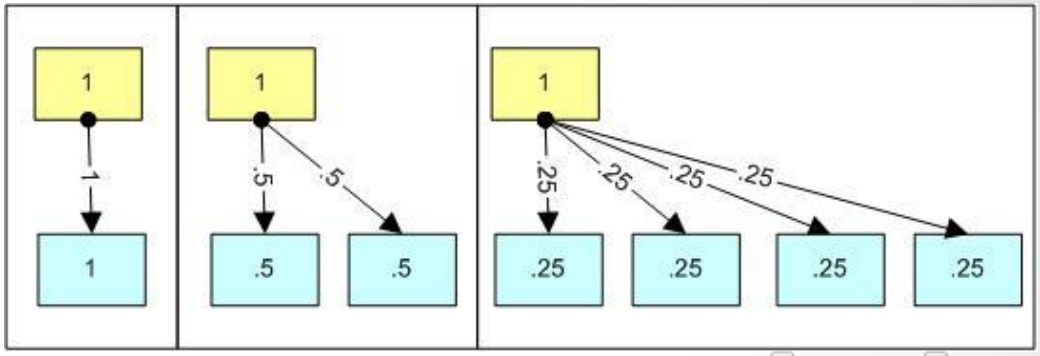
Содержание

Введение (1)

Задача: передать один или несколько файлов от одного пользователя к другому через Интернет.

Решение: HTTP или FTP сервер: файлы хранятся на сервере, пользователи одновременно скачивают файлы



Проблемы:

- Пропускной канал сервера не бесконечен – чем больше пользователей, тем меньше скорость скачивания каждого
- Если сервер «упадет» - скачивание станет невозможным

Вывод: нужна альтернатива, которой и стали пиринговые сети, и BitTorrent в частности.

BitTórrent (англ. «битовый поток») — пиринговый (p2p) сетевой протокол для кооперативного обмена файлами через Интернет.

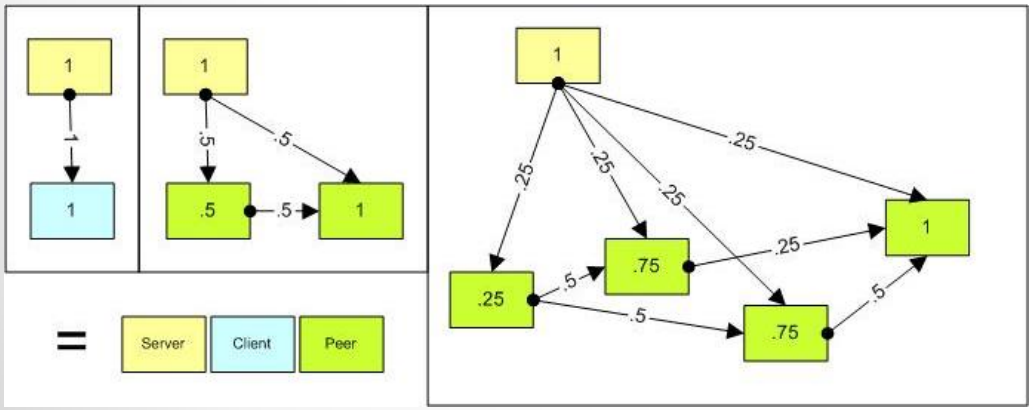
Введение (2)

Пи́ринговая (от англ. *peer-to-peer, P2P* — равный к равному) **сеть** - это компьютерная сеть, основанная на равноправии участников.

- выделенные серверы отсутствуют
- каждый узел (peer) – как клиент, так и сервер.
- файлы хранятся на каждом узле
- скачивание может производиться с любого количества узлов одновременно

Децентрализованная ← Пиринговая сеть → Частично децентрализованная (гибридная)

Существуют сервера для вспомогательной работы, сопутствующей файлообмену



История

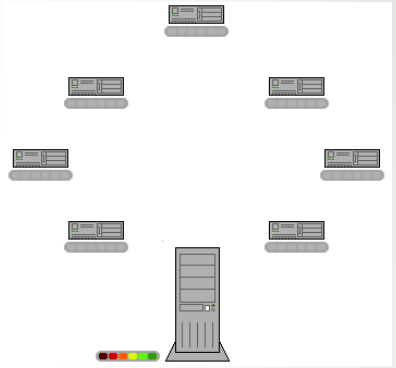
- **Napster** – 1999-2001гг. – первый широко используемый сервис свободного обмена музыкой. Остановлен по решению суда.
- **Kad** – 2002г – децентрализованная сеть поиска, используется совместно с сетью обмена eDonkey. Основана на реализации распределенной хэш таблицы Kademlia.
- **eDonkey** – частично децентрализованная сеть обмена файлов, серверы используются только для координации клиентов. Отсутствует возможность модерации контента.

