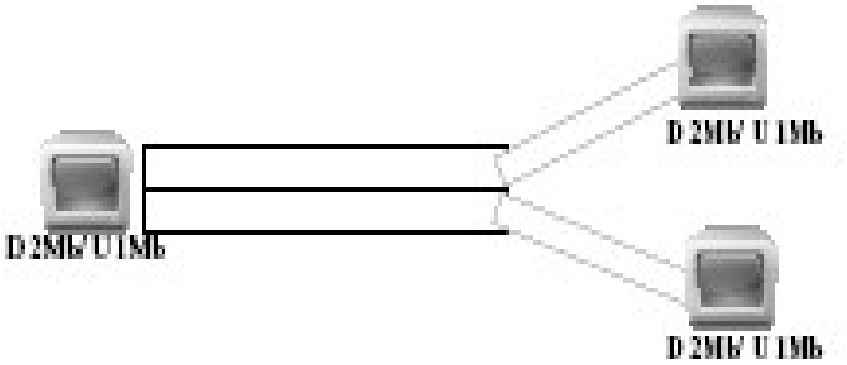
Lezione n.13
P2P CONTENT DISTRIBUTION: BITTORRENT
9/5/2009

CONTENT DISTRIBUTION NETWORKS

- Content Distribution Network (Content Delivery Network) (CDN): Insieme distribuito di hosts che cooperano per distribuire grosse quantità di dati (ad esempio dati multimediali) agli end user
- Motivazione: se un unico web server memorizza dati molto popolari, il server stesso può spesso risultare congestionato e non essere capace di gestire tutte le risorse
- Fenomeno del flash crowd: un sito web 'attira', a causa di qualche evento, l'attenzione di un gran numero di persone e deve gestire un tasso di traffico altissimo ed inaspettato
- CDN: basate sulla replicazione dei dati e/o servizi su diversi mirror server
- Basati su uno sfruttamento migliore della banda
- Esempi:
  - Commerciali: Akami, AppStream, Globix, Bittorrent
  - Accademici: Coral (P2P), Globule

BITTORRENT: OBIETTIVI

- distribuire il carico relativo alla distribuzione del file
- sfruttare pienamente la banda di download per sistemi P2P con connessioni asimmetriche
  - Banda di downlaod >> Banda di upload
  - Es: 4 Mbit/s di downlink, 600Kbit/s di uplink
- idea principale: redistribuire il carico dell'upload su tutti i peer che concorrono allo scaricamento del file



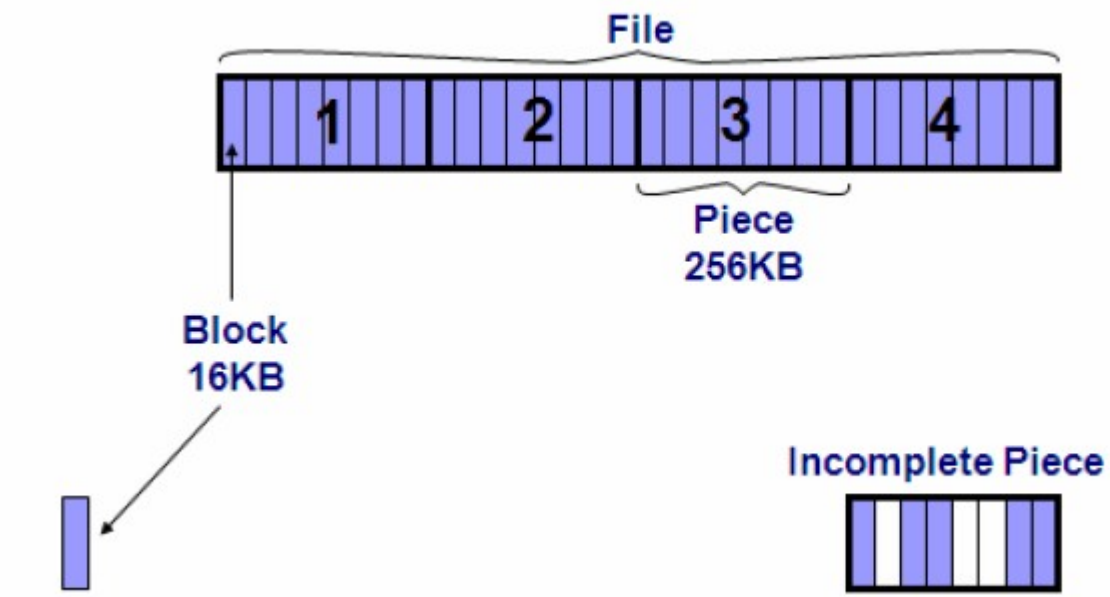
BITTORRENT:CARATTERISTICHE GENERALI

- BitTorrent: l'obiettivo principale è quello della distribuzione di files, non sono definiti meccanismi per la ricerca di files
- Sviluppato da Bram Cohen in Python e presentato al CodeCon 2002 (una conferenza che si tiene negli States in cui sviluppatori di software ed hacker possono liberamente presentare nuovi progetti)
- idea di Cohen: adattare il sistema di memorizzazione dei files adottato nell'azienda di Cohen al P2P.
  - i files vengono divisi in parti crittografate e distribuite tra i diversi computers dell'azienda
- Responsabile di almeno il 35% del traffico Internet
- Utilizzato per la distribuzione legale di software open source
  - Ubuntu/SUSE Linux Distribution
  - OpenOffice
  - Game patches (es: War of Warcraft)

BITTORRENT: FUNZIONAMENTO GENERALE

- Ogni file viene decomposto in parti
- Il peer U che intende condividere un file genera un descrittore del file, il file .torrent, e lo pubblica su un server HTTP.
- Il descrittore contiene un riferimento ad un ulteriore server, il tracker , che coordina la distribuzione del file
- Il peer D che desidera scaricare il file,
  - scarica il descrittore del file e lo apre con il client BitTorrent.
  - si connette al tracker e lo informa della propria esistenza
  - riceve dal tracker una lista di ulteriori peer che stanno scaricando o condividendo il file
- i peer si scambiano informazioni relative alle parti del file che essi possiedono e ogni peer richiede le parti di file a cui è interessato
- quando un peer P ha terminato lo scaricamento del file, P può decidere di rimanere online, e continuare a distribuire il file

