,B) = |A xor B|. Чем меньше полученное значение D, тем меньше расстояние между этими двумя ключами.

Каждый узел обслуживает свою таблицу маршрутизации. Узлы в таблице используются как отправная точка для поиска в DHT. В таблицу маршрутизации также попадают узлы, информация о которых была получена от других узлов. Стоит заметить, что не все узлы равны в таблице маршрутизации: их можно разделить на «хорошие» и «плохие». Множество узлов, участвующих в DHT сети, могут как отправлять запросы, так и получать их, но не могут отвечать на запросы. Важно, чтобы таблица маршрутизации каждого узла состояла только из «хороших» узлов. «Хорошим» узел является, если ответил хотя бы на один запрос за последние 15 минут. Также «хорошим» узел считается, если он был получен в качестве ответа от другого «хорошего» узла и отправил хотя бы один запрос за последние 15 минут. Узел становится «плохим», если не ответил на один запрос. Узлы, которые считаются «хорошими», имеют более высокий приоритет по сравнению с теми, о которых ничего не известно.

Механизм поиска участников

Когда узел ищет участников для работы с файлом, он сравнивает расстояние между хешем файла и списком ID из своей таблицы маршрутизации. Затем он связывается с известными ему узлами, чей ID наиболее близок к хешу и запрашивает у них информацию об участниках, скачивающих этот файл. Если узел имеет в своей таблице маршрутизации информацию о таких участниках, то в качестве ответа он отправляет необходимую информацию о них. В противном случае узел должен отправить в ответ на запрос информацию об узлах из своей таблицы маршрутизации, чьи ID наиболее близки к запрашиваемому хешу. Запрашивающий узел итеративно отправляет запросы к наиболее близким узлам до тех пор, пока в ответ не получит узлы ещё ближе. После окончания поиска клиент добавляет в свою таблицу маршрутизации информацию о полученных узлах, ID которых наиболее близки к искомому ключу.

Кроме того, информация, полученная узлом на запрос участников, содержит в себе некоторый опознавательный знак (token). В протоколе BitTorrent опознавательный знак является SHA1 хешем от IP-адреса, соединенного с секретным словом, которое меняется каждые пять минут. Для того чтобы сообщить другому участнику, что узел имеет запрашиваемый файл, он должен вместе с остальной информацией отправить и этот опознавательный знак. Это делается для того, чтобы определить, пришёл запрос от известного нам участника или нет; из этого следует, что в BitTorrent-сеть передаются лишь информация об авторизованных таким образом участниках. Кроме того, сети DHT и BitTorrent не связаны между собой напрямую, то есть, общаясь с узлами через DHT, мы лишь получаем список участников для обмена в BitTorrent-сети; в тоже время при обмене в BitTorrent-сети DHT не получает никакой информации о передаваемых файлах. При получении BitTorrent-клиентом информации об участнике обмена проверяется лишь опознавательный знак, а не IP-адрес. Сделано это для защиты BitTorrent-сети от неавторизованных участников, которые могут внедриться в сеть обмена. Опознавательный знак действителен ограниченное время, и в текущей спецификации DHT-протокола, используемого в BitTorrent, это время составляет до 10 минут. Таким образом, можно сказать, что авторизация участников происходит каждые 10 минут.

5. Управление трафиком в DHT-сетях

Внедрение агентских узлов

Как было описано ранее, владельцы трекеров и в некоторых случаях интернет-провайдеры, могут влиять на перераспределение трафика в BitTorrent-сети, используя ретрекер, пристрастный выбор участников и другие способы, но при использовании DHT данные решения не приносят результатов. Но если обратить внимание на алгоритм поиска участни-

ков в DHT, то появляется теоретическая возможность влияния на список участников, распространяемый между узлами.

