



Università
degli Studi
della Campania
Luigi Vanvitelli

Reti di Calcolatori e Cybersecurity

P2P – Peer to Peer

Ing. Vincenzo Abate

P2P

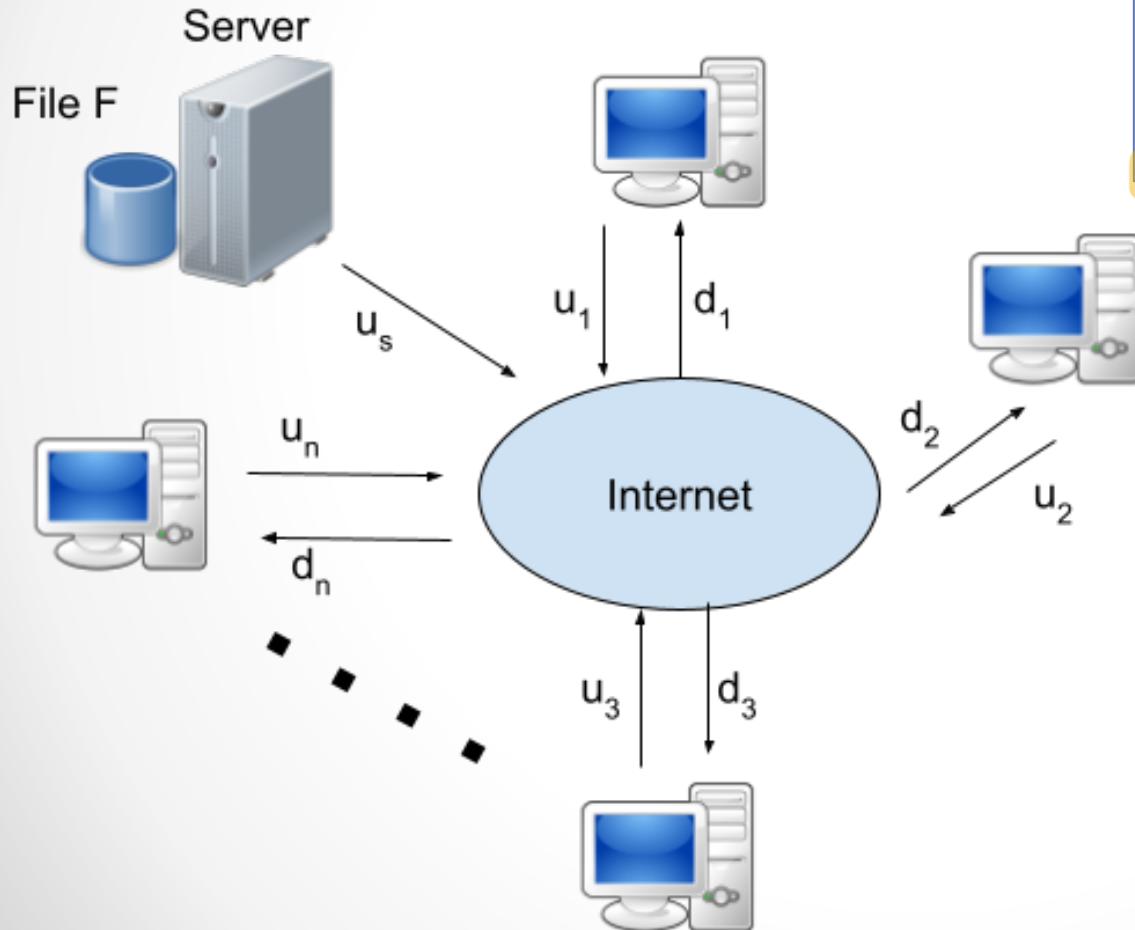
- P2P: un modello alternativo al client/server
- Non esistono server sempre connessi (always-on server)
- Non esiste una differenziazione funzionale tra client e server
- End-system (peer) comunicano direttamente
- I peer sono connessi ad intermittenza e cambiano il proprio IP

Peer: quale standard che cambia indirizzo: provider da' indirizzo diverse. Come risolve?

Quali aspetti va a smussare?

