

Bittorrent - Ett distribuerat nedladdningssystem

Contents

1	Introduktion	3
2	Vad är BitTorrent?	3
2.1	Metainfo-filer	3
2.2	Jargong	3
3	Teknisk information	4
3.1	Datatyper	4
3.2	Trackers	4
3.3	Metainfo	4
3.3.1	Metainfofält	4
3.4	Fildelar	5
3.4.1	Sub-delar	5
3.5	Anslutningar	5
3.5.1	Choking	5
3.5.2	Handskakning	6
3.5.3	Följande meddelanden	6
4	Prioritering av delar	6
4.1	En del i taget	7
4.2	Ovanligast först	7
4.3	Specialfall i prioritering	7
4.4	Färdiga nedladdningar	7
5	Strypande algoritmer	7
5.1	Prisoner's Dilemma	7
5.2	Paretoeffektivitet	8
6	BitTorrents choking algoritim	9
7	Optimistisk avstrypning	9
8	Anti-snubbing	9
9	Endast uppladdning	10
10	Etikett	10
11	Juridik	10
11.1	BitTorrentsidor	10
11.2	Digital Millenium Copyright Act - DMCA	11
11.3	EU Copyright Directive - EUCD	11
11.4	Juridiska fördelar	11
12	Användningsområden	11
12.1	Framtida användningsområden	12
13	Källförteckning	12

1 Introduktion

BitTorrent är ett distribuerat fildelningssystem som använder sig av “tit for tat” principen. BitTorrent är mer robust mot förändringar och har ett bättre utnyttjande av resurser än något annat samarbetsystem inom den här typen av område. Något som är mycket intressant med BitTorrent är hur teorier som härstammar från bl a spelteori och användandet av statistik kan användas för att skapa ett fildelningssystem.

2 Vad är BitTorrent?

BitTorrent är ett distribuerat fildelningssystem som delar upp filer i smådelar (“pieces”) som sedan kan skickas oberoende av varandra. Den huvudsakliga fördelen är att när en “piece” väl har hämtats till en klient, så kan denna “piece” delas ut till andra klienter som inte har den, på så vis så behöver inte en server ladda upp en fil 1000 gånger för att 1000 klienter ska få filen.

2.1 Metainfo-filer

För att dela ut en fil med hjälp av BitTorrent, behöver man bygga en metainfo-fil som innehåller information om filen och hur BitTorrent kommer att hantera den. Den metainfo-fil som skapas görs sedan tillgänglig, vanligtvis med hjälp av en websida. Användare kan gå in på hemsidan och enkelt ladda ner metainfo-filen, vars storlek är obetydlig. Efter detta så sköter klienten resten, förutsatt att det finns användare att ladda ner ifrån.

2.2 Jargong

Det kan kännas oviktigt att känna till jargongen, men faktum är att det är mer eller mindre nödvändigt för att förstå de kommentarer som man kan läsa om vissa torrents.

- torrent - Filen som laddas ner. Det är antingen metainfo-filen eller filen som denna beskriver som avses.
- tracker - En server som sköter en torrent, trackern hanterar kontakt mellan klienter.
- availability - hur många kopior som finns av filen
- leech - Typiskt en användare som laddat upp en bråkdelen av vad den laddat ner. Vanligtvis betecknas vanliga användare som ännu inte laddat klart en fil som leeches.
- peer - En vanlig användare som inte laddat klart en fil, samma som en leech fast utan den negativa implikationen.
- seed - En användare som laddat klart en fil och endast laddar upp.
- superseed - I begynnelseläget när en fil delas ut, så kan mycket bandbredd slösas bort genom ursprungsklienten laddar upp samma del flera gånger, detta kan kontrasteras med “superseed”-läget, där en klient endast

försöker dela ut delar som inte begärts tidigare. Det är ganska vanligt att användare begär att andra användare ska "seeda" en torrent på forum som är anslutna till en torrent, det som avses är då att användare som är i uppladdningsläge ska ligga i superseed-läge.

- swarm - De användare som är anslutna till en torrent.

3 Teknisk information

3.1 Datatyper

Det finns en mängd datatyper i BitTorrent, som är utformade för att det ska vara enkelt att implementera.