- **Direct Connect** – гибридная сеть, состоящая из серверов – хабов, и клиентов. Имеет развитый чат, наличие операторов, контролирующих соблюдение правил чата и файлообмена. Отсутствие шифрования данных.
- **BitTorrent** – гибридная сеть,состоящая из серверов – трекеров, и клиентов.

Протокол BitTorrent разработан в 2001г. Бремом Коэном.

Общий принцип работы сети BitTorrent (1)

Задача: передать один или несколько файлов от одного пользователя к другому через Интернет.



Решение при помощи сети BitTorrent:

1. Источник файла/ов создает файл метаданных.

Это множество файлов теперь называется *раздачей*.

2. Источник передает пользователям файл метаданных,

связывая его с сервером – трекером.

3. При помощи торрент клиента источник «встает на раздачу», и становится сидом.

Сид, *сидер* (англ. *seeder* — сеятель) — пир, имеющий файл в полном объеме, источник раздачи.

4. Остальные пользователи начинают скачивать файлы, используя клиент, и становятся личерами

Лич, личер (англ. *leech* — пиявка) — пир, не имеющий пока файлов целиком, т.е. продолжающий скачивание.

5. Личеры, полностью скачавшие файлы, становятся сидами.

Общий принцип работы сети BitTorrent (2)

Преимущества:

- Устранены проблемы решения «сервер-клиент»
- Эффективный распределенный обмен
  - Раздача делится на части одинакового размера – сегменты (англ. *pieces*). По умолчанию размер - 2^{18}=256Кб
  - Сегменты делятся на части одинакового размера – блоки (англ. blocks). Размер 64-4096Кб
  - Каждый пир скачивает каждый сегмент только один раз (при отсутствии ошибок)
- Проверка целостности скачанной раздачи и каждого сегмента в отдельности
- Отсутствие очередей и ограничений на скорость скачивания (в сравнении с eDonkey и DC сетей с квотами)
- Возможность создания сообщества на основе трекера
  - Контроль раздаваемого материала

Файл метаданных

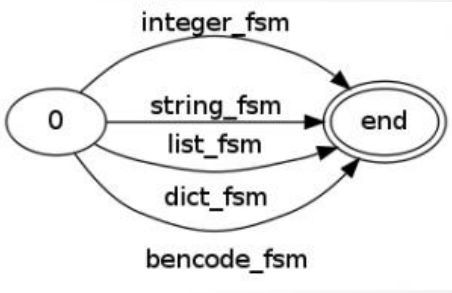
- словарь в *bencode* формате с расширением *.torrent*

Содержит метаданные различного типа:

- информация о файлах в раздаче
  - имена
  - размер
  - расположение
  - хэш файлов и раздачи в целом
- ссылки на торрент трекеры
- данные используемые расширениями
  - поддержка альтернативных протоколов обмена
  - дополнительные опции

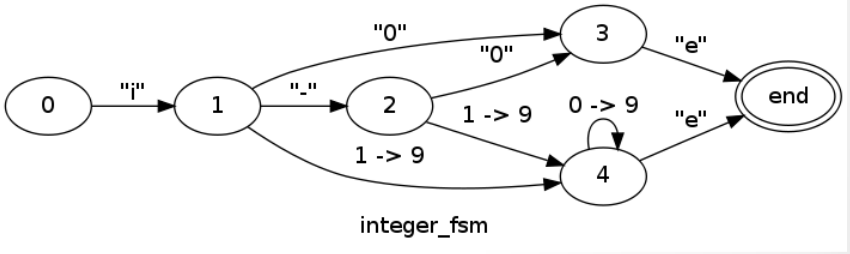
Кодирование bencode

- Использует числа и ASCII символы
- Прост в кодировании и декодировании
- Легко расширяется