Client-Server vs P2P

- Quanto tempo serve per distribuire un file da un server ad N peer (tempo di distribuzione, D)?



u_s : server upload bandwidth
 u_i : peer i upload bandwidth
 d_i : peer i download bandwidth

Server custodisce file F

N, numero di peer che richiedono una copia di F

Client-Server

Il server invia N copie del file

Il fattore limitante può essere u_s o d_i

Se il fattore limitante è u_s allora il tempo richiesto è NF/u_s

Se il fattore limitante è $\min(d_i)$ allora il tempo richiesto per completare il trasferimento è il tempo di download del client "più lento" pari a $F/\min(d_i)$

2 fattori limitanti: lato server o lato client
→ altri parametri trascurabili

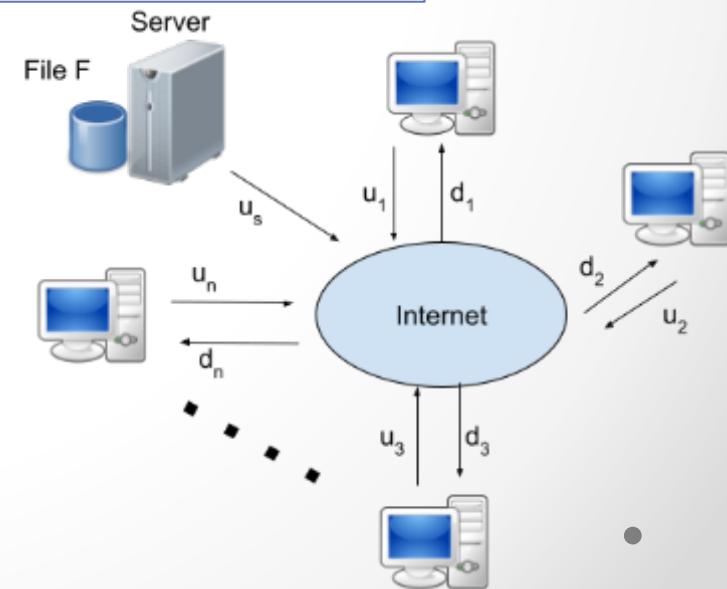
Tempo per distribuire

$$F \text{ bit a } N \text{ client} = D_{cs} = \max \{ NF/u_s, F/\min(d_i) \}$$



per N elevato, il termine NF/u_s è dominante:
incremento lineare al crescere di N

Tempo passato nel quale tutti hanno il file



File all'interno ce l'ha un peer. Quanto al chiede la risorsa diretta anche l'altro peer della rete.
Presta attenzione da quanti peer sono coinvolti.

P2P

Dipende dal numero di peer coinvolti ma anche dall'ordine con cui il contenuto è trasferito ai peer...

Il server deve inviare almeno una copia. Tempo richiesto: F/u_s

Il peer i impiega un tempo pari a F/d_i più scaricarlo.

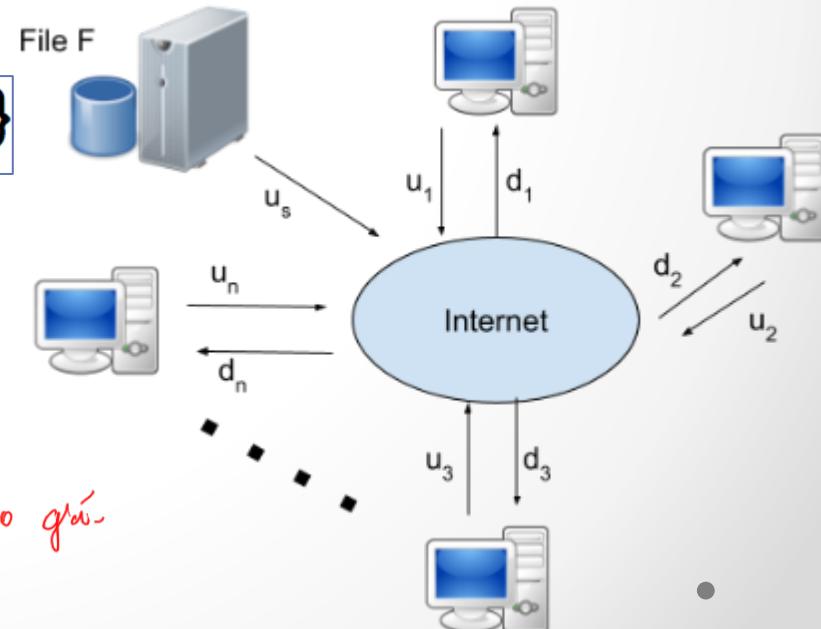
Il più lento impiega (F/d_{\min})