BITTORRENT: PARTIZIONAMENTO DEI FILES

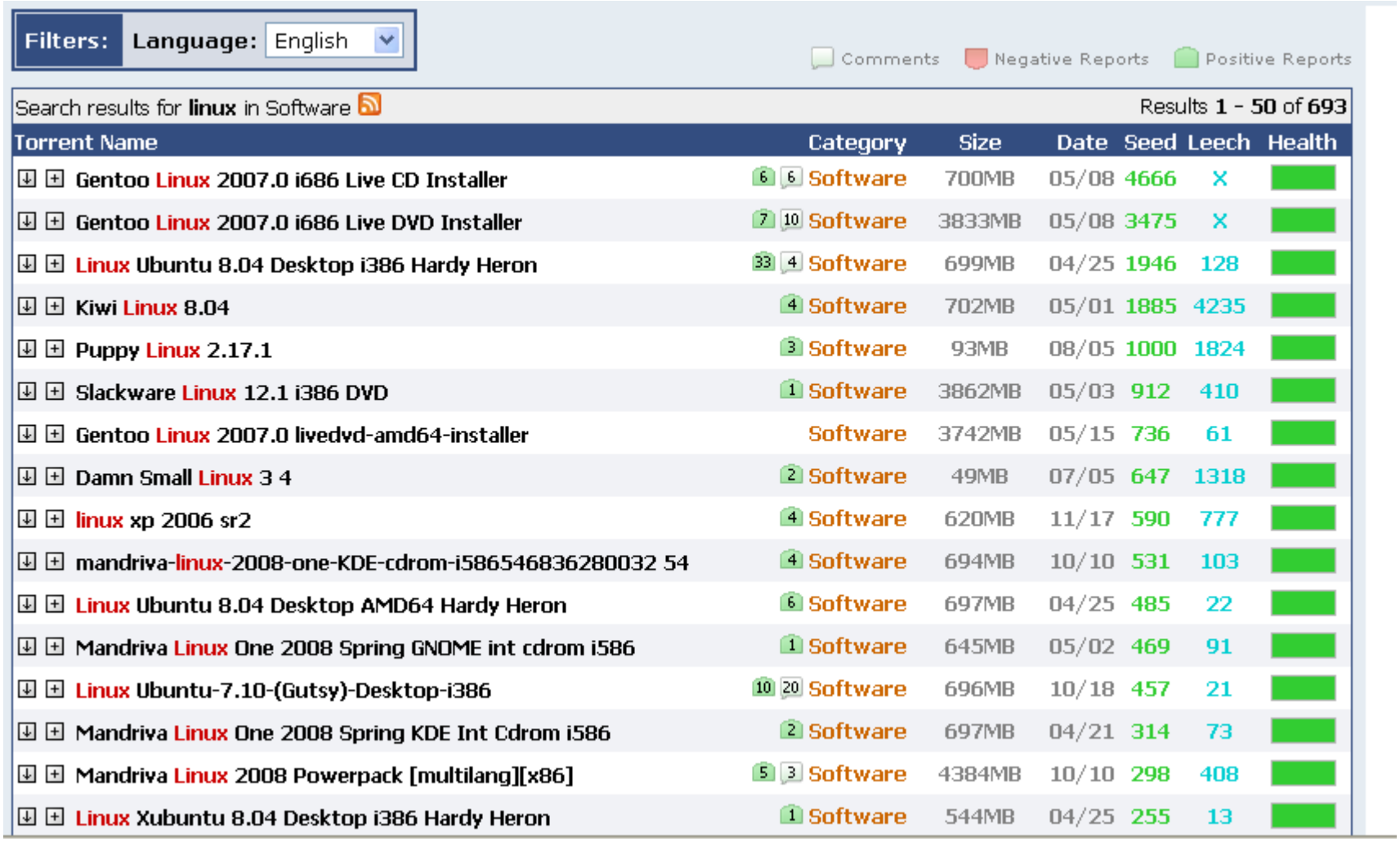


- BitTorrent divide ogni file in pezzi, ognuno di 256 KB
- viene calcolato l'hash di ogni pezzo
- ogni pezzo è quindi suddiviso in blocchi di 16 KB
- trasferimento di blocchi tra i peer

BITTORRENT: IL FILE .TORRENT

- Pubblicazione di risorse:
  - il protocollo BitTorrent non definisce un meccanismo per la ricerca di files
  - il peer che intende pubblicare un file genera un descrittore del file e lo pubblica su un sito web
  - ricerca di un file = individuazione di un sito web che indicizzi i file .torrent
  - ogni sito web fornisce un servizio di indicizzazione dei .torrent memorizzati e può memorizzare milioni di .torrent
  - la ricerca di un file richiede come primo passo la ricerca del .torrent
- Descrittore di un file:
  - file statico con estensione .torrent generato mediante appositi programmi
  - in generale è un file di poche decine di Kbytes, codifica i dati secondo un formato indicato come bencoding.
  - contiene meta informazioni sul file condiviso come nome del file, lunghezza del file, set di valori hash, riferimenti ad un server (tracker) che memorizza i peer che stanno distribuendo il file

BITTORRENT: INDICIZZAZIONE DEI TORRENT



Filters: Language: English

Comments Negative Reports Positive Reports

Search results for **linux** in Software

Results **1** - **50** of **693**

Torrent Name	Category	Size	Date	Seed	Leech	Health
Gentoo Linux 2007.0 i686 Live CD Installer	6 6 Software	700MB	05/08	4666	X	
Gentoo Linux 2007.0 i686 Live DVD Installer	7 10 Software	3833MB	05/08	3475	X	
Linux Ubuntu 8.04 Desktop i386 Hardy Heron	33 4 Software	699MB	04/25	1946	128	
Kiwi Linux 8.04	4 Software	702MB	05/01	1885	4235	
Puppy Linux 2.17.1	3 Software	93MB	08/05	1000	1824	
Slackware Linux 12.1 i386 DVD	1 Software	3862MB	05/03	912	410	
Gentoo Linux 2007.0 livedvd-amd64-installer	Software	3742MB	05/15	736	61	
Damn Small Linux 3 4	2 Software	49MB	07/05	647	1318	
linux xp 2006 sr2	4 Software	620MB	11/17	590	777	
mandriva-linux-2008-one-KDE-cdrom-i586546836280032 54	4 Software	694MB	10/10	531	103	
Linux Ubuntu 8.04 Desktop AMD64 Hardy Heron	6 Software	697MB	04/25	485	22	
Mandriva Linux One 2008 Spring GNOME int cdrom i586	1 Software	645MB	05/02	469	91	
Linux Ubuntu-7.10-(Gutsy)-Desktop-i386	10 20 Software	696MB	10/18	457	21	
Mandriva Linux One 2008 Spring KDE Int Cdrom i586	2 Software	697MB	04/21	314	73	
Mandriva Linux 2008 Powerpack [multilang][x86]	5 3 Software	4384MB	10/10	298	408	
Linux Xubuntu 8.04 Desktop i386 Hardy Heron	1 Software	544MB	04/25	255	13	