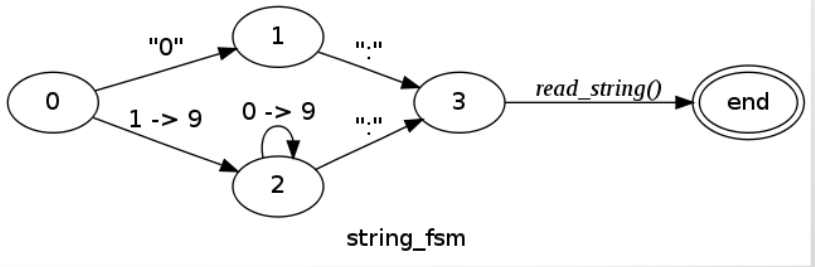


Поддерживаемые типы:

- Целое число
  - Синтаксис: i<число>e
  - Примеры: i30e, i-42e, i0e

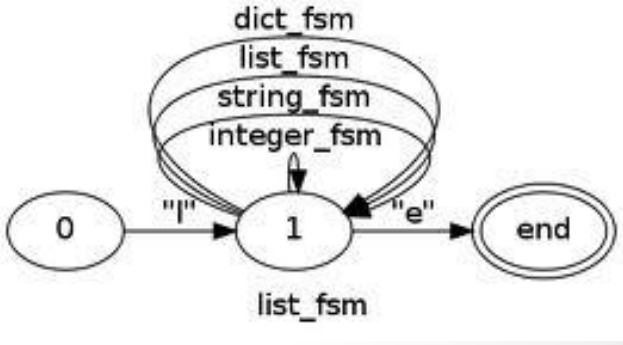


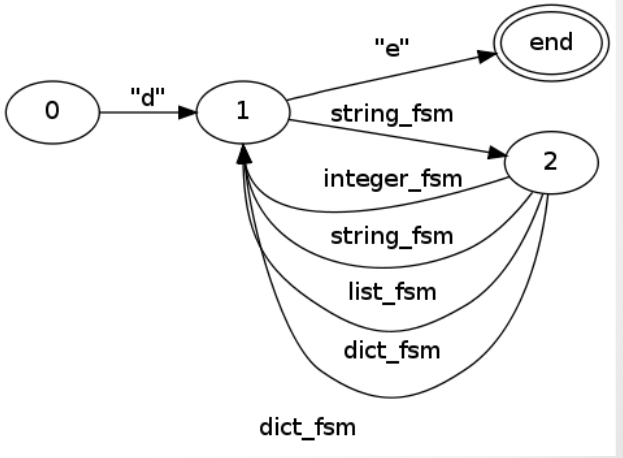
- Строка байтов
  - Синтаксис - <длина строки>:<строка>
  - Примеры: 6:primat, 0:



Кодирование bencode (2)

Поддерживаемые типы:





- Список значений
  - Синтаксис – l<список>e
  - <список> - последовательность bencoded значений
  - Пример: **l**i30e3:fooi0e**e**
- Словарь
  - синтаксис – d<словарь>e
  - <словарь> - список пар <ключ><значение>
  - Ключ - байтовая строка
  - Сортировку следует делать с
  - помощью бинарного, а не "естественного" (алфавитноцифрового), сравнения
    Пример: **d**9:publisher3:bob18:publisher.location4:home 17:publisherwebpage15:www.example.com**e**

Кодирование bencode: вопросы

Как будет закодирован словарь

- { "cow" => "moo", "spam" => "eggs" }

  d3:cow3:moo4:spam4:eggs**e**
- { "spam" => ["a", "b"] }

  d4:spam**l**1:a1:b**ee**

Структура файла метаданных

- announce – URL трекера
- info – словарь
  - name - имя файла/имя корневой директории
  - piece length - размер сегмента. Наиболее используемый 2^{18} байт = 256 Кб
  - pieces - строка, составленная объединением всех SHA1 хешей (по одному на каждый кусок)
  - length - размер файла в байтах, если файл один в раздаче
  - files - список словарей для каждого файла в раздаче. Структура каждого словаря:
    - path - путь до файла относительно корневой директории
    - length - размер файла в байтах
- announce-list, creation date, comment, created by

