



# Università degli studi di Bari

Corso di Laurea Magistrale in Informatica

Sistemi Distribuiti: architettura e modellazione

Modulo B – modellazione

Anno Accademico 2008-2009

## Modellazione del protocollo BitTorrent attraverso il formalismo delle Abstract State Machines(ASM)



Docente

Alessandro Bianchi

Studente

Vito Astone



# Il protocollo BitTorrent



## Che cosa è

- Un protocollo peer-to-peer per il file sharing;
- E' il nome del client ufficiale;
- Scritto da Bram Cohen;
- Implementato in linguaggio Python;
- Pensato per file di grossa dimensione (ISO, DVD, ...).



## I Componenti

- .torrent*: un file che contiene le informazioni necessarie per scaricare il file voluto;
- tracker*: coordina l'attività dei peer;
- peer*: è sia colui che scarica un dato file e sia colui che permette di scaricarlo;
- qualcuno che ci fornisce il file .torrent*: webserver, posta elettronica,...



## I componenti

---

### .torrent

- Nome del file da scaricare;
- Dimensione del file;
- L'hash di ciascuna parte in cui è stato diviso il file. In questo modo abbiamo la certezza di scaricare la parti di file corretti;
- La dimensione delle parti;
- L'indirizzo del tracker.



## I componenti

---

### tracker

- Coordina l'attività dei peer;
- Conosce l'indirizzo di ogni peer;
- Di ogni peer conosce lo stato di download;
- Per poter assistere meglio i peer è anche informato su quali parti del file siano possedute da ciascun peer;
- Viene interrogato periodicamente.



## I componenti

---

### peer

- Esistono due differenti tipi di peer
  - Seeder;
  - Leecher.



## I componenti

---

### seeder

- E' detto seeder quel peer che possiede una intera copia del file;
- E' necessario che almeno inizialmente si sia almeno un seeder che possa insemminare la rete;
- In determinate circostanze, ci potrebbe non essere alcun seeder ma, malgrado ciò, sia ancora possibile recuperare tutte le parti necessarie per mettere insieme l'intero file. In questo caso si parla di copia distribuita.



## I componenti

### leecher

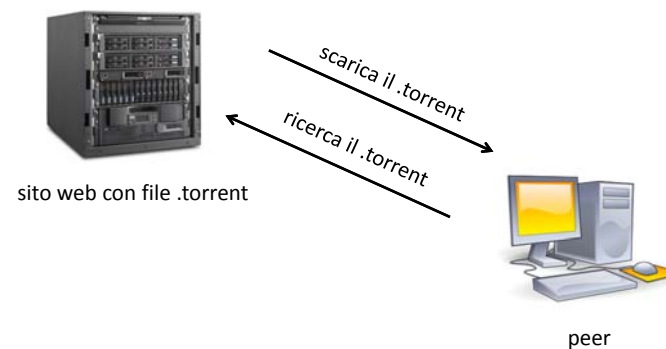
- Un leecher è un peer che non ha ancora una copia completa dei un particolare file;
- Quando dei nuovi peer cominciano a scaricare diventano leecher finchè non avranno una copia completa diventando in tal modo seeder;
- In bittorrent un leecher ha un ruolo attivo, in quanto permette l'upload;
- E' interesse del leecher garantire banda per l'upload, in quanto la sua velocità di download dipende dalla sua velocità di upload: più si da più si riceve.



## Come funziona

### Primo passo

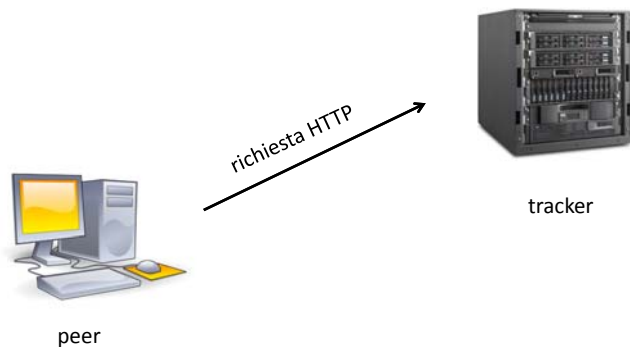
- Chi desidera scaricare un file deve innanzitutto ottenere il file .torrent relativo al file;
- Successivamente deve aprire il file .torrent con un client BitTorrent.



## Come funziona

### Secondo passo

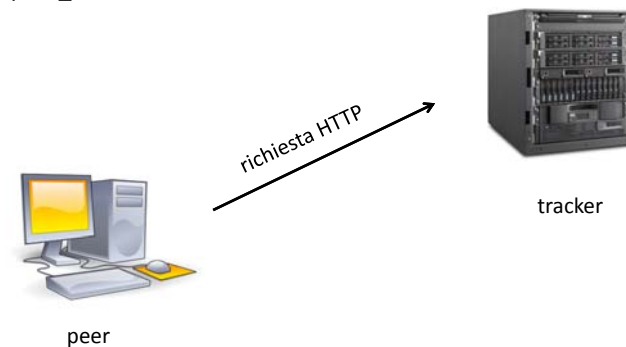
- Si avvia il client BitTorrent;
- Viene contattato il tracker indicato dal .torrent;
- Lo scambio di messaggi avviene attraverso HTTP request e response.