NF bit totali devono essere 'scaricati': velocità massima di upload: $u_s + \sum u_i$ → Se nel file è passato a tutti, diventano tutti server

$$D_{P2P} = \max \{ F/u_s, F/\min(d_i), NF/(u_s + \sum_i u_i) \}$$

N volte dimensione del file / velocità di upload di tutti

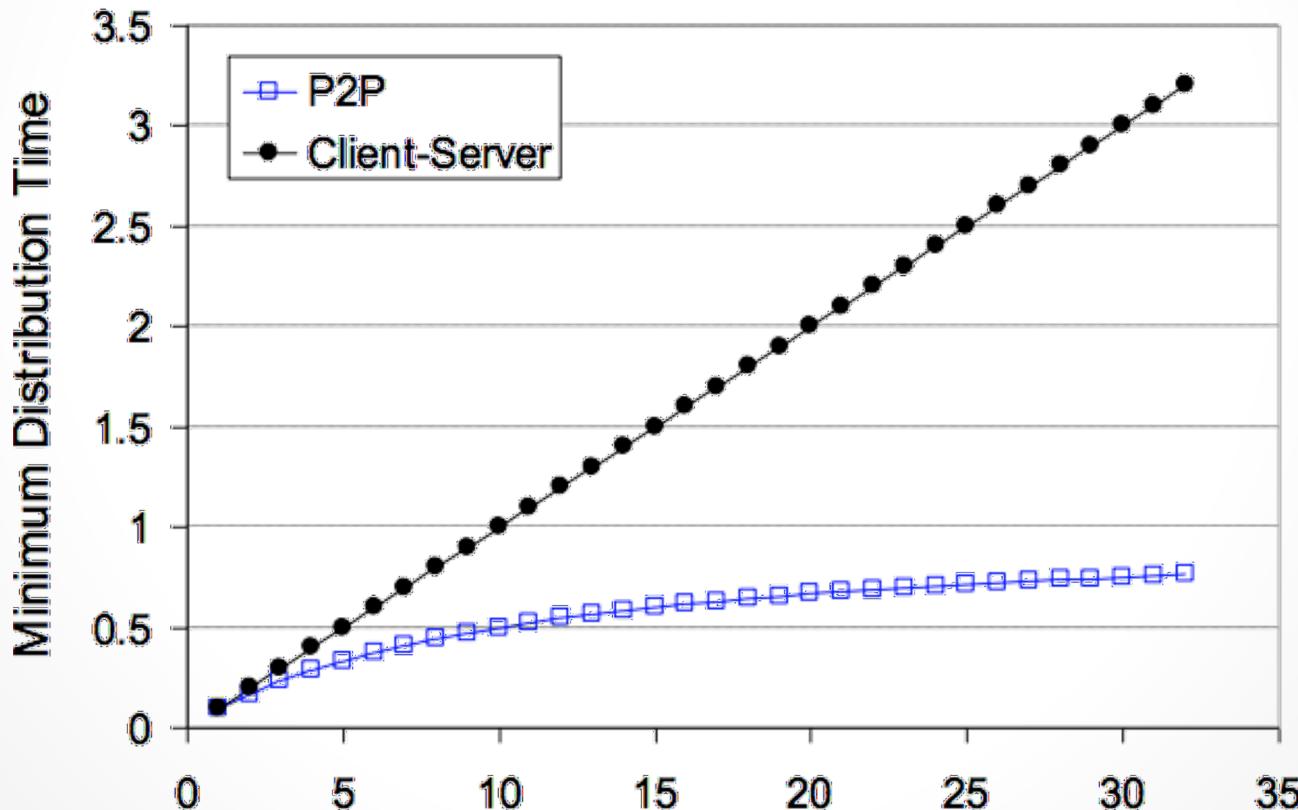
gli n che ce l'hanno già



Client-Server vs P2p

Nel caso in cui:

