

# Reading Report: Cohen03

Ricard Medina Amado

May 8, 2023

Upload your report in PDF format.

Use this LaTeX template to format the report, keeping the proposed headers.

The length of the report must not exceed **5 pages**.

## 1 Content

### 1.1 *Identify the genre<sup>1</sup> of the document, its purpose, and its target audience.*

Es tracta d'un paper escrit per Bram Cohen al 2003. El seu propòsit és explicar que fa i com funciona BitTorrent.

### 1.2 *Summarize the document, indicating the key concepts<sup>2</sup>.*

## 2 Summary

### What BitTorrent Does

Quan un fitxer està disponible utilitzant HTTP tot el cost recau sobre el host. En canvi amb l'ús de BitTorrent, quan algú descarrega l'arxiu, contribueix pujant per la seva compte part del fitxer, el que ajuda a la distribució del cost de pujada. No és un sistema aplicable a gran escala degut a que causari a excessiu overhead. Per una altra banda, els peers tenen connexions de temps limitat i hem d'aconseguir que la suma de pujades dels peers sigui igual a la de baixada, ho farem limitant la baixada en funció de la pujada aportada a la xarxa.

- **BitTorrent Interface:** L'interface utilitzada per BitTorrent és el més senzill possible. Com qualsevol pàgina típica es basa en un botó de download i una barra de progrés de la descàrrega. BitTorrent ja s'encarrega de pujar part dels teus fitxers per als altres peers.
- **Deployment:** La decisió d'utilitzar BitTorrent és sempre del propietari del fitxer. Molts dels usuaris després de completar la descàrrega deixen d'utilitzar BitTorrent i, per tant, de compartir els fitxers. Aquesta pràctica no és del tot correcta moralment, ja que aprofiten el servei però no ajuden a la seva contribució.

---

<sup>1</sup>Genres: book, article, essay, report, review, manual, white paper, data sheet, weblog, etc.

<sup>2</sup>The summary should help you to answer the questions about the reading in the exam.

## Technical Framework

- **Publishing Content:** Per a publicar un fitxer s'ha de pujar a un servidor un arxiu .torrent amb la seva mida, nom, URL, informació de hash, etc. Els trackers són els encarregats d'ajudar als diferents 'downloaders' a trobar-se entre si. Per a comunicar-se entre ells i saber detalls de les descàrregues es fa via HTTP. Els requeriments d'ampla de banda són baixos, però la seed ha d'enviar mínim una còpia completa del arxiu original.
- **Peer distribution:** Encara que el tracker reb informació de les pujades i baixades aquesta informació només s'utilitza per a estadístiques, la seva missió principal és intercomunicar els peers. Encara que els trackers són l'única manera que tenen els peers per trobar-se i coordinar-se, l'algorisme estàndard del tracker és retornar una llista aleatòria de peers i els peers descarreguen continuament peces de tots els peers amb els que poden connectar. BitTorrent talla els fitxers en peces de mida fixa, i cada descarregador informa a tots els seus peers de les peces que té perquè puguin intercanviar-les. S'inclouen sumes de verificació SHA1 de totes les peces per garantir la integritat de les dades. BitTorrent utilitza un enfocament senzill, en què els peers anuncien el que tenen, i això resulta en menys d'un deu per cent d'amplada de banda sobrecàrrega i utilitza tota la capacitat de pujada disponible.
- **Pipelining:** Quan s'utilitza comunicació via TCP, com fa BitTorrent, és necessari tenir varies requests a la vegada, per a evitar excessiu delay. BitTorrent ho fa segmentant les peces en sub-peces de normalment 16 KB i sempre mantenint al menys 5 requests a la pipeline. Sempre que arriba una subpeça es fa un nou request.
- **Piece selection:** És important tenir un algorisme que esculli l'ordre de descàrrega de les peces correctament per a assolir un bon funcionament. D'altra manera és possible que tinguis les mateixes peces que la resta i no ofereixis res útil.
- **Strict Priority:** La primera política de selecció de peces de BitTorrent és que un cop s'ha sol·licitat una sub-peça, les sub-peces restants d'aquella peça en particular són sol·licitades abans que les sub-peces d'altres peces. Això permet obtenir peces completes tan ràpid com sigui possible.
- **Rarest First:** El "rarest first" és una tècnica que utilitzen els peers de BitTorrent per seleccionar quina peça descarregar a continuació, prioritzant les peces que tenen menys peers propis. Aquesta tècnica assegura que els peers tenen les peces que tots els seus peers volen i que les peces més comunes es deixen per més tard. Això redueix la probabilitat que un peer que ofereixi pujada no tingui res d'interès més tard. També és més eficient per als desplegaments amb un sol seed que té una capacitat de pujada limitada, ja que redueix les descàrregues redundants i ajuda a evitar la pèrdua de peces a mesura que els peers deixen de pujar. Així doncs, el "rarest first" és una tècnica que ajuda a maximitzar la velocitat de descàrrega i a mantenir la disponibilitat de les peces a través de la xarxa.

