



# Reti

(già “Reti di Calcolatori”)

## Applicazioni P2P

Renato Lo Cigno

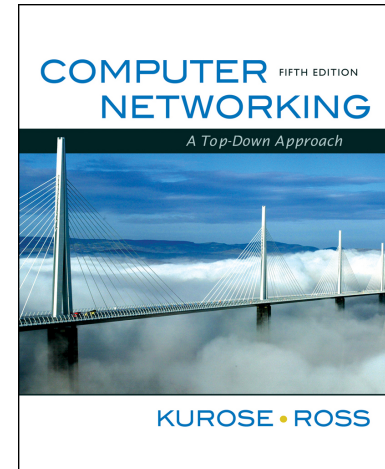
<http://disi.unitn.it/locigno/teaching-duties/reti>

## A note on the use of these slides:

These slides are an adaptation from the freely available version provided by the book authors to all (faculty, students, readers). The originals are in PowerPoint and English.

Adaptation is by Renato Lo Cigno, and Fabrizio Granelli

All material copyright 1996-2016  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking:  
A Top Down Approach,  
6th edition.*

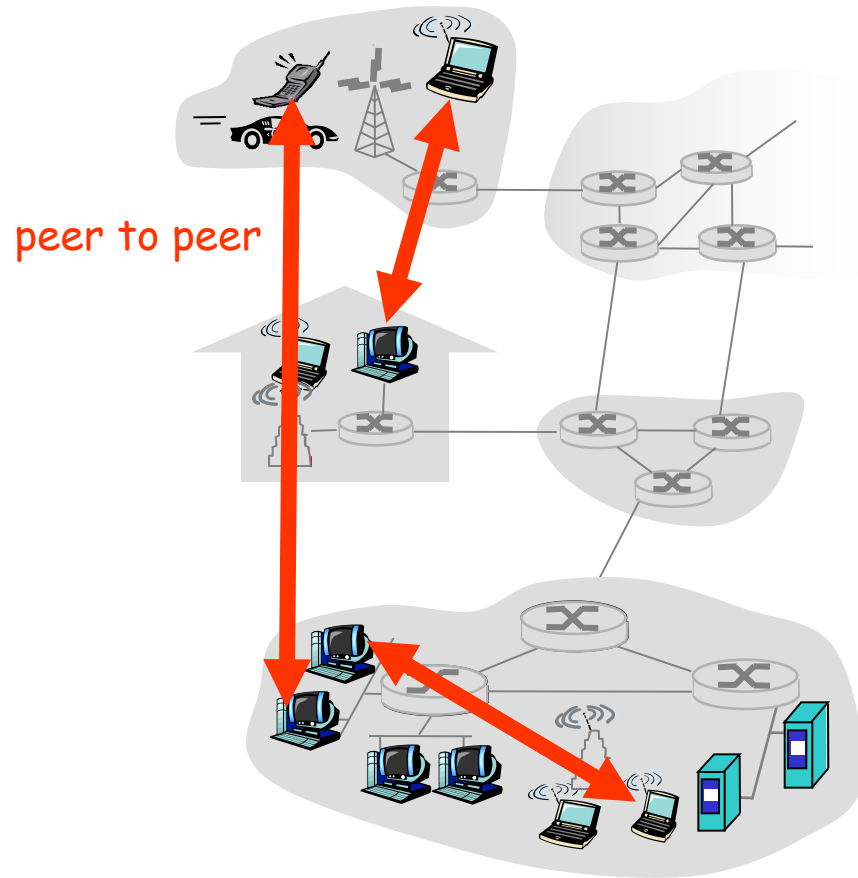
Jim Kurose, Keith Ross  
Addison-Wesley



## □ 2.6 Applicazioni P2P

## □ 2.7 Cloud computing

- ❑ non c'è un server sempre attivo
- ❑ coppie arbitrarie di host (peer) comunicano direttamente tra loro
- ❑ i peer non devono necessariamente essere sempre attivi, e cambiano indirizzo IP
- ❑ **Tre argomenti chiave:**
  - ❑ Distribuzione di file
  - ❑ Ricerca informazioni
  - ❑ Caso di studio: Skype