- Tutti i peer hanno lo stesso rate di upload = u
- $F/u = 1$ ora, $u_s = 10u$, $d_{\min} \geq u_s$



Al crescere di N , cresce $NF / (\mu_s + \sum \mu_i)$, dunque kN songhi del file. Io lo passo a A. A lo passa a B. A e B a C e D.

P2P file sharing

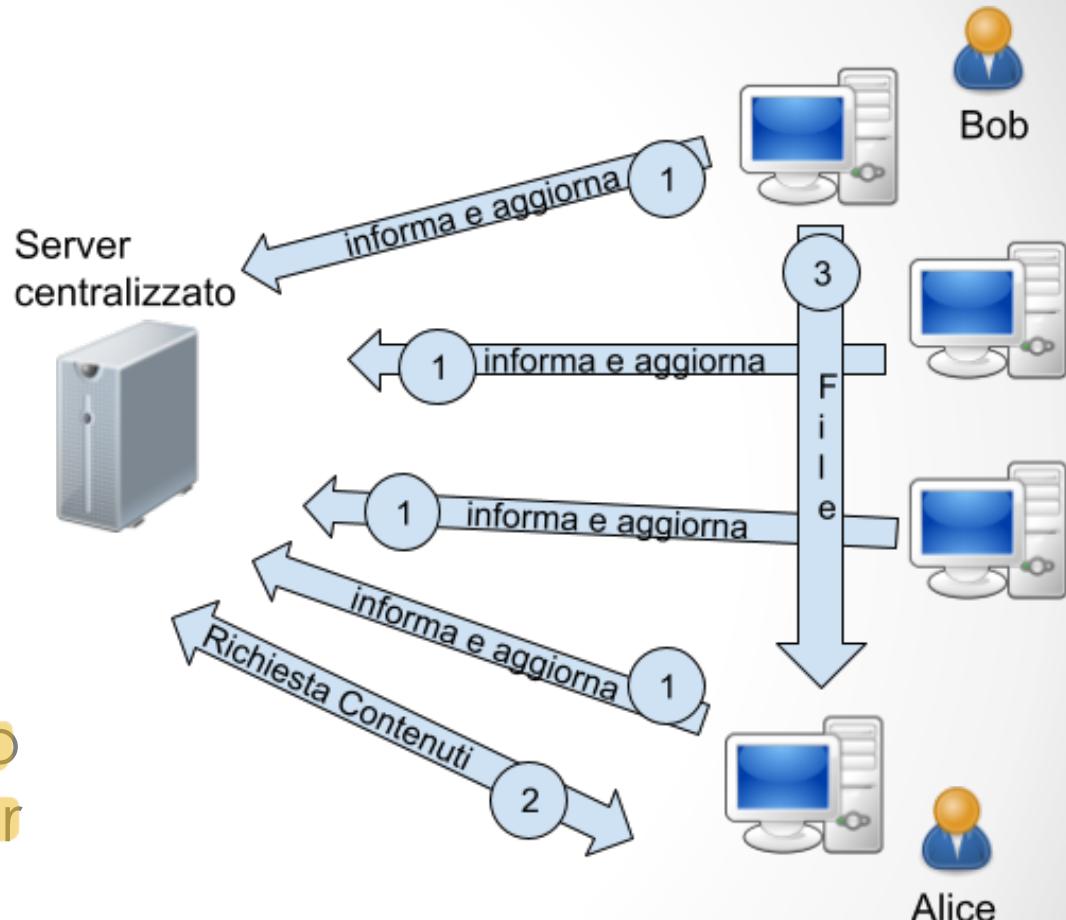
- Alice è connessa ad Internet e ha lanciato la sua applicazione di file sharing P2P sul suo PC
- Alice non è perennemente connessa ad Internet, non ha un hostname, l'IP cambia ad ogni connessione
- Alice chiede "Hey Jude"
- L'applicazione mostra tutti gli altri peer che hanno una copia di Hey Jude
- Alice sceglie uno dei peer, Bob
- Il file è trasferito (copiato) dal PC di Bob a quello di Alice usando HTTP
- Mentre Alice effettua il download, altri peer possono prendere file da Alice. Alice è quindi sia un client che un server