BITTORRENT: IL FILE .TORRENT

Formato del file .torrent

Info Key

- Length : Lunghezza del file in bytes
- Name: nome del file
- Piece length: lunghezza dei pezzi in cui è stato suddiviso il file
- Pieces: viene calcolato un hash mediante SHA1 di 20 bytes per ogni pezzo in cui è stato decomposto il file e gli hash vengono concatenati in una unica stringa di bytes

Announce

- URL del tracker.

TRACKER PROTOCOL

- il tracker protocol utilizza HTTP/HTTPS
- l'host che ospita il tracker deve ospitare un server HTTP
- il peer D che intende effettuare il download invia al tracker alcune informazioni che lo identificano
- Formato della richiesta: GET + URL ricavata mediante il .torrent + alcune variabili CGI
- Formato della richiesta

  https: //some.url.com/announce?var1=value1&var2=value2&...
- Valore di alcune variabili:
  - Identificatore, IP+ porta di D.
  - Uploaded: quantità di dati inviati
  - Left: quantità di bytes che occorre ancora ricevere per completare il download

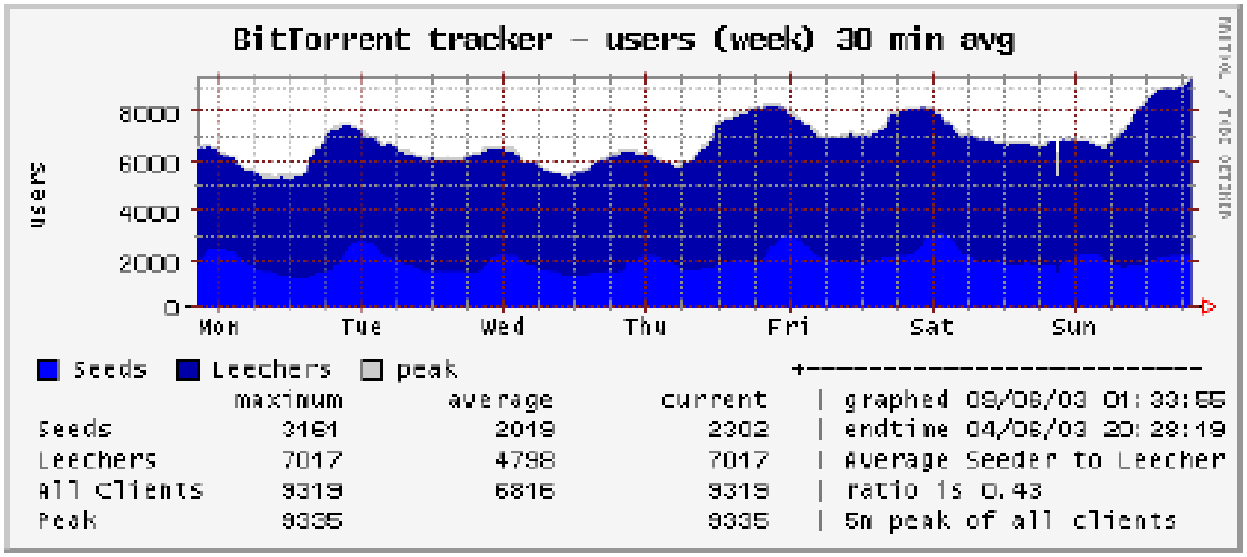
TRACKER PROTOCOL

- swarm: insieme di peers che collaborano alla distribuzione di uno stesso file e sono coordinati da uno stesso tracker
- il tracker conosce l'indirizzo di ogni peer partecipante allo swarm e consente quindi ad ogni peer di individuare altri peers nello stesso swarm
- il tracker invia ad un peer che lo ha contattato una lista di peers individuati in modo random. Per ogni peer viene indicato: identificatore logico + (IP e porta)
- mediante questa lista un nuovo peer P costruisce l' insieme iniziale dei peer conosciuti con cui scambierà pezzi di files
- a questo insieme successivamente vengono aggiunti nuovi peer che ottengono l'indirizzo di P dal tracker e si connettono direttamente a P
- topologia della rete = grafo random
- i peer contattano successivamente il tracker (ogni 30 minuti), ma unicamente per inviargli informazioni statistiche: esempio quantità dati inviati e ricevuti

BITTORRENT: SEEDER E LEECHERS

- Seeder (alla lettera inseminatore): peer che possiede tutte le parti del file
  - all'inizio un unico seeder, il peer che ha pubblicato il contenuto. Di solito questo peer rimane permanentemente sulla rete, ma può accadere che esca dalla rete dopo aver 'inseminato' un numero sufficiente di peer
  - in seguito si vengono a creare nuovi seeder. Sono i peer che hanno scaricato il file e rimangono all'interno della rete per altruismo o per caso
  - Crea il file .torrent e lo pubblica
- Leecher (alla lettera succhiatore):
  - peer che possiede qualche parte o nessuna parte del file e cerca di scaricare il file dai seeder e/o da altri leecher
  - All inizio ricerca il file .torrent per iniziare il download

BITTORRENT: IL TRACKER



- restituisce una lista di 50 seeders/leechers scelti in modo random
- ogni peer contatta un sottoinsieme di 20-40 peers
- peer Cache: IP, port, peerid
- informazioni sui peer:
  - completati
  - downloading
  - report status