## Come funziona

### Secondo passo

- Il peer comunica al tracker
  - Quale file vuole scaricare;
  - In quale porta si attende di ricevere la lista dei peer da cui scaricare;
  - Un peer\_id che lo identifica univocamente.





## Come funziona

### Secondo passo

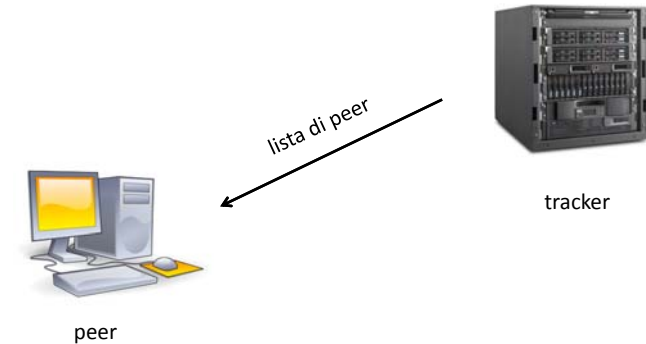
- ❑ Altri parametri che vengono inviati al tracker
  - *info\_hash*: viene utilizzato SHA1;
  - *port*: numero di porta sulla quale il client è in ascolto. Le porte tipiche sono nel range 6881-6900;
  - *uploaded*: bytes inviati dal quel client agli altri client dall'inizio della sessione;
  - *downloaded*: bytes scaricati dagli altri client dall'inizio della sessione;
  - *left*: bytes rimanenti al completamento del file.



## Come funziona

### Secondo passo

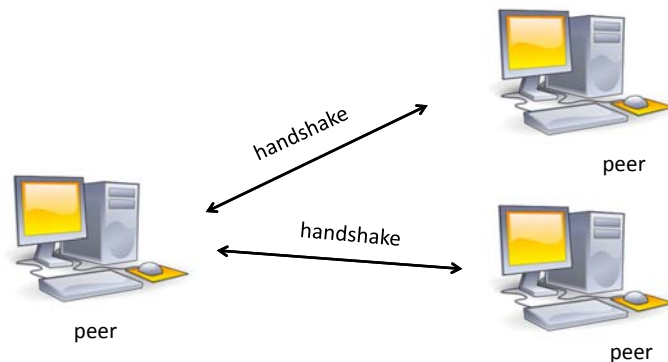
- ❑ Il tracker restituisce una lista di peer da cui è possibile scaricare.



## Come funziona

### Terzo passo

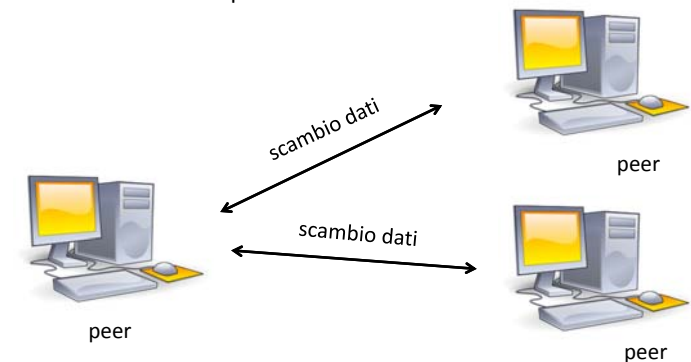
- ❑ Viene istanziata una connessione con i peer presenti nella lista fornita dal tracker;
- ❑ I messaggi vengono scambiati attraverso il TCP socket;
- ❑ Fase di handshake.