Tutti i peer sono anche server: architettura altamente scalabile



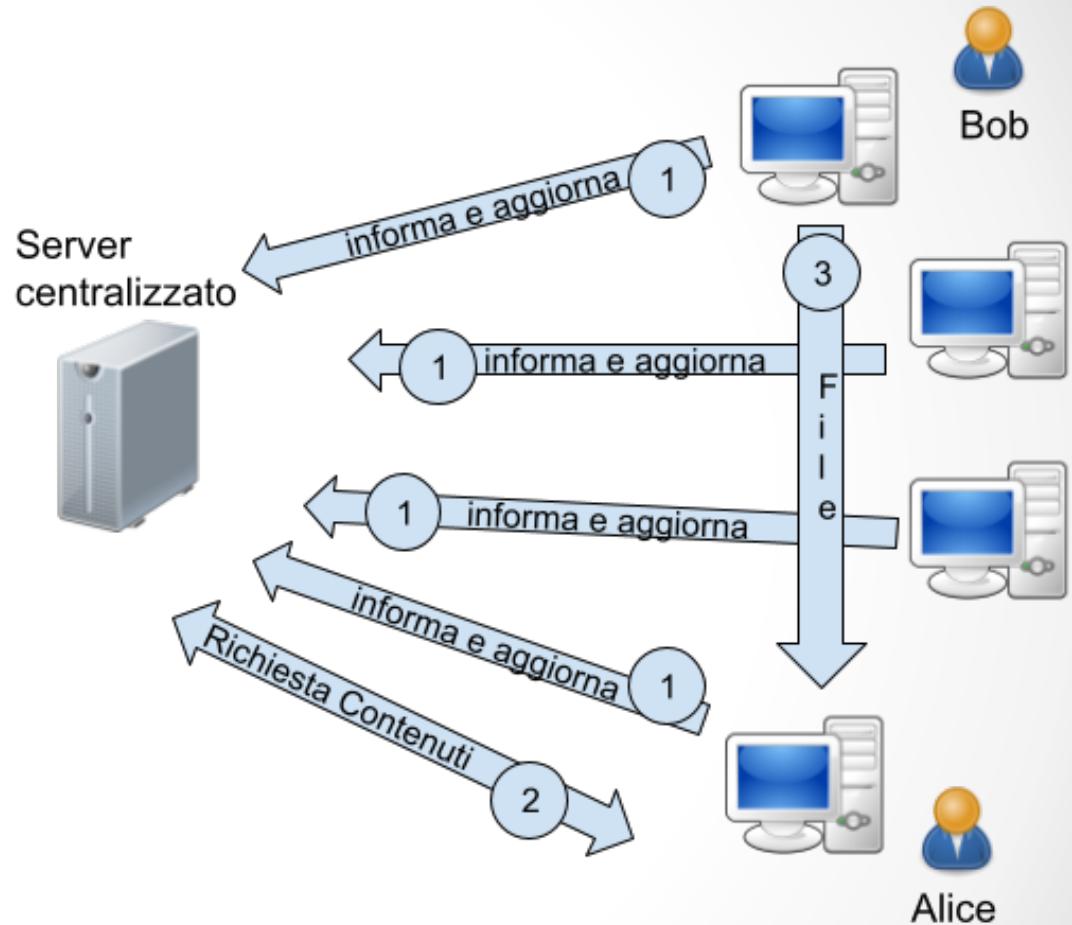
P2P directory centralizzata

- Quando un peer si connette alla rete si collega ad un server centralizzato fornendo:
 - Il proprio indirizzo IP
 - Il nome degli oggetti resi disponibili per la condivisione
- In questo modo raccoglie le info sui peer attivi e le aggiorna dinamicamente