- Strängar - en sträng anges med en siffra som anger strängens längd, och ett kolon följt av själva strängen i ASCII. Exempel: "4:spam"
- Integers - en integer anges med ett 'i' följt av talet i ASCII följt av 'e'. Exempel: "i32e". Negativa tal anges med minustecken, -0 är ogiltigt, begynnande nollor är också ogiltigt.
- Listor - Listor anges med 'l' följt av en sträng följt av en datatyp (listor, ordlistor, integers eller strängar) följt av 'e'. Exempel: "li32e4:spam3:fooe" som bygger listan [32, "spam", "foo"].
- Ordlistor - ordlistor anges med 'd' och avslutas med 'e'. Nycklar måste vara strängar och de värden de korresponderar mot kan vara datatyper. Exempel: "d3:cow3:moo4:spam4:eggse" skapar ordlistan [{"cow": "moo", "spam": "eggs"}].

3.2 Trackers

Den centrala administrationen av en torrent sköts av en tracker. Trackern sköter en lista med klienter som är anslutna, och kontaktinformation till dessa klienter (IP-adress och port). Klienter kommunicerar med trackern med hjälp av ett enkelt protokoll som är implementerat på HTTP. Klienter skickar information till trackern om vilken fil de laddar ner, och hur de själva ska kontaktas. Trackern svarar med en lista med andra klienter som också har filen, och deras kontaktinformation.

3.3 Metainfo

En .torrent-fil är egentligen en metainfofil som innehåller information om vilka filer som en torrent innehåller, hur de är uppdelade och egenskaper hos delarna.

3.3.1 Metainfofält

En metainfofil innehåller följande fält:

- announce - URL till trackern
- info - en typ av lista som BitTorrent hanterar som en slags lista, eller "dictionary".

Infofältet innehåller i sin tur flera delar:

- name - föreslaget namn eller sökväg till filen, beroende på om torrenten innehåller en eller flera filer.
- piece length - storleken på varje del i en torrent
- pieces - En lista som har en längd som är en multipel av 20, listan innehåller av alla delars SHA1-hashar
- length/files - om length anges, så anger length filens längd i bytes, om files anges, så anges filerna som torrenten innehåller, deras sökväg och längd i bytes.

3.4 Fildelar

BitTorrent delar som sagt upp en fil i smådelar som kallas "pieces" , storleken beror på hur man ställer in en fil när man delar ut den, men måste vara en tvåpotens, och är normalt 256 K. Detta beror dock på hur stor filen är, större filer har normalt större delstorlek, en fil med dvd-storlek har normalt en delstorlek som är 4 MB. Äldre standarder använder 1 MB som standardstorlek.

3.4.1 Sub-delar

För att hantera svagheter hos TCP, så bryter BitTorrent ner varje fildel i ytterligare delar, normalt 16kB stora. BitTorrent håller sedan flera förfrågningar igång samtidigt, normalt 5 stycken. Så fort en sub-del har överförts, så frågar klienten om en ny sub-del. Detta refereras till som "pipelining".

3.5 Anslutningar

Anslutningar i BitTorrent hanteras över TCP och är symmetriska, det innebär att meddelanden ser likadana ut oavsett vilket håll de skickas åt. Data kan även gå åt båda håll i anslutningen. Båda dessa egenskaper är viktiga peer-to-peer egenskaper. När en klient har fått en del så jämför den delen mot hashen indexerat med delens id. Om hashen stämmer så upplyser klienten alla klienter som är anslutna om att den har delen i fråga. På så vis vet alla klienter vad klienterna de är anslutna till har.

3.5.1 Choking

Varje anslutning har två tillstånd på varje sida: "choking"/"not choking" och "interested"/"not interested". "Choking" innebär att ingen data kommer att skickas över anslutningen förrän en "unchoke" inträffar. För att en överföring skall kunna inträffa måste den som ska ta emot data vara 'interested' och den som ska skicka vara 'not choking'. Interested måste hållas uppdaterad hela tiden, så att en klient vet vilka andra klienter som skulle påbörja en överföring om en unchoke inträffade. Alla anslutningar är initialt "choked" och "not interested".

3.5.2 Handskakning

En handskakning påbörjas med strängen ”BitTorrent protocol”, kommatecknet är där eftersom det är tecken 19, och den följande substrängen är 19 tecken lång, det används alltså som en längd-prefix. Efter begynnelsesträngen kommer 8 reserverade bytes, satta till 0. Efter den reserverade bytesträngen kommer det en 20 bytes lång SHA1 hash av metainfo-filen, om två klienter inte har överrensstämmande metainfo-hash så bryter de uppkopplingen. Efter hashen kommer det ett 20 bytes peer-id, som identifierar klienten. Om detta id inte överrensstämmer med det värde som erhållits från trackern, så bryts uppkopplingen.