Поскольку поиск участников производится по ID, который соответствует хеш-сумме файла, то становится очевидным, что список участников, который будет использоваться при обмене, будет получен от узлов с ID, наиболее близким к хеш-сумме файла. Как было описано выше, список участников, полученный от «хороших» узлов, имеет больший приоритет; следовательно, чем больше будет таких узлов, тем большее вероятность использования полученных от них участников при обмене в BitTorrent-сети.

Идея состоит в том, чтобы внедрить в сеть DHT собственные узлы, назовем их агентами, которые будут, отвечая на DHT запросы, имитировать поведение «хорошего» узла и сообщать вместо реальных адресов – IP-адреса участников, обмен между которыми приоритетнее.

Представим себе сеть BitTorrent одного файла, в котором есть около 1000 участников, 100 из которых находятся в одной физической сети; и для провайдера, обслуживающего этих клиентов, выгоднее было бы перевести трафик в эту сеть. При обычном развитии ситуации в устоявшемся режиме (когда поиск участников завершён и прошло несколько циклов поиска) трафик будет распределён абсолютно случайным образом и разбросан по всей BitTorrent-сети. Внедряя агентов, которые на все запросы из одной сети будут отвечать узлами из этой же сети, участники будут вынуждены обмениваться данными друг с другом. При этом остальные 900 участников, которые находятся вне сети провайдера, должны получить список узлов, как в случае обычного обмена без внедрения агентов. Таким образом, трафик 100 участников переместится в определённый сегмент сети. Естественно, полностью подавить трафик извне невозможно, но это и к лучшему, ведь пока файл полностью не получен одним из локальных участников, остальные также не смогут получить его целиком.

Основной вопрос, как заставить участников обращаться к агентам. Согласно спецификации, запросы участников сходятся к узлам, чьи ID окажутся наиболее близкими к хешу файла. Следовательно, ID агентов должны быть выбраны не случайным образом, а специально сформированы так, чтобы расстояние между ID и хешем было минимально.

Очевидно, что количество агентов должно быть достаточным, чтобы оказывать влияние на сеть в целом. При этом их ID не должны совпадать. Без проведения экспериментов невозможно точно сказать, какого количества агентов будет достаточно для заданных целей, но ясно, что это значение не фиксировано и зависит от общего количества участников в сети. Кроме того, не известно, насколько близкие ключи следует выбрать для идентификаторов внедрённых агентов. При большом расстоянии запросы к агентам будут редки и не окажут влияния на распределение участников, а при расстоянии, равном 1, все запросы сведутся к агентам. Данные значения можно подобрать лишь экспериментальным путём.

Известные проблемы

Как было сказано выше, количество агентов должно быть достаточно большим, поэтому при большом количестве обслуживаемых файлов будет создаваться большая нагрузка на оборудование. Таким образом, мы можем внедрять агентов для обслуживания ограниченного количества файлов.

Поскольку для внедрения агентов необходим хеш файла, то даже при неограниченных ресурсах оборудования мы не можем обслуживать всю BitTorrent-сеть. Кроме того, чтобы внедрить агентов, необходимо определить, какими файлами в данный момент времени обмениваются внутри BitTorrent-сети. Решением этой проблемы может быть совмещение данного способа и ретрекера. С помощью ретрекера можно получать информацию о файлах, которыми обмениваются в данный момент, и какие из них являются наиболее популярными, чтобы затем передать хеши файлов агентам. Недостатком этого решения может быть то, что ретрекер получает не все запросы от участников и мы не сможем построить полной картины популярности файлов в BitTorrent-сети.

6. Заключение

Разработанный способ требует реализации и проведения экспериментов, но уже сейчас можно сказать, что его эффективность будет высокой. Кроме того, на сегодняшний день нет каких-либо других методов влияния на распределение трафика в BitTorrent-сети, основанной на DHT.

Доля BitTorrent-трафика в сетях провайдеров очень высока и растёт. Поскольку трафик внутри сети провайдера дешевле, чем внешний трафик, то каждый провайдер стремится сместить передачу внутрь сети. Данный способ поможет им в какой-то мере решить эту проблему.