BITTORRENT: SCAMBIO DEI PEZZI

- un peer
  - sceglie un sottoinsieme dei peer ricevuti dal tracker e stabilisce con ognuno di essi una connessione TCP
  - lascia spazio per alcune connessioni che possono venir aperte dal peer remoti
  - invia il proprio ID e l'hash SHA1 del il file ricavato dal .torrent
- il peer che riceve una richiesta di connessione effettua alcuni controlli ed in certi casi può decidere di rifiutare la connessione ad esempio
  - se l'hash non corrisponde a nessuno degli swarm a cui attualmente partecipa
  - se l'ID del peer non corrisponde a quelli ricevuti dal tracker
- una volta stabilita la connessione, ogni peer comunica a tutti gli altri gli indici dei pezzi dei files che possiede
- ogni peer in uno swarm conosce quindi la distribuzione dei pezzi all'interno dello swarm

BITTORRENT: ALGORITMI

- per la selezione delle parti dei files da scaricare
  - Stricy Priority
  - Random First Piece
  - Rarest First
  - End Game
- per la selezione dei peer con cui scambiare i dati
  - Chocking/ Unchocking (basati sulla teoria dei giochi)
  - Optimistic Unchocking

BITTORRENT: STRICT PRIORITY

- ogni pezzo viene decomposto in blocchi, tipicamente di dimensione 16kB
- ogni volta che si chiede un pezzo ad un peer le successive richieste riguarderanno i restanti blocchi di quel pezzo.
- finchè un intero pezzo non è stato scaricato completamente da un peer, da quel peer si scaricano solamente blocchi di quel pezzo
- in questo modo si favorisce l' 'assemblaggio veloce' dei pezzi
- in questo modo il peer può venire velocemente in possesso di pezzi completi che può scambiare con altri peer

BITTORRENT: RAREST PIECE FIRST

- Politica rarest piece first: ogni peer determina i pezzi più rari tra quelli posseduti tra i peer vicini e scarica per primi questi pezzi
- obiettivi di questa strategia:
  - se il seeder di un file si disconnette dallo swarm esiste il rischio che un pezzo raro risulti non più disponibile. Questo renderebbe impossibile la ricostruzione dell'intero file
  - se esistono pochi seeder, con capacità di upload limitata, questa strategia garantisce che nuovi downloaders scarichino nuove parti del file dai seeders, replicando così rapidamente l'intero file e diminuendo così il carico del download dal seeder
  - un peer acquisce pezzi rari molto richiesti da altri peer e viene scelto per il download da molti peer. In questo modo è più probabile che venga inserito nelle liste di upload di diversi peer.

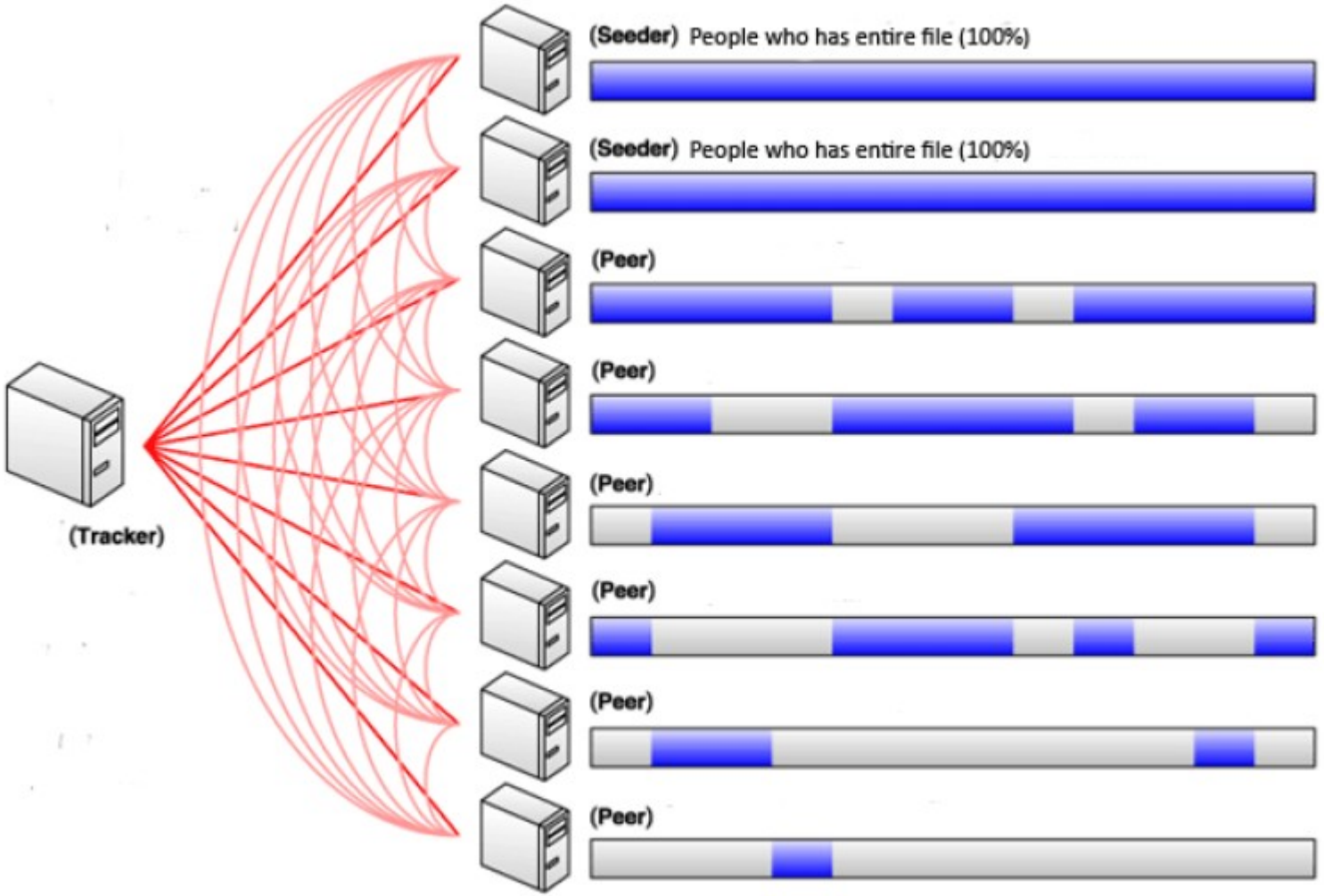
BITTORRENT: RANDOM FIRST PIECE

- esiste un'eccezione in cui non conviene applicare la politica rarest piece first.
- Politica Random First Piece: il primo pezzo da scaricare viene scelto in modo casuale, quindi si passa al rarest piece first.
- inizialmente un peer non possiede alcun pezzo e quindi non può offrire niente agli altri peer dello swarm.
- è importante che acquisica un pezzo prima possibile per poter iniziare a negoziare pezzi con altri peer
- i pezzi più rari
  - sono presenti su un solo peer o su un numero limitato di peer, per cui il loro download può risultare parecchio lento.
  - Minor scelta di peer = maggior probabilità di scegliere un peer lento
- Random First Piece= "partenza più veloce"