Все строки кодируются в UTF-8

Примеры структуры файла метаданных

Один файл

```
{
  'announce':
'http://tracker.site1.com/announce',
  'info': {
    'name': ‘Disk.iso',
    'piece length': 262144,
    'length': 678301696,
    'pieces':<SHA-1 160-bit hash>
  }
}
```

Несколько файлов

```
{
  'announce':
'http://tracker.site1.com/announce',
  'info': {
    'name': 'directoryName',
    'piece length': 262144,
    'files': [ {'path': '111.txt', 'length': 111},
               {'path': '222.txt', 'length': 222} ],
    'pieces':  <SHA-1 160-bit hash x2>
  }
}
```

BitTorrent-трекер

— веб-сервер, осуществляющий координацию клиентов BitTorrent.

Трекер (англ. tracker) - от track – отслеживать

Чаще всего работает при поддержке веб-сайта для пользователей – форума или каталога.

Роль и задачи:

- «связывает» клиентов друг с другом
  - при отключении новые клиенты не могут друг друга «найти», при этом уже соединившиеся продолжают обмениваться файлами
- сохраняет в базе данных статистику раздач
  - объёмы переданных данных

	Посл. обновл.	Сегодня	Вчера	Всего учтено
Скачано	0 B	**0 B**	555 MB	**879.02 GB**
Отдано	0 B	**0 B**	4.8 GB	**7.382 TB**

  - количество узлов на каждой раздаче

Статистика раздачи

Размер: **1.72 GB** | Зарегистрирован: **2 дня 19 часов** | .torrent скачан: **7397 раз**

Сиды: **1061** [18 MB/s] Личи: **1514** [18 MB/s]

Трекер никак не участвует в файлообмене и не хранит файлы из раздач!

Этапы работы протокола

- Обновление (англ. *announce*)
  - Клиент подсоединяется к трекеру, сообщая информацию о файле
  - Трекер сообщает клиенту адреса других клиентов
- Основной этап
  - Клиенты обмениваются сегментами
  - Клиенты информируют трекер о ходе процесса и обновляют список адресов
- Завершение (режим *End Game*)
- Сидирование

Этап обновления (1)

- HTTP протокол между клиентом и трекером

- Параметры запроса

info_hash	SHA-1 хэш значения ключа info файла метаданных
peer_id	уникальный ID клиента для данной раздачи
peer_ip	адрес клиента (необязателен)
port	порт, который слушает клиент (обычно 6881-6889)
uploaded	общее количество скачанных и отданных «данных» со старта закачки
downloaded	
left	количество байт, оставшееся клиенту до завершения
event	Тип события: пустой, started, stopped или completed. started/stopped – клиент подключился/отключился от раздачи, completed – клиент завершил скачивание

Этап обновления (2)

Ответ трекера – bencoded словарь

1. Ошибка:
   - failure reason – строка с информацией
2. Успех:
   - interval - интервал времени в секундах до следующего запроса
   - tracker id - строка, которую клиент должен посылать обратно последующих оповещениях
   - peers – список словарей. Структура словаря каждого пира:
     - peer_id – уникальный идентификатор пира
     - ip – ip адрес пира
     - port – порт пира

Протокол scrape

- дополнительный протокол запроса клиента к трекеру, при котором трекер сообщает клиенту общее количество сидов и пиров на раздаче

Scrape – соскоб (от прагерм. skrapojan «скрести»)

В отличие от announce, запрос scrape:

- не имеет прямого отношения к скачиванию раздачи
- является необязательным
- может запрашиваться и для остановленных в клиенте заданий
- отнимает меньше ресурсов у клиента и трекера
- может одним запросом получить информацию сразу по нескольким торрентам (multi-scrape)

Этап обмена данными

- На транспортном уровне используется протокол TCP
- Соединения между пирами симметричны
- Протокол оперирует сегментами и блоками по индексу
- Обмен ведется блоками