3.5.3 Följande meddelanden

Efter handskakning följer en alternerande ström av kontrollmeddelanden och data. Meddelanden med längd 0 är “keepalive”, dessa skickas normalt ungefär varannan minut.

De meddelanden som inte är “keepalive” börjar med en byte som anger deras värde. De möjliga värdena är:

- 0 - choke
- 1 - unchoke
- 2 - interested
- 3 - not interested
- 4 - have
- 5 - bitfield
- 6 - request
- 7 - piece
- 8 - cancel

’choke’, ’unchoke’, ’interested’ och ’not interested’ innehåller ingen följande data. ’bitfield’ skickas endast som första meddelandet, det innehåller då ett bitfält med bitarna som motsvarar de delar som en klient har satta till 1 och resten till 0. Mest signifikanta biten har index 0 i varje byte, bitar som blir över sätts till 0. ’have’ har ett tal som data, vilket är index på den del som klienten precis mottagit och kontrollerat hashen på. ’request’ innehåller index, start och längd. Längd är normalt en tvåpotens om den inte trunkeas i slutet av en fil. ’cancel’ har samma innehåll som request. ’piece’ innehåller index, start, och delen som index motsvarar.

4 Prioritering av delar

När BitTorrent hämtar delar, så finns det en mängd olika sätt att välja vilken del som ska hämtas. BitTorrent använder greed för att välja vilken del som ska hämtas.

4.1 En del i taget

När BitTorrent väl börjat hämta en del, så föredrar det att hämta färdigt den aktuella delen, för att sedan kunna dela ut den. Detta bidrar till att fler klienter har mer att dela ut.

4.2 Ovanligast först

BitTorrent använder sig av Greed-metoden när det väljer delar, och föredrar att hämta den ovanligaste delen först, vilket bidrar till robusthet och snabbhet. Den ovanligaste delen är ju den svagaste länken i kedjan, eftersom klienter måste ha alla delar av en torrent. Att fler får tillgång till den ovanligaste delen har således en tvådelad fördel, dels försäkring mot att delen försvinner om klienter som har den skulle koppla ner, och gör att fler klienter kan skicka den aktuella delen. Att ta den ovanligaste delen först bidrar också med att klienten alltid har något som andra klienter vill ha.

4.3 Specialfall i prioritering

Det finns två specialfall i nedladdningsprioriteringen, den ena är då en klient inte ännu har några delar, och den andra är då en klient nästan är klar med sin nedladdning. Då en klient inte ännu har några delar är det dåligt att hämta den ovanligaste delen, eftersom man oftast får vänta längre på en ovanlig del än en vanlig. I det här läget väljs del slumpvis. I det läge att en klient skickat förfrågan om alla subdelar av en torrent, så skickas extra-förfrågningar ut till alla anslutna klienter. Detta är för att slippa vänta på klienter som har långsamma uppkopplingar, vilket ofta är skälet till att delen inte har laddat klart. Om en sub-del är mottagen skickas cancel ut till anslutna klienter för att spara bandbredd.

4.4 Färdiga nedladdningar

När en klient har laddat klart en fil, går den över i uppladdningsläge.

5 Strypande algoritmer

Eftersom alla klienter är i BitTorrent är ganska självständiga och trackern inte på något sätt kan kontrollera och motverka fuskare måste klienterna själva upptäcka och straffa fuskare. Detta görs med hjälp av existerande problem och algoritmer som effektivt löser dessa problem.