BITTORRENT:ENDGAME

- Osservazione: molti downloads tendono a rallentare quando si è vicini al completamento del download stesso (circa il 99%)
- questa situazione può essere causata da un peer S caratterizzato da una bassa banda di trasferimento:
- se il pezzo richiesto ad S è uno dei primi, questo può non incidere sul tempo di download dell'intero file,
- se il pezzo richiesto ad S è uno degli ultimi, S può ritardare il completamento del download dell'intero file
- End Game Policy
  - quando ad un peer rimangono pochi pezzi del file da scaricare questi pezzi vengono richiesti a tutti i peer, invece che ad un insieme random di essi
  - per evitare di sprecare banda, quando il pezzo richiesto viene ricevuto da un peer, gli altri download vengono annullati
  - modesto spreco di banda, in quanto l'end game viene eseguito per un breve periodo di tempo

BITTORRENT: FUNZIONAMENTO GENERALE



IL PROBLEMA DEI FREE RIDERS

- Free Rider = E' un individuo che nasconde le proprie preferenze per un bene, evitando di pagarne il prezzo e scaricando su altri individui il prezzo del bene stesso
- Se vi sono individui interessati ad un bene pubblico, il free rider si rende conto che potrà beneficiare ugualmente del bene, senza essere costretto a pagare.
- Esempio: un gruppo di studenti devono decidere se comprare una elettrodomestico per il l'appartamento comune.
  - qualcuno può nascondere di volere il televisore per evitare di pagarne il prezzo.
  - una volta acquistato, però, non risulta facile impedirgli l'utilizzazione del bene comune (proprio in quanto comune)
- Il Free Riding è un comportamento che prende il nome da colui che sale sull'autobus senza comprare il biglietto.

IL PROBLEMA DEI FREE RIDERS

- La causa della possibilità di Free Riders è la caratteristica di non escludibilità di un bene pubblico: è difficile escludere dal godimento di un bene pubblico chi non ne ha pagato il prezzo.
- Il problema viene identificato del free riding si può studiare mediante la teoria dei giochi
- Nel caso di content distibution P2P il free rider è un peer che utilizza la propria banda unicamente per il download e non ne riserva una parte per eventuali upload
- Studio del fenomeno dei free riders
  - Gnutella:
    - la maggioranza dei peer partecipanti alla rete Gnutella non condivide alcun file o ne condivide pochi
    - alcuni peer condividono solo file poco popolari comportandosi, di fatto,come dei free riders
  - Emule: il sistema dei crediti è un meccanismo troppo semplice per evitare il free riding

FREE RIDERS IN BITTORRENT

- Free riding in BitTorrent: peers che non offrono l'upload
  - presenza di banda asimmetrica di upload e download (ADSL)
  - a livello applicazione, esistono diversi client BitTorrent (non officiali) che consentono all'utente di limitare la banda di upload a proprio piacimento
- il buon funzionamento di BitTorrent dipende in larga parte dall' 'atteggiamento cooperativo' dei peer nei confronti della comunità, ovvero dalla eliminazione dei free riders
- È difficile imporre un certo comportamento al client Bittorrent, perchè è sempre possibile modificare il client mediante reverse engineering
- soluzione al problema del fenomeno dei free riders
  - implementazione di una strategia basata su risulati di teoria dei giochi (strategia Tit for Tat), implementata all'interno del protocollo stesso
  - Tit for Tat basato sulla reciprocità: ottiene un buon servizio solo chi fornisce un buon servizio

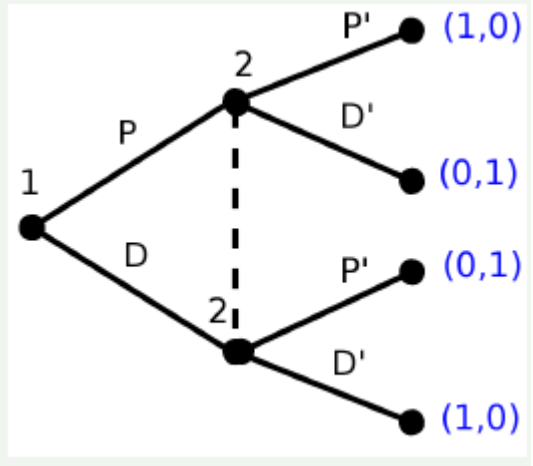
TEORIA DEI GIOCHI

- Branca della matematica che studia situazioni in cui i giocatori effettuano scelte tra diverse strategie per massimizzare i loro risultati
- Punto fondamentale: individui razionali che interagiscono tra di loro
- Applicata in economia, biologia, psicologia, sociologia, filosofia, ed informatica
- Nel nostro caso
  - Giocatori = peer
  - Strategia = Comportamenti adottati dai peer quando interagiscono

TEORIA DEI GIOCHI

Esempio: Pari o Dispari

- due giocatori, 1 e 2
- entrambi scelgono contemporaneamente un numero
- se la somma è pari, vince 1. Altrimenti, vince 2
- Pay off = in teoria dei giochi, il pay off è la fase conclusiva del gioco. Può essere reso in italiano come: risultato, premio, ricompensa, pagamento.