P2P directory centralizzata

- Singolo punto di fallimento
- Collo di bottiglia per le prestazioni
- Violazione del diritto di autore



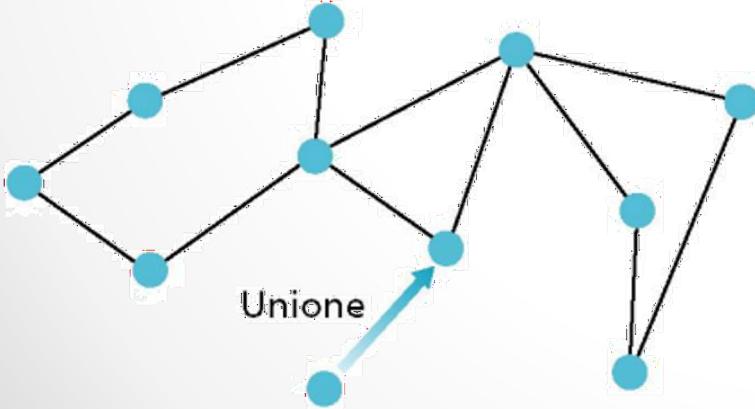
Il trasferimento dei file è decentralizzato, ma la localizzazione dei contenuti è pesantemente centralizzata !!!

P2P directory decentralizzata

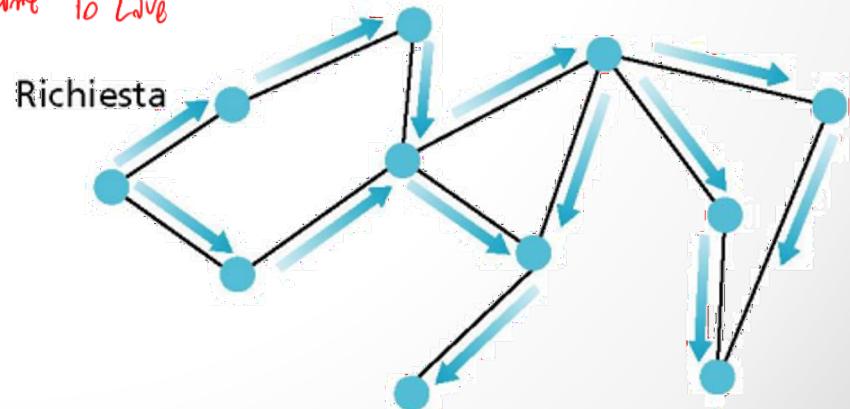
- Architettura completamente distribuita (no server centralizzati)
- Si realizza un'architettura di rete sovrapposta (overlay network) fatta da connessioni TCP tra peer → *Sopra la rete normale*
- L'overlay network ha una struttura paritetica
- Nonostante la rete possa avere centinaia di migliaia di peer, ogni peer è connesso al max a 10 altri peer nella overlay
- Due problemi:
 - come viene costruita e gestita la rete di peer
 - come un peer localizza un contenuto

P2P directory decentralizzata

- I peer, una volta unitisi alla rete, inviano richieste mediante la tecnica del flooding (inondazione), query flooding
 - Gnutella Query e QueryHit
 - Query flooding a raggio limitato
 - scope del messaggio, 7 ad esempio
- Client si aggiunge, invia query ad vicini.
I vicini inviano query ad altri.
Chi ha risorsa risponde con QueryHit,
con parametri che consente di interrompere
client maggiore. Maggiore è scuro.*