5.1 Prisoner's Dilemma

BitTorrent använder sig av "Prisoner's dilemma" problemet för att modellera hur klienter samarbetar med varandra. Problemet är klassiskt i spelteori och vissa menar att en variant av systemet är grundläggande i utvecklingen av samhällen. I "Prisoner's dilemma" separeras två brottslingar som tillsammans genomfört ett brott. De förhörs separat och de kan antingen samarbeta med varandra eller förråda varandra. Om de samarbetar får båda 6 månaders fängelse var och om de förråder varandra får båda 2 års fängelse var. Om brottsling A

förråder brottsling B, men B samarbetar (det vill säga håller tyst) så går A fri, medans B får 10 års fängelse. Eftersom den garanterade belöningen i att förråda överväger den osäkra betalningen i att samarbeta i det grundläggande fallet så kommer alla alltid att förråda sin kamrat. Som alla förstår är detta ingen bra princip när man ska bygga ett filöverföringsprotokoll som bygger på att man delar med sig. Det är dock inte så illa som det låter. I BitTorrent är det inte bara ett filsegment som ska skickas utan flera, därför kan vi använda oss av specialfallet "Iterated prisoner's dilemma". Som lösning på detta problem finns den kända och synnerligen välkända lösningen "tit for tat". Den fungerar så här:

- Om inte klienten slutar motta data på en uppkoppling, så kommer klienten alltid att samarbeta och skicka data.
- Om klienten slutar motta data från en annan så kommer den att straffa den andra klienten och sluta skicka data.
- Klienten förlåter andra klienter snabbt. Om den andra klienten börjar samarbeta så börjar vi också samarbeta.
- Det måste finnas minst en 2/3-chans att vi utför mer än ett utbyte av data med den andra klienten.

Det finns inget bevis för att denna algoritm är den bästa, men många andra långt mer avancerade algoritmer har ställts upp mot den har visats sämre, även sådana som byggts för att lura "tit for tat". Detta innebär att det inte går att bygga en klient som lurar andra klienter att samarbeta utan att den själv samarbetar. BitTorrent använder visserligen en mer avancerad variant av algoritmen. Huvudsakligen gäller det mer avancerade sätt att förlåta andra användare eftersom alternativet att båda förråder den andra leder till låsning. Den grundläggande mekaniken är dock densamma.

5.2 Paretoeffektivitet

BitTorrent strävar att genom sin variant av "Prisoner's Dilemma" att uppnå något som kallas Paretoeffektivitet eller Paretooptimalitet. Paretoeffektivitet är en term inom ekonomi som definierar en typ av optimalt system. Man har från början ett system med individer och deras resurssamlingar. En förflyttning av en resurs till en annan resurssamling är en Paretoförbättring om den innebär att minst en individ får det bättre och ingen individ får det sämre. När inga fler Paretoförbättringar kan göras är systemet Paretoeffektivt. Om vissa grundläggande antaganden kan göras, så kan det bevisas att fria marknader ger upphov till ett Paretoeffektivt system.

BitTorrent kan ses som ett system enligt ovanstående modell. Individerna är klienter och resurser motsvaras av överföringshastighet. Paretoeffektivitet innebär alltså i sammanhanget att klienter i par försöker förbättra deras överföringshastighet genom att samarbeta. En viktig detalj med ett Paretoeffektivt system är att dess optimeringar är lokala och samlingen av optimala lokala lösningar kan vara långt sämre den optimala globala lösningen. Eftersom ingen ändring får göras som medför att någon får det sämre har systemet låst sig i en suboptimal lösning.

Detta gäller även i fallet med dataöverföring, för att hindra att systemet låser sig i en dålig lösning så testas oanvända anslutningar med jämna mellanrum.

6 BitTorrents choking algoritm

BitTorrent förlåter klienter enligt “tit for tat” genom “unchoking”. Klienten väljer med jämna mellanrum ut ett antal klienter (standard är 4) som den börjar samarbeta med igen. Frågan är bara vilka uppkopplingar klienten ska börja samarbeta med igen. BitTorrent väljer sina “unchokes” endast genom nedladdningshastigheten på anslutningen. Nu uppstår problemet hur man meningsfullt ska tolka nedladdningshastigheten på ett meningsfullt sätt. Flera metoder har testats, bland annat en metod som använde sig av långsiktiga mängder av överförd data. Metoder som använder sig av allt för långsiktiga mätningar lider av att de svarar för trögt på förändring i överföringshastigheter när resurser blir upptagna eller lediga. Metoder som arbetar med för korta tidsrymder lider å andra sidan av att de är för känsliga för samma förändringar, exempelvis skulle ett system som mäter momentana värden kunna utföra sina mätningar i ett läge där TCPs “additive-increase-multiplicative-decrease” precis har sänkt överföringshastigheten för att undvika stockning. Lösningen som används är ett rullande 20-sekunders medelvärde. Denna lösning är en kompromiss mellan ett system som hinner reagera på snabba förändringar och ett system som är robust mot snabba svängningar. 20 sekunder är gott om tid för att TCP ska hinna bygga upp ett representativt medelvärde på överföringshastighet. För att undvika att slösa resurser på “chokes” och “unchokes” hela tiden så räknar BitTorrent ut vilka anslutningar som ska bli “choked” och “unchoked” var tionde sekund.