- **Random first piece:** L'excepció del mètode 'rarest first' és l'inici de la descàrrega, quan el peer no té cap peça per pujar. En aquest cas necessita obtenir una peça per a pujar el més ràpid possible així que es selecciona una aleatòriament. Les peces rares són de baixada més lenta degut a que solen estar a un solpeer, per el que no és útil utilitzar 'rarest first'.
- **Endgame mode:** A vegades es pot demanar una peça a un peer amb una velocitat de transferència molt lenta. Això no és un problema al mig d'una descàrrega, però pot retardar el final de la descàrrega. Per evitar això, un cop s'estan sol·licitant totes les sub-peces que un peer no té, s'envien sol·licituds per a totes les sub-peces a tots els peers. Es cancel·len les sub-peces que arriben per evitar malgastar massa ample de banda en enviaments redundants. En la pràctica, no es malgasta molta amplada de banda d'aquesta manera, ja que el període final és molt curt, i el final d'un fitxer es descarrega sempre ràpidament.

### Chocking algorithms

En BitTorrent, no hi ha una assignació de recursos centralitzada, cada peer intenta maximitzar la seva pròpia taxa de descàrrega descarregant de qualsevol peer i decidint quins peers carregar. Els peers cooperen carregant, i si no cooperen, s'aplica el "choking" que és una negativa temporal per carregar, però la descàrrega encara pot continuar. L'algorisme de choking no és part del protocol BitTorrent, però és important per a un bon rendiment, i ha de ser resistent als peers que només descarreguen i no carreguen.

- **Pareto efficiency:** BitTorrent busca assolir l'eficiència de Pareto, en què cap de les parts no pot millorar sense empitjorar la situació de l'altra, per optimitzar la transferència de fitxers. Aquesta eficiència es busca a través d'algoritmes de "tit-for-tat" més detallats que els usats en el joc del dilema del presoner. BitTorrent intenta que els peers cooperin entre ells per intercanviar fitxers de manera eficient, per exemple, carregant dades als peers que carreguen les seves dades. Això ajuda a maximitzar la taxa de descàrrega i millorar el rendiment del sistema.
- **BitTorrent's choking algorithm:** En BitTorrent, cada peer permet un nombre fix d'altres peers, i la decisió de quins escollir es basa en la taxa de descàrrega actual. Això es fa per permetre que el control de congestió de TCP sature de manera fiable la capacitat de càrrega. El càlcul de la taxa de descàrrega actual és un problema complex, per la qual cosa l'implementació actual utilitza una mitjana mòbil de 20 segons. Per evitar el malbaratament de recursos, els peers recalculen qui volen "chokejar" cada deu segons i després deixen la situació com està fins que passin deu segons. Això permet que TCP pugui augmentar les noves transferències fins a la seva capacitat total.
- **Optimistic unchoking:** Pujar fitxers als peers que proporcionen la millor taxa de descàrrega no és eficaç perquè no hi ha cap mètode per descobrir si les connexions no utilitzades són millors que les que s'estan utilitzant actualment. Per solucionar això, cada peer de BitTorrent té sempre una "optimistic unchoke", que es desbloqueja independentment de la taxa de descàrrega actual. Quin peer serà l'optimistic unchoke és rotat cada tercer període de rechoke (30 segons). Els desbloquejos optimistes

corresponen molt fortament a cooperar sempre en el primer moviment en el dilema del presoner.

- **Anti-snubbing:** A vegades un peer de BitTorrent es queda sense connexió amb els altres peers dels quals estava descarregant i continua obtenint males taxes de descàrrega fins que l'optimistic unchoke troba millors peers. Per resoldre aquest problema, BitTorrent assumeix que està "ignorant" pel peer si no rep cap part seva durant un minut i no li puja res excepte com a optimistic unchoke. Això pot provocar més d'un optimistic unchoke concurrent, el que ajuda a recuperar ràpidament les taxes de descàrrega quan baixen.
- **Upload only:** Un cop un peer ha acabat de descarregar, ja no té taxes de descàrrega útils per decidir a quins peers pujar-hi informació. En aquest cas, la implementació actual passa a preferir els peers amb els quals té millors taxes de pujada

### Real world experience

BitTorrent ja està implementat i desplegat a gran escala, oferint rutinàriament fitxers de centenars de megabytes a centenars de descarregadors simultanis. El principal obstacle de l'escalabilitat actual sembla ser la sobrecàrrega de la amplada de banda del tracker, que encara no s'ha assolit, però es creu que es pot reduir a una milèsima part del total actual amb algunes extensions menors del protocol.

## 3 Assessment

### 3.1 *Rate the readability of the document: easy, readable, difficult, unreadable.*

Ha estat un article fàcil de llegir ja que no té gairebé tecnicismes.

### 3.2 *Give your opinion of the reading assignment, indicating whether it should be included in next year's course or not.*

La temàtica està bé, encara que no m'ha semblat tant interessant com els anteriors.