## Come funziona

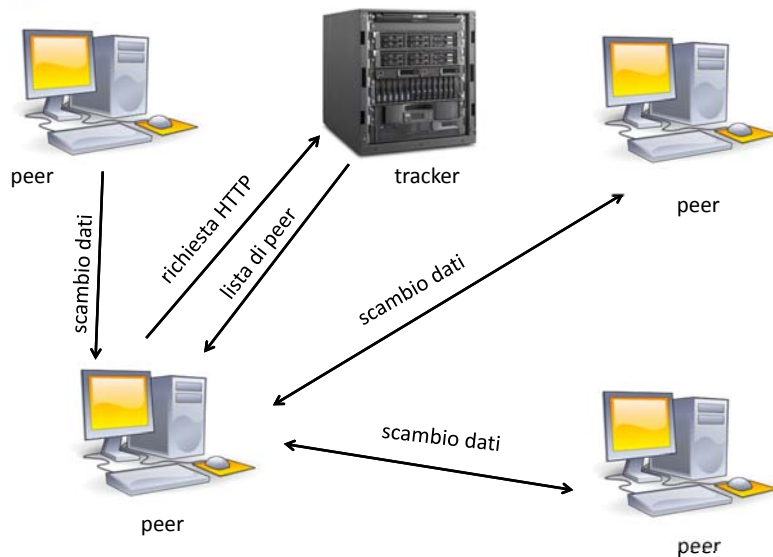
### Quarto passo

- ❑ Ogni peer informa gli altri di quali parti di file ha a disposizione;
- ❑ Un peer annuncia il possesso di una parte di file solo dopo averne verificato l'integrità con l'algoritmo SHA1;
- ❑ Lo scambio di dati è reciproco.





## Come funziona



## Selezione delle parti

Per poter ottenere una buona performance del sistema, bittorrent applica alcuni criteri di selezione delle parti di file da scaricare

### Strict priority

Una volta che un blocco viene richiesto ad un peer, tutte le future richieste a quel peer riguarderanno i restanti blocchi della parte. Questo permette di completare velocemente parti intere del file.

### Rarest first

Quando il peer seleziona quale parte scaricare sceglie sempre quella con minore occorrenza. Questo criterio permette al peer di essere concorrenziale, possedendo delle parti di file molto richieste. Questo metodo risulta particolarmente utile soprattutto nel momento dell'introduzione di un nuovo torrent, alleggerendo di molto il carico sul seeder originario.



## Selezione delle parti

### Random first

L'unica eccezione alla regola del rarest first accade quando si inizia il download di un file. E' importante che un lecheer inizi ad uplodare le parti che possiede in quanto la sua velocità di download dipende dalla sua velocità di upload. Per questa ragione la prima parte da scaricare viene scelta in modo casuale.

### Endgame mode

Risulta molto utile quando si è prossimi a terminare il download del file. Viene inviata una richiesta per le parti mancanti ad ogni peer. Quando una parte arriva deve essere inviata una richiesta in cui si esprime che quella parte non è più di nostro interesse.



## Algoritmo choking

Ogni peer ha anche il compito di scegliere se respingere (choke) o accettare (unchoke) una richiesta di upload. In realtà l'atto del choking è un temporaneo rifiuto dell'upload.

Queste decisioni avvengono in base ad alcuni criteri:

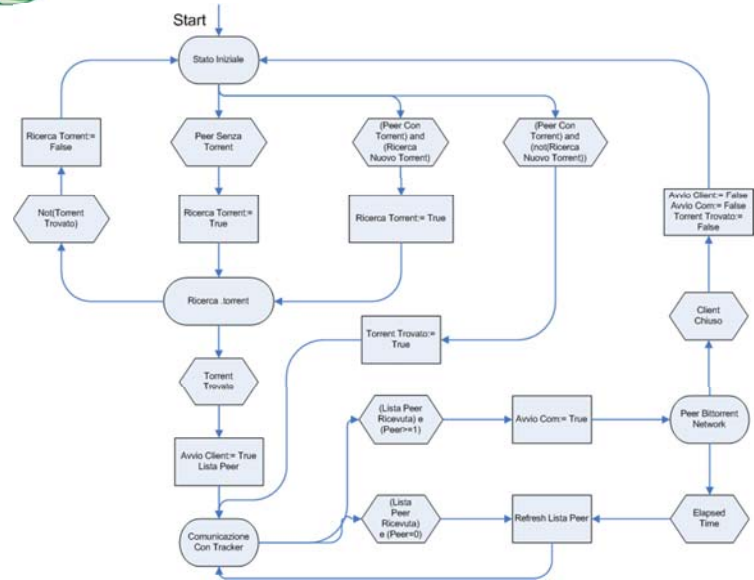
- Un peer manda dati a peer che ne hanno mandati a lui (tit-for-tat);
- Il peer serve al più 4 peer, privilegiando quelli che scaricano a una velocità maggiore;
- Ogni 10 secondi si calcola la lista dei peer da soffocare (choke) e quelli con cui cooperare (unchoking);
- Ogni 30 secondi il peer abilita un nuovo peer fra quelli in choke, a prescindere dalle sue prestazioni.