№ ...	Размер	Блоков	Полоса загрузки блоков
146	2.00 MB	128	
523	2.00 MB	128	

Стадии обмена

- Приветствие - соединение пиров, обмен информации об имеющихся сегментах
- Обмен данными по специальному алгоритму
- Проверка скачанных данных

Алгоритм обмена данными (1)

Входные данные:

Соединение между 2-мя клиентами: «мы» и пир

Параметры соединения с каждой стороны:

- choking – «мы» душим пира, т. е. не желаем передавать ему данные

Choke – душить

- interested – «мы» заинтересованы в пире, у него есть сегменты или блоки, которые «нам» нужны

Изначально все друг друга душат и не интересуются

am_choking := 1, am_interested := 0

peer_choking := 1, peer_interested := 0

Алгоритм обмена данными (2)

Основная идея алгоритма:

Пока есть не все сегменты

1. Обновление параметров соединения
   - обновление битов заинтересованности при появлении у пира требуемых сегментов (даже если он нас душит)
   - обновление битов удушения по эвристикам и правилам
2. Передача данных
   1. условие передачи от нас к пиру
   am_choking = 0, peer_interested = 1
   2. выбор сегментов для обмена
      - приоритет редким сегментам
      - среди редких – выбор случаен

Обновление состояния удушения

- Каждый пир должен стремиться отдать столько же сколько и скачать – работает идиома «ты мне – я тебе» (англ. Tit-for-tat-ish)
- Если пир нарушает эту идиому – к нему применяется удушение – временная блокировка отдачи.
- Разблокировка происходит как только пир выравнивает отношение скачанного к отданному (англ. upload rate).
- Оптимистическая разблокировка (англ. optimistic unchoking)
  - Всегда есть пир разблокированный независимо от upload rate
  - Этот пир регулярно меняется каждые 30 сек
- Проблема «пренебрежения» (англ. anti-snubbing) - все пиры в состоянии choking с нами.
  - Если пир не передает нам данные больше минуты, он нами пренебрегает: прекращаем ему передавать в ответ
  - Мы получим хотя бы один optimistic unchoke и продолжим скачивание

Протокол обмена данными

1. Приветствие (англ. *Handshaking* – рукопожатие)
   - префикс – ASCII символ с кодом 19
   - строка “BitTorrent protocol”
   - SHA-1 hash в bencoded форме из метафайла
   - peer id
2. Обмен сообщениями
   - Keepalives – сообщения длины 0, для поддержания соединения. Обычно посылаются с таймаутом 2 минуты
   - Non-keepalives – сообщения ненулевой длины. Состоят из типа и данных. Тип – число от 0 до 8, размер – байт. Размер данных зависит от типа.

Типы сообщений

1. choke(0), unchoke(1), interested(2), not interested(3). Сопутствующие данные отсутствуют.
2. bitfield(4). Посылается единожды в начале. Данные – битовая маска сегментов, имеющихся у пира.
3. have(5) – информирует пира о новом полученном и проверенном сегменте. Данные – число, индекс сегмента.
4. request(6) – запрос блока. Данные – индекс сегмента, смещение и размер.
5. piece(7) – получение части сегмента. Данные – индекс сегмента, смещение, данные
6. cancel(8) – отмена запроса. Данные - индекс, смещение, размер
7. (9-255) – сообщения доступные расширениям

Режим End Game

Специальный режим в конце скачивания

- Клиент запрашивает все оставшиеся блоки у всех подключенных пиров
- После получения блока отменяет запрос всем остальным пирам
- Условие входа не фиксировано спецификацией. Варианты:
  1. Все сегменты запрошены
  2. Кол-во оставшихся блоков меньше кол-ва передаваемых блоков, но не больше чем 20

Режим сидирования

Режим клиента, когда он владеет всеми сегментами в раздаче.

- отдает данные
- производит обновление на трекере
- не делает choke по рейтингу

Режим супер-сида (super-seed mode)

Сид маскируется под пира, не имеющего никаких сегментов.

При подключении пира, оповещает его о наличии единственного сегмента, которого еще ни у кого нет.