7 Optimistisk avstrykning

Om klienten bara skickade data på de anslutningar som den nedladdar mest ifrån så skulle klienten inte ha någon sätt att märka ifall för tillfället oanvända anslutningar är bättre än de anslutningar som vi använder. Dessutom leder Paretoförbättringarna vi gjort till ett lokalt optima som troligtvis inte överensstämmer med den globalt bästa lösningen. Som lösning så har samtliga anslutningar vår klient har varsin “optimistic unchoke”, som betyder att anslutningen blir “unchoked” oavsett vad för nedladdningshastighet vi har från den. En anslutning åt gången är “optimistic unchoke” och vilken anslutning som är vald roteras var trettionde sekund. Tidsperioden är avvägd så att klienten på andra sidan anslutningen ska hinna reagera på samarbetet, skicka data tillbaka och att vår klient ska hinna bygga upp en representativ överföringshastighet. Tanken är att en “optimistic unchoke” motsvarar principen att alltid samarbeta från början i “tit for tat”.

8 Anti-snubbing

Ibland kommer klienten att bli “choked” av samtliga anslutningar som den nedladdar ifrån. I ett sådant fall kommer klienten att få dåliga nedladdningshastigheter tills “optimistic unchoke” funktionen hittar bättre anslutningar.

För att motverka denna situation så kommer att anta att den har blivit "snubbed" av en anslutning när den inte motagit en "piece" på en minut av anslutningen. En sådan anslutning kommer inte klienten att uppladda till förutom de fall då anslutningen valts som "optimistic unchoke". Detta resulterar regelbundet i fler än en samtidig "optimistic unchoke". Detta medför att nedladdningshastigheter återhämtar sig snabbare än den förfaller.

9 Endast uppladdning

När nedladdningen är klar så kommer klienten gå över till att endast uppladda. Eftersom klienten inte längre har några nedladdningshastigheter för att bestämma vilka anslutningar den ska uppladda till så behöver den alternativt sätt att bestämma detta. Den nuvarande implementationen väljer de anslutningar som den har högst uppladdningshastighet till. Metoden gör ett hyfsat jobb att använda så mycket av anslutningen som möjligt och att uppladda till andra klienter som inte nedladdar så mycket.

10 Etikett

Eftersom effektiviteten i BitTorrent baserar sig på upströms bandbredden hos användarna, och fler användare resulterar i en större sammanlagd bandbredd, så är det sett som god etikett att lämna sin BitTorrent-klient uppkopplad även efter man är klar med sin nerladdning.

Det är inte helt klart hur länge man bör fortsätta var uppkopplad efter man är klar. De flesta klienter innehåller information om hur mycket data man laddat ner respektive laddat upp. Vanligt är också en så kallad utdelnings ratio. Den anges som uppladdad data delat med nedladdad data. Denna ratio är mycket viktigare för BitTorrents användare än användare på andra fildelningstjänster, många BitTorrent trackers brukar nämligen kräva att deras användare uppehåller en minimumratio. En vanlig gräns brukar var 0.5. Idealt så skulle alla användare ha en ratio på 1.0, det vill säga att de har bidragit med lika mycket som de har belastat. Praktiskt sett är det en omöjlighet. För att uppmuntra folk att försöka har vissa nätverk implementerat en spärr till nya torrents de första 24-48 timmarna för användare med en ratio under 1.0.

11 Juridik

BitTorrent används till stor del för att kopiera upphovsrättsskyddat material, vilket gör att det fått mycket negativ uppmärksamhet. Det går inte att förneka att websidor som förmedlar torrent-filer förenklar piratkopiering, men frågan huruvida spridningen av torrent-filer, som endast är metainfo om den upphovsrättsskyddade filen är piratkopiering är en omdebatterad fråga.