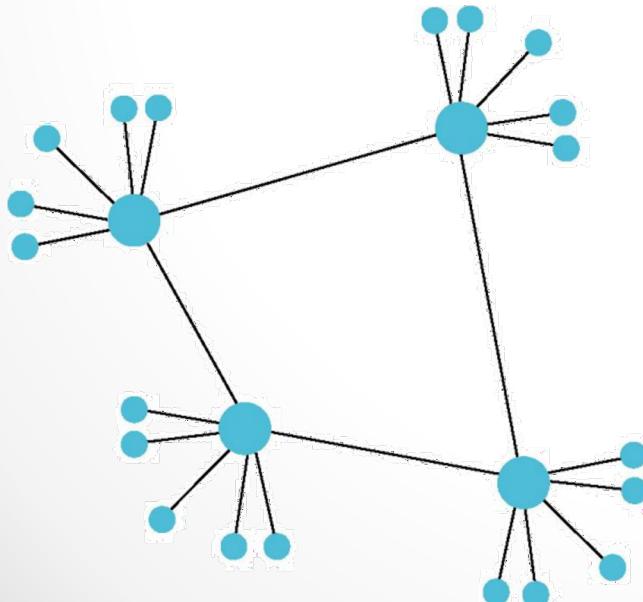


To Time To Live



P2P directory decentralizzata - variante

- Ogni peer è associato ad un group leader (mini hub) che è esso stesso un peer
- Un group leader memorizza le informazioni in condivisione dei "figli"
- Ogni group leader è in grado di interrogare altri group leader
 - query flooding applicato alla rete dei group leader



Legenda:

- Peer standard
- Peer leader del gruppo
- Relazioni tra vicini nella rete sovrapposta

P2P directory decentralizzata - variante

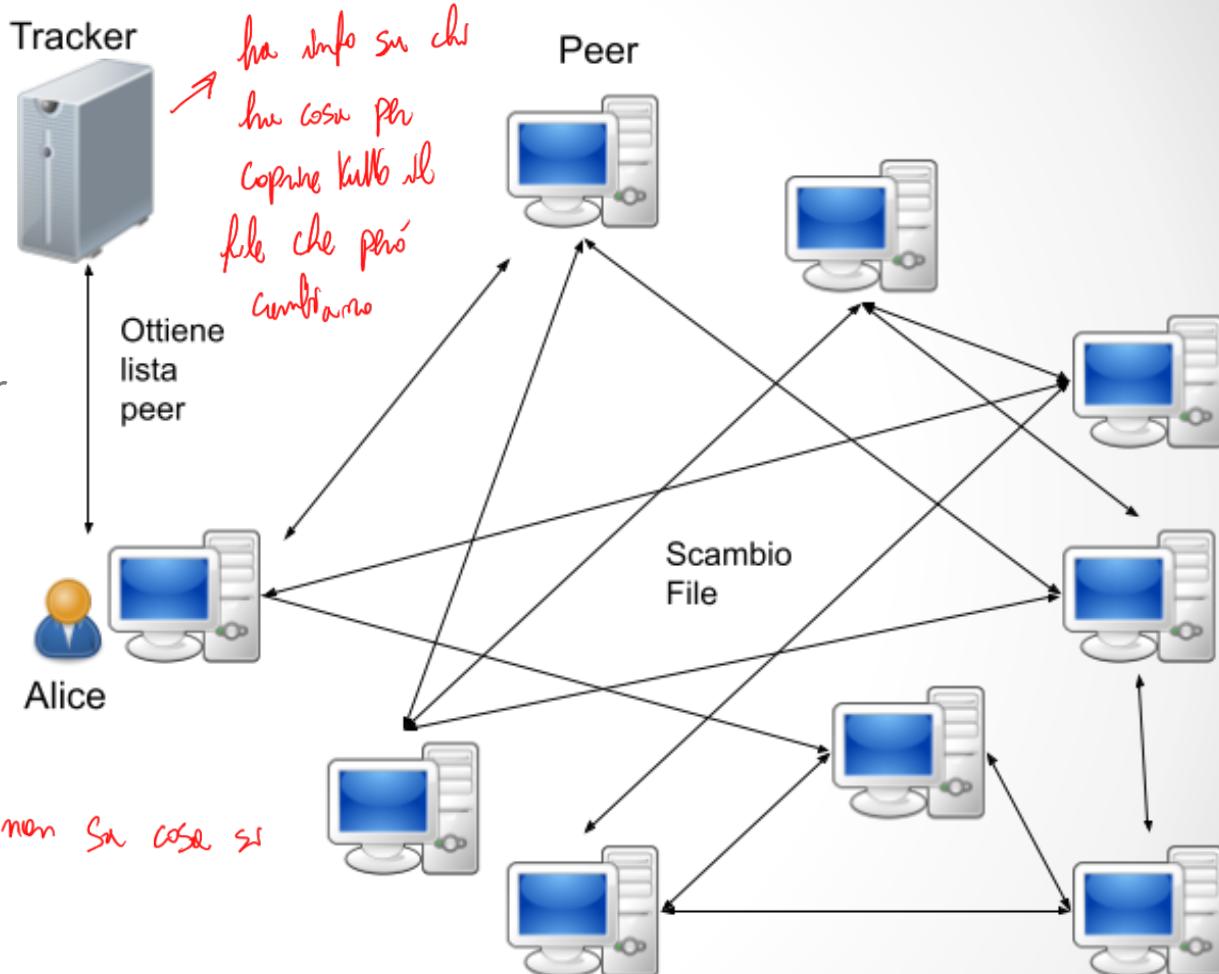
- **Fase di bootstrap:** un peer che si connette deve essere associato ad un group leader o deve essere designato group leader
- L'**overlay** è costituita da connessioni TCP tra peer e group leader e tra coppie di group leader
- Ogni file possiede un identificatore hash e un descrittore
- I peer (client) spediscono le query al proprio group leader; quest'ultimo risponde per ogni richiesta con l'indirizzo IP del detentore della risorsa, l'hash, e dei metadati associati alla risorsa.
- Il group leader inoltra sia le richieste sia le eventuali risposte da parte di altri group leader
- Il peer (client) seleziona la risorsa (file) per il download e invia una richiesta HTTP al detentore della risorsa usando l'hash come identificatore
- Tecniche per migliorare le prestazioni
 - Accodamento delle richieste e limitazione del numero degli upload simultanei
 - Downloading parallelo di parti dello stesso file da più utenti