TEORIA DEI GIOCHI

Forma strategica

1 \ 2	P	D
P	1,0	0,1
D	0,1	1,0

- Rappresentazione tabellare del gioco
- Comoda se i giocatori sono due
- I payoff si ottengono cercando riga e colonna indicati dalle due strategie corrispondenti ai due giocatori
- Scelta razionale: i due giocatori vogliono massimizzare il loro payoff

TEORIA DEI GIOCHI

Date due strategie A, B per il giocatore X, se qualsiasi sia la scelta di strategie per gli altri giocatori, i payoff ottenuti scegliendo A sono maggiori di quelli ottenuti scegliendo B, si dice che

A domina fortemente B

Esempio

1 \ 2	D	E	F
A	2,0	4,1	5,2
B	3,1	3,2	2,1
C	1,4	2,2	0,3

Esempio: A domina fortemente C, E domina fortemente D

IL DILEMMA DEL PRIGIONIERO

- Famoso esempio di teoria dei giochi
- A e B vengono arrestati dalla polizia
  - vengono rinchiusi in celle separate, non possono comunicare
  - vengono interrogati separatamente e ognuno può
    - resistere all'interrogatorio e proclamarsi innocente
    - tradire l'altro e confessare il crimine
- A seconda del loro comportamento, la polizia si comporta nel modo seguente
  - se entrambi resistono (cooperano) vengono incarcerati per 2 anni per reati minori
  - se A confessa (tradisce) e B no, il traditore (A) viene liberato e l'altro (B) resta al fresco per 5 anni
  - se entrambi tradiscono, vengono entrambi incarcerati per quattro anni
- A e B si chiedono: quale è la strategia migliore? Quale porta ad una pena minore?

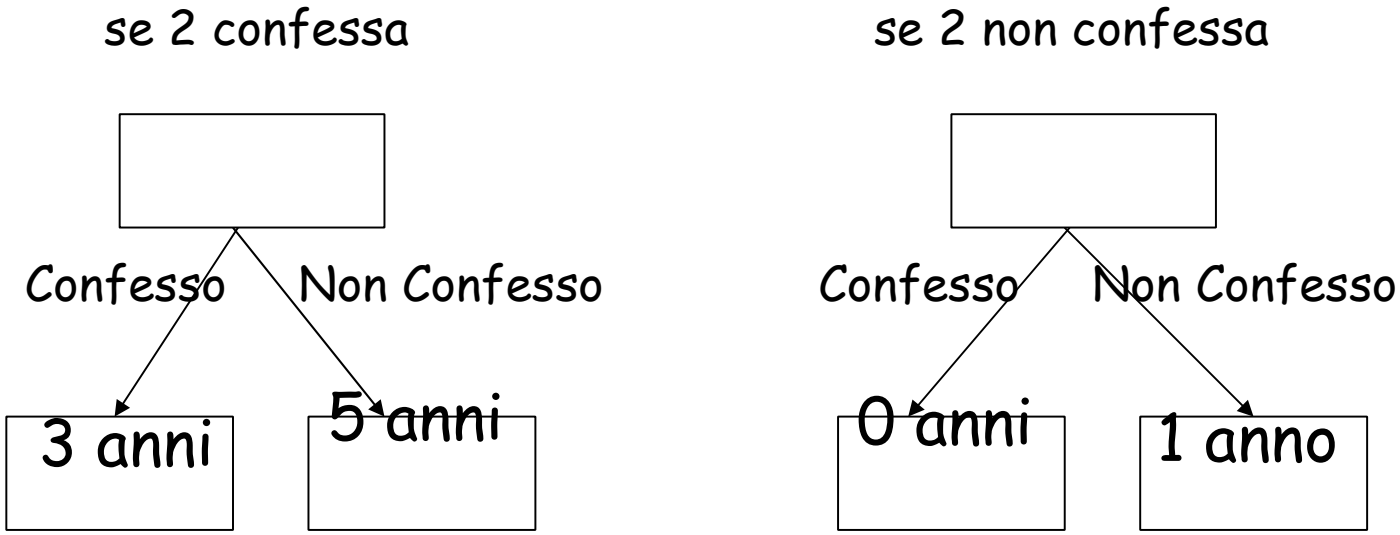
LA FORMA STRATEGICA

1 \ 2	C	D
C	3,3	0,5
D	5,0	1,1

- per entrambe i giocatori:
  - D = Defezione, tradimento,
  - C = Cooperazione
- payoff = numero di anni risparmiati
- per entrambe i giocatori la strategia D domina C
- risultato: Ogni giocatore ottiene il beneficio maggiore mediante la defezione, piuttosto che mediante la collaborazione