11.1 BitTorrentsidor

Många BitTorrentsidor har blivit nedstängda av myndigheter som försöker stoppa piratkopiering. "The Pirate Bay" (www.thepiratebay.org), en svensk sida som

ursprungligen skapades av svenska "piratbyrån", skyddas av svensk lagstiftning. Man har kommit fram till att det inte är olagligt att peka på upphovsrättsskyddat material utan tillåtelse, eftersom ingen otillåten kopiering sker. "The Pirate Bay" har en avdelning på sin hemsida som handlar om just den juridiska post de mottagit av olika företag som försöker skydda sin egen eller andras upphovsrätt. Man kan också läsa de svar som pirate bay skickar tillbaka till företagen, som alltid är humoristiskt betonade och förlöjligar hoten från företagen. Hela fenomenet att ha en "legal threats"-sektion på en hemsida är egentligen ingen ny ide, men följer i det här fallet sidans politiska mål.

11.2 Digital Millenium Copyright Act - DMCA

Digital Millenium Copyrigth Act är något som är intressant i det här sammanhanget. Lagen säger nämligen att även produkter och sätt som används för att kringgå upphovsrättsskydd är kriminellt. Detta gör BitTorrentsidor som tillhandahåller filer som kan användas för att bryta mot upphovsrätten olagliga i sig.

11.3 EU Copyright Directive - EUCD

Det finns en lag som liknar DMCA, EU Copyright Directive. Lagen gör mjukvara som kringgår upphovsrättsskydd olaglig. Lagen är dock otydlig i hur den skall implementeras i EUs medlemsländer, och anses av många vara "oviktig, och möjligtvis ogiltig". Lagen gör däremot inte hemsidor som tillhandahåller "olagliga torrents" olagliga, iallafall anses den inte göra det i Sverige.

11.4 Juridiska fördelar

Det finns några fördelar ur juridisk synpunkt med BitTorrent. En av dem är att BitTorrent i sig inte bidrar något sätt att hitta upphovsrättsskyddat material, användare måste i så fall hitta torrents själva, till exempel via en BitTorrentsida som tillhandahåller den typen av torrents. En annan fördel är att BitTorrent inte på något sätt försöker dölja ursprungsklienten, det vill säga den användare som skapade torrent-filen. Det gör det mycket lättare för myndigheter att hitta användare som bryter mot upphovsrättslagar, alternativa sätt brukar vara att ladda ner filer av en person och sedan anklaga denna person för att ha delat ut filen utan tillåtelse.

12 Användningsområden

BitTorrent har förutom piratkopiering en hel del användningsområden. Ett av dessa är patchning av program, där normalt en väldigt stor mängd användare vill hämta en fil snart efter att den blivit tillgänglig, ibland av säkerhetsskäl. Genom att sprida ut uppladdningen över fler användare når man en större mängd användare på en given tid, utan att behöva betala för det. Till exempel använder Blizzard en modifierad version av BitTorrent för att distribuera patchar snabbt. Det har utvecklats verktyg för att bloggare enkelt ska kunna lägga upp filer på sina bloggar, som användare sedan kan hämta med hjälp av BitTorrent. Detta är mycket användbart, eftersom många bloggare ofta inte

har råd med att betala sin ISP de belopp som skulle behövas för att kunna tillhandahålla större filer.

12.1 Framtida användningsområden

Eftersom BitTorrent tillåter tillhandahållande av filer utan stora kostnader i form av bandbredd, är det attraktivt för företag som vill tillhandahålla filer som är eftertraktade av olika skäl, men som inte vill betala för det. Det finns förslag på att kombinera RSS (Really Simple Syndication) med BitTorrent, för att skapa en form av "broadcasting". Tanken är att applikationen själv ska upptäcka när nya filer finns att hämta, och hämta dem.

13 Källförteckning

- Wikipedia:
 - BitTorrent
<http://en.wikipedia.org/wiki/Bittorrent>
 - Digital Copyright Millenium Act
<http://en.wikipedia.org/wiki/DMCA>
 - European Union Copyright Directive
http://en.wikipedia.org/wiki/EU_Copyright_Directive
 - Pareto Efficiency
http://en.wikipedia.org/wiki/Pareto_Efficiency
 - Prisoner's Dilemma
http://en.wikipedia.org/wiki/Prisoner%27s_dilemma
 - Tit for tat
http://en.wikipedia.org/wiki/Tit_for_tat
- BitTorrent, Inc.
<http://www.bittorrent.com/bittorrentecon.pdf>
- BitTorrent protcol specification
<http://www.bittorrent.org/protocol.htm>
- The Pirate Bay
<http://thepiratebay.org>