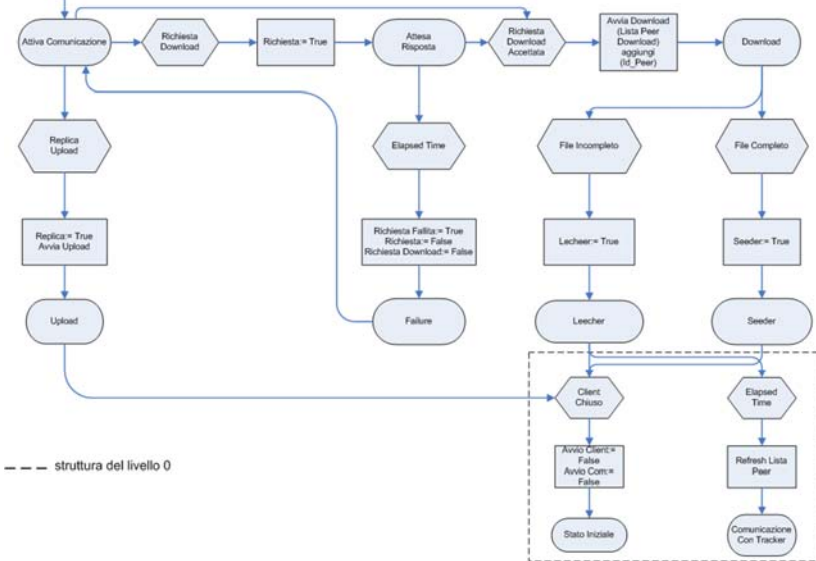
## Modellizzazione del protocollo BitTorrent con le ASM



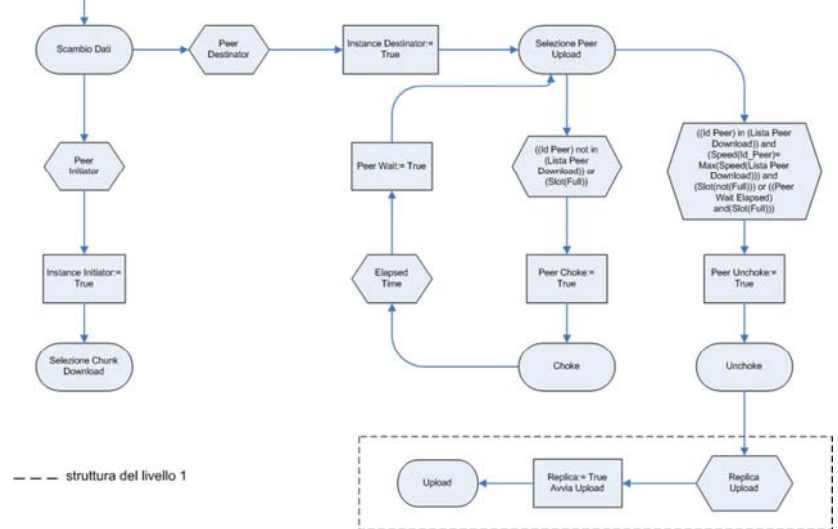
## Ground Model – Livello 0



## Peer BitTorrent Network – livello 1

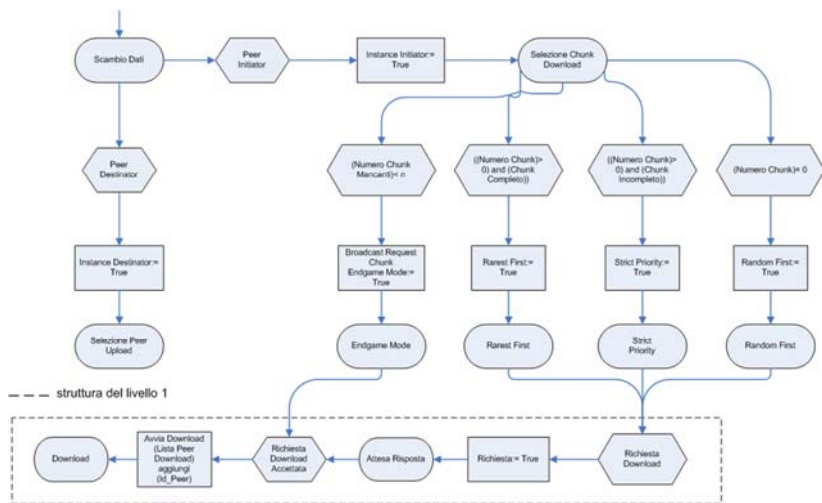


## Attiva Comunicazione - 1 – Livello 2





## Attiva Comunicazione - 2 – Livello 2



## Tabelle degli stati - Ground Model – Livello 0

Stato Iniziale	
Torrent Trovato	False
Ricerca Torrent	False
Ricerca Nuovo Torrent	True, False o Indefinito
Avvio Client	False
Lista Peer	Vuota

Ricerca Torrent	
Torrent Trovato	False
Ricerca Torrent	True
Ricerca Nuovo Torrent	True

Comunicazione Con Tracker	
Torrent Trovato	True
Avvio Client	True
Lista Peer	Vuota

Peer BitTorrent Network	
Avvia Com	True
Client Chiuso	False
Elapsed Time	False
Avvio Client	True
Lista Peer	>=1