ALBERO DI DECISIONE

Consideriamo il punto di vista di 1

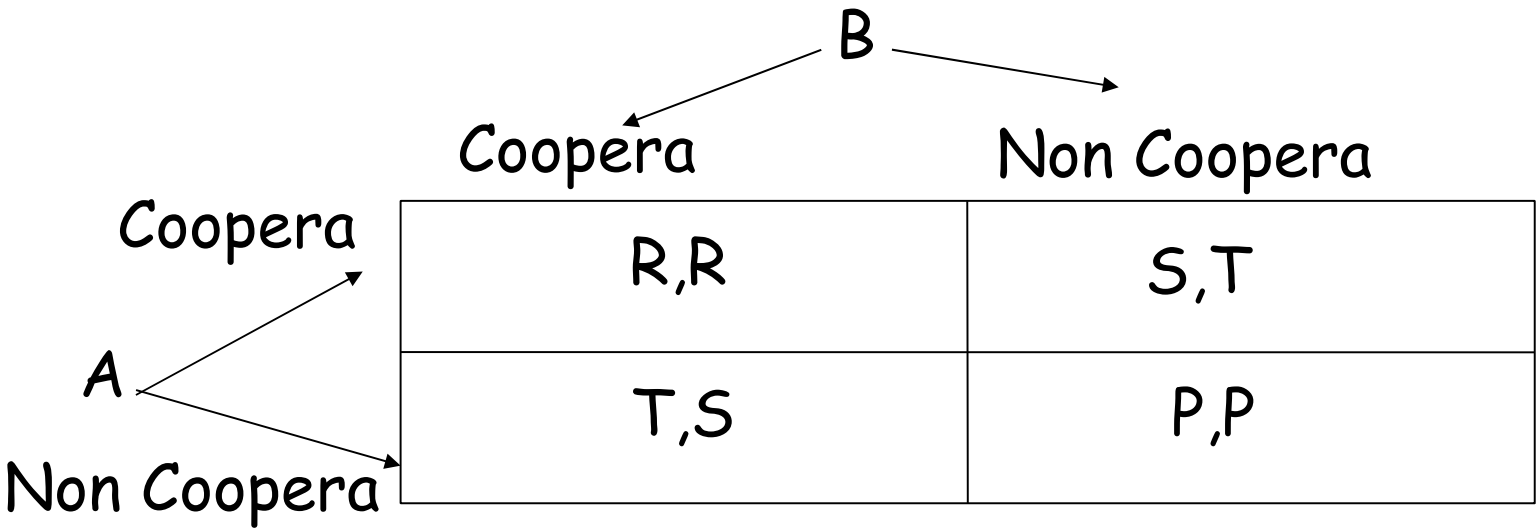


- qualsiasi sia la decisione di 2, confessare è la strategia migliore per 2
- la strategia migliore per il singolo giocatore è confessare, ovvero tradire il compagno
- il guadagno complessivo è però minore di quello che si avrebbe dalla mutua cooperazione.

VERSIONI ALTERNATIVE

- Versione alternativa del dilemma del prigioniero: scambi commerciali tra due persone
- Due commercianti vogliono scambiarsi due oggetti
- Ognuno di essi deve decidere se inviare oppure non inviare il proprio oggetto (merce) all'altro senza conoscere la decisione dell'altro
- Se entrambi decidono di inviare il proprio oggetto, entrambi beneficiano dell'oggetto ricevuto dall'altro
- Se uno invia e l'altro no, quello che riceve dall'altro l'oggetto possederà entrambe gli oggetti
- Se entrambi non inviano , nessuno perde niente, ma nessuno beneficia dell'oggetto ricevuto
- Cooperazione = Invio della merce
- Defezione = Mancato invio della merce

IL DILEMMA DEL PRIGIONIERO: FORMA GENERALE



- R = Reward, P= Punishment, T= Temptation, S = Sucker
- Condizione per il Dilemma del Prigioniero:

$$T>R>P>S$$

$$5>3>1>0$$

DILEMMA DEL PRIGIONIERO ITERATO (IPD)

- Consideriamo una sequenza di round in cui ad ogni round si affronta il problema del prigioniero
- Supponiamo inoltre di non conoscere il numero di round a priori (non si conosce quando il gioco ha termine)
- Quale è la strategia migliore, in questo caso?
- Strategia tit-for-tat (alla lettere, pan per focaccia, botta e risposta)
  - cooperare sempre al primo turno
  - quindi replicare la mossa effettuata dall'avversario al passo precedente
- Il tit-for-tat
  - si è rilevata la strategia migliore in diversi tornei a cui hanno partecipato esperti di teoria dei giochi [Axelrod]
  - è alla base dell'algoritmo di chocking definito da BitTorrent

DILEMMA DEL PRIGIONIERO ITERATO (IPD)

- Axelrod verifica che la collaborazione 'paga' nel caso del dilemma del prigioniero iterato
- Ogni giocatore è in grado di ricordare il comportamento dell'avversario nell'ultima sessione di gioco
- I giocatori accumulano i payoff corrsipondenti alla matrice dei payoff corrispondente al dilemma del prigioniero
- Vince chi ha accumulato il valore massimo dei payoff
- La strategia del Tit for Tat se pur molto semplice, si è rilevata migliore di altre strategie molto più complesse
- Il Tit for Tat non riesce a sconfiggere un singolo oppositore, perchè collabora sempre come prima mossa, ma riesce ugualmente a vincere i tornei
- Non sconfigge gli avversari singolarmente, ma risulta globalmente vincente

TIT FOR TAT: UN ESEMPIO

- Consideriamo 4 giocatori, due di loro (TFT) utilizzano la strategia TitForTat, gli altri due (A) cercano sempre di ottenere il massimo vantaggio, accusando ad ogni turno l'altro giocatore.
- Ad ogni round del gioco, ogni giocatore gioca contro tutti gli altri ed il gioco prevede 6 rounds
- La matrice delle ricompense è la seguente

	cooperazione	accusa
cooperazione	3,3	0,5
accusa	5,0	1,1

TIT FOR TAT: UN ESEMPIO

- Quando un TFT gioca contro un A, nel primo turno il primo acquisice 0 punti, l'A acquisice 5 punti. Nei rimanenti 5 rounds, entrambi si accusano e quindi acquisicono 1 punto a testa. Punteggio finale TFT- 5 A-10
- Quando i due TFT si affrontano cooperano al primo round e poi a tutti i round successivi TFT(1)-18 TFT(2)-18
- Quando i due A si affrontano si accusano a vicenda in tutti i 6 rounds. Punteggio finale A(1)-6 A(2)-6
- Punteggi finali:
  - per ogni TFT 5+5+18 = 28
  - per ogni A 10+10+6= 26
- La strategia vincente risulta essere il Tit-for-Tat !
- I TFT non vincono i singoli round e gli A non perdono i singoli round, ma alla fine vincono i TFT, perchè il punteggio finale non è determinato dai singoli vincitori, ma dai punti accumulati

TIT FOR TAT: CARATTERISTICHE

- Cooperativa: si collabora sempre alla prima mossa. In BitTorrent vedremo che questo corrisponde a fornire sempre dei dati, come prima mossa
- Contraccambio: si smette di collaborare se l'alto lo fa. In Bittorrent questo significa interrompere l'upload dei dati se l'altro non ci invia dati oppure invia dati ad una velocità bassa
- Perdono: si ritorna a collaborare se il partner collabora, anche se non aveva collaborato in precedenza. In BitTorrent un peer riprende l'upload dei dati quando un peer gli fornisce di nuovo dei dati
- Non invidiosa: non si tenta di far meglio degli altri