Очевидно, что наряду с распространением реальных адресов участников обмена, можно распространять и адреса несуществующих участников или заведомо ложных. Таким образом, мы можем и снизить обмен в сети до минимума. С точки зрения пользователей это, конечно же, плохо, но для правообладателей, чей контент распространяется нелегально, это может быть отличной возможностью для ограничения распространения «пиратского» контента.

Дальнейшая работа предполагает реализацию DHT-клиента в соответствии с разработанными требованиями и экспериментальное внедрение в сети крупного провайдера.

Литература

1. *Bindal R., Cao P. and Chan W.* Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. [Электронный ресурс]. – Режим доступа: `http://crypto.stanford.edu/~cao/biased-bt.pdf`, свободный.
2. *Cohen, B. (2003).* Incetives build robustness in Bittorrent. [Электронный ресурс]. – Режим доступа: `http://www.bittorrent.org/bittorrentecon.pdf`, свободный.
3. *Cohen, B. (2008).* The BitTorrent Protocol Specification. [Электронный ресурс]. – Режим доступа: `http://www.bittorrent.org/beps/bep_0003.html`, свободный.
4. *Hoffman J.* BEP 12: Multitracker Metadata Extension. [Электронный ресурс]. – Режим доступа: `http://bittorrent.org/beps/bep_0012.html`, свободный.
5. *Loewenstern A. (2008).* DHT Protocol. [Электронный ресурс]. – Режим доступа: `http://www.bittorrent.org/beps/bep_0005.html`, свободный.
6. *Moni Naor and Udi Wieder.* Novel Architectures for P2P Applications: the Continuous-Discrete Approach. Proc. SPAA, 2003. [Электронный ресурс]. – Режим доступа: `http://research.microsoft.com/pubs/73859/dhpaper_final_hp.pdf`, свободный.
7. *Rabinovich E.V., Shestakov A.A.* Traffic localization in BitTorrent Network via Retracker // Proceedings of RFBR and DST Sponsored «The 2-nd Russian – Indian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics», 10 – 13 September. – 2011. - Additional volume, P. 50–55.
8. *Schulze H., Mochalski K. (2009).* Internet Study 2008/2009. Ipoque. [Электронный ресурс]. – Режим доступа: `http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf`, свободный.

Статья поступила в редакцию 19.04.2012;
переработанный вариант — 24.09.2012

Рабинович Евгений Владимирович
доктор технических наук, профессор кафедры вычислительной техники Новосибирского государственного технического университета.
e-mail: evr@vt.cs.nstu.ru

Шестаков Алексей Александрович
аспирант кафедры вычислительной техники Новосибирского государственного технического университета.
e-mail: shestakov.alexey@gmail.com

Traffic Control Mode in Bit Torrent Networks with the Help of DHT Protocol

V. E. Rabinovich, A. A. Shestakov

Bit Torrent protocol is a leader of file transfer protocols. The main aim of the protocol is to distribute the file to a lot of participants without reducing the file distribution accessibility. While creating, the protocol was intended as a central node (tracker), which stores and distributes participants' list of distribution. As the owner of the tracker, you can manage traffic distribution inside the Bit Torrent network, simply changing the list of participants.

Over time, the question of decentralization arose. It was facilitated by several factors: the inability of trackers to serve huge number of participants, a large number of independent trackers, right possessors' claims for content distribution protected by copyright. The solution was to add to the protocol BitTorrent[5] describing the distribution of the participants with the help of DHT protocol. As a result, the possibility to influence traffic distribution or to manage the list of participants was eliminated.

This paper describes a method you can influence the network traffic and BitTorrent protocol and to manage if not fully but partially the process of file distribution. It can be useful as for the Internet providers for data flows redistribution within their network and for the right possessors to limit file distribution.

Keywords: computer systems, computer network, distributed systems, server, customer, protocol, HTTP, BitTorrent, tracker, DHT.