P2P Bit Torrent

No group leader

- **Tracker:** tiene traccia dei peer che compongono un "torrente"
- **Swarm:** gruppo di peer che si scambiano porzioni (chunk) di un stesso file

Come capisco chi ha un chunk?



Tracker non ha colpe perché non sa cosa si stanno scambiando.

P2P Bit Torrent

- Il file è diviso in chunk di 256KB.
- Quando un peer si aggiunge ad uno "swarm":
 - si registra presso il tracker per avere la lista dei peer
 - si connette ad un sottoinsieme di tali peer ("neighbours")
 - non possiede chunk, ma ne accumulerà nel tempo
- Durante il download, il peer esegue l'upload di chunk verso altri peer
 - *↳ perché li sto scaricando: pezzi che ho scaricato*
- I peer possono attivarsi e disattivarsi dinamicamente
- Una volta scaricato l'intero file, il peer può (egoisticamente) abbandonare, o (altruisticamente) rimanere nello "swarm"

P2P Bit Torrent

Prelievo di chunk

- Peer differenti possiedono differenti sottoinsiemi di chunk del file
- Periodicamente, un peer chiede a tutti i 'neighbour' la lista dei chunk in loro possesso
- Tale peer invia richieste per i propri chunk mancanti
- Tecnica 'rarest first'
Scatta prima un chunk più raro, un po' un castro

Invio di chunk: 'tit-for-tat'

- L'idea di fondo è quella di dare priorità a chi fornisce dati al rate più alto.
- Un peer invia chunk ai 4 neighbour attualmente più veloci (che gli inviano chunk al rate più elevato)
- i 'top 4' sono ricalcolati ogni 10 secondi
- Ogni 30 secondi: si seleziona in maniera casuale un altro peer, e si inizia ad inviargli chunk
- il peer appena scelto può essere aggiunto ai 'top 4'
gl 4° viene buttato e ripescato.
- "optimistically unchoke"
Punto a diffondere file su più peer possibili
Meglio è un choke con velocità alta che si combra.
Tra le persone che lo hanno richiesto

P2P Bit Torrent

- Alice effettua l' "unchoke" ottimistico di Bob
- Alice diventa uno dei 'top-four provider' di Bob, il quale 'ricambia'
- Bob diventa uno dei 'top-four provider' di Alice



Con upload rate più elevato, si trovano partner 'migliori' e si scarica il file più velocemente!

Scambio comunque con tutta la rete.