CONDIZIONI PER IPD

Ulteriore condizione necessaria per il problema iterato del prigioniero

$$R > (S+T)/2$$

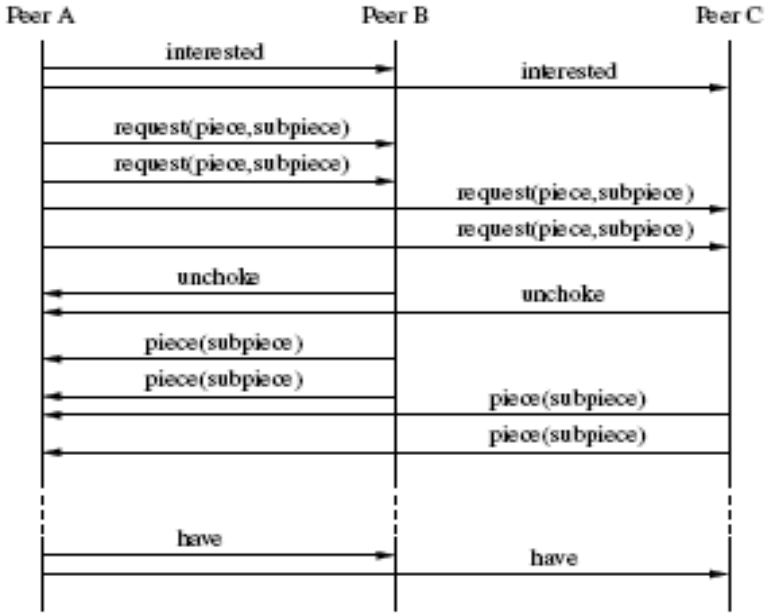
- si vuole evitare che i giocatori risolvano il dilemma semplicemente accusandosi a vicenda, a turno.
- supponiamo di giocare per k turni
- si vuole premiare la mutua cooperazione pittosto che l'alternanza di reciproche accuse.
- R ricompensa media su k mosse per aver collaborato,
- (S+T)/2 ricompensa media se si sono alternate collaborazioni ed accuse

BITTORRENT E TIT FOR TAT

- modellare lo scambio di dati come un dilemma del prigioniero iterato
  - Cooperare = fornire banda sufficiente per l' upload
  - Defezionare = 'soffocare' l'upload
- utilizzare una strategia basata su Tit for Tat per promuovere la collaborazione e diminuire il problema dei free riders
  - la prima mossa di un peer è sempre un upload (cooperazione)
    - equivale alla prima mossa ottimistica del tit for tat
  - successivamente un peer effettua l'upload solo verso i peer che gli hanno inviato dati con velocità ragionavole (hanno collaborato)

SCAMBIO DI PEZZI TRA PEERS

- Il peer downloader D esprime il suo interesse per un pezzo del file inviando degli interested messages ai peer uploader U che possiedono i pezzi
- U può decidere se effettuare l'upload o meno a seconda che la connessione con quel peer sia stata 'soffocata' o meno



BITTORRENT: CHOKING

- Ogni peer apre una connessione TCP (connessione bidirezionale) verso un insieme di altri peer i cui indirizzi sono stati ricevuti dal tracker
- ogni connessione si può trovare, in uno dei due seguenti stati
  - unchocked: la connessione è utilizzata per l'upload dei dati
  - chocked (soffocata): la connessione viene 'soffocata' cioè non viene utilizzata per il trasferimento dei dati, anche se viene mantenuta aperta
- Algoritmo di chocking:
  - ogni peer mantiene un numero limitato di vicini unchoked (in genere 4)
  - periodicamente valuta, per ogni vicino, la velocità di download da quel vicino nel round precedente e decide quali vicini 'strozzare'
- Chocking = rifiuto (temporaneo) ad effettuare l'upload, può essere revocato
- Le connessioni vengono comunque lasciate aperte, per non pagare più di una volte il costo dell'apertura della connessione

BITTORRENT: CHOCKING

- al massimo 4 unchocked peers, in ogni istante
- algoritmo di chocking eseguito da un leecher (come specificato nella prima versione del protocollo)
  - regular unchoking: ogni 10 secondi: ordinare i peer remoti che hanno richiesto qualche pezzo, secondo la loro velocità di download nell'ultimo periodo ed effettuare l'unchoke dei primi tre peer
    - 10 secondi in termine di connessione TCP sono un tempo relativamente lungo per permettere il pieno sfuttamento del canale di comunicazione
  - optimistic unchocking: Ogni 30 secondi, scegliere un ulteriore peer, in modo casuale ed effettuarne l'unchocking (corrisponde alla prima mossa del dilemma dl prigioniero)
- Optimistic Unchoking: effettua l'upload su una connessione non utilizzata nell'ultimo periodo di tempo per provare se tale connessione può fornire una migliore velocità di download

BITTORRENT:ANTI SNUBBING

- optimistic unchocking:
  - consente di valutare la capacità di download di nuovi peer
  - permette il bootstrap di nuovi peer sulla rete che non hanno alcun pezzo da offrire (per questo è importante all'inizio chiedere un pezzo non raro al maggior numero di peer possibile)
- algoritmo di chocking eseguito da un seeder S
  - non può ordinare i peer in base alla velocità di dowload verso quei peer, perchè non scarica nulla dagli altri peer
  - ordina i peer interessati in base alla loro velocità di download da S
  - favorisce i peer con un'alta velocità di download, indipendentemente dal loro contributo allo swarm
  - modificato nell'ultima versione di BitTorrent

TRACKERLESS BITTORRENT

- Il tracker centralizzato coordin allo swarm, cioè l'insieme dei peer interessati nello scaricamento di un singolo
- Ogni peer interessato notifica la propria presenza al tracker mediante un annuncio e riceve dal tracker un sottoinsieme dei peer dello swarm
- Il tracker
  - costituisce un punto di centralizzazione che può diventare un collo di bottiglia
  - single point of failure
- Eliminazione del singolo punto di centralizzazione
  - definizione di più tracker per ogni swarm
  - utilizzo di distributed hash tables (Kademlia)

TRACKERLESS BITTORRENT

Soluzione basate su DHT (Kademlia)

- Ogni peer esegue anche le funzionalità di tracker
- Ad ogni peer viene associato un identificatore di 160 bit e nello spazio degli identificatori vengono mappati i descrittori dei files (InfoHash)
- Per ogni file
  - Chiave = InfoHash del file
  - Informazione = Lista di peers appartenenti allo swarm

Soluzioni basate su MultiTrackers

- Definisce una lista di tracker per ogni torrent
- I tracker interagiscono periodicamente scambiandosi informazioni