- Разные пиры получают разные сегменты
- Режим отлично подходит для старта большой раздачи с единственным источником
- Время первого скачивания раздачи уменьшается в 1,5-2 раза

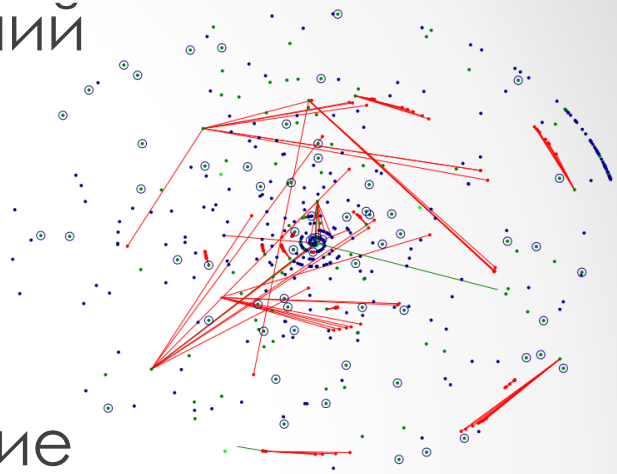
Расширения протокола

- Расширения поддерживаются различными клиентами
- Определение поддерживаемых расширений во время приветствия – новое тип сообщения extended, данные – 64 бита – маска поддерживаемых расширений
- Официальные расширения – поддерживаются всеми клиентами
  - Fast Peers
    - Добавляет сообщения: Have All (0x0E), Have None (0x0F), Suggest piece (0x0D), Reject Request (0x10), Allowed Fast (0x11)
  - Distributed Hash Table – расширение безтрекерного обмена
  - Шифрование соединения
- Неофициальные расширения
  - WebSeeding
  - Extension protocol
  - BitTorrent Location protocol
  - BitComet extension protocol
  - Azureus messaging protocol

Безтрекерный обмен

Реализован при помощи расширений

1. DHT – Distributed Hash Table – механизм обнаружения и соединения пиров, в основе - реализация Kademlia
2. PEX – Peer exchange – расширение протокола для обмена списками известных пиров
3. Magnet links – ссылка на файлы, находящиеся у источника в p2p сети.



Классификация трекеров

- Большая часть трекеров – с регистрацией по приглашениям
- Трекеры с регистрацией учитывают отношение скачанного к розданному – рейтинг.
  - Пользователям приходится находиться на раздаче
  - Почти нет мертвых раздач
- Пути окупаемости затрат на содержание трекера
  - Реклама
  - Продажа рейтинга
  - Продажа инвайтов

Большинство трекеров работают только благодаря энтузиазму владельцев и добровольным пожертвованиям пользователей!

Вопросы легальности

Файлообмен в сетях BitTorrent трудноконтролируем, причем только через трекеры. Безтрекерный обмен неподвластен контролю.

Дилемма:

Разрешить трекеры – допустить пиратство и нарушение авторского права.

Запретить трекеры – уничтожить очень удобный доступ обмена легальной информацией.

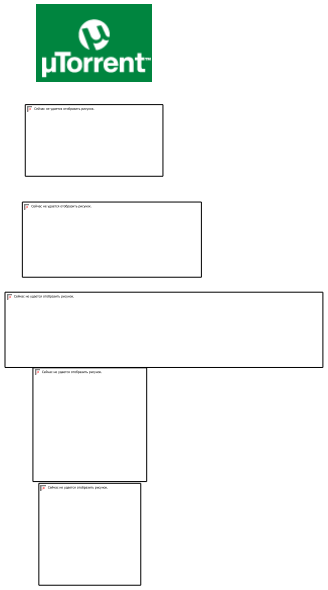
Формально, протокол и трекеры НЕ нарушают никаких законов.

Примеры:

- Закрытие Napster
- Судебный процесс над создателями ThePirateBay.org
- Перспективы развития трекеров в России

Программное обеспечение

- Торрент клиенты
  - µTorrent
  - BitTorrent
  - Vuze/Azureus
  - BitComet
  - Transmission
  - Xunlei



- Торрент трекеры
  - XBT Tracker (XBTT)
  - OpenTracker
  - Bitstorm
  - MonoTracker

Подробнее:
http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients
http://en.wikipedia.org/wiki/BitTorrent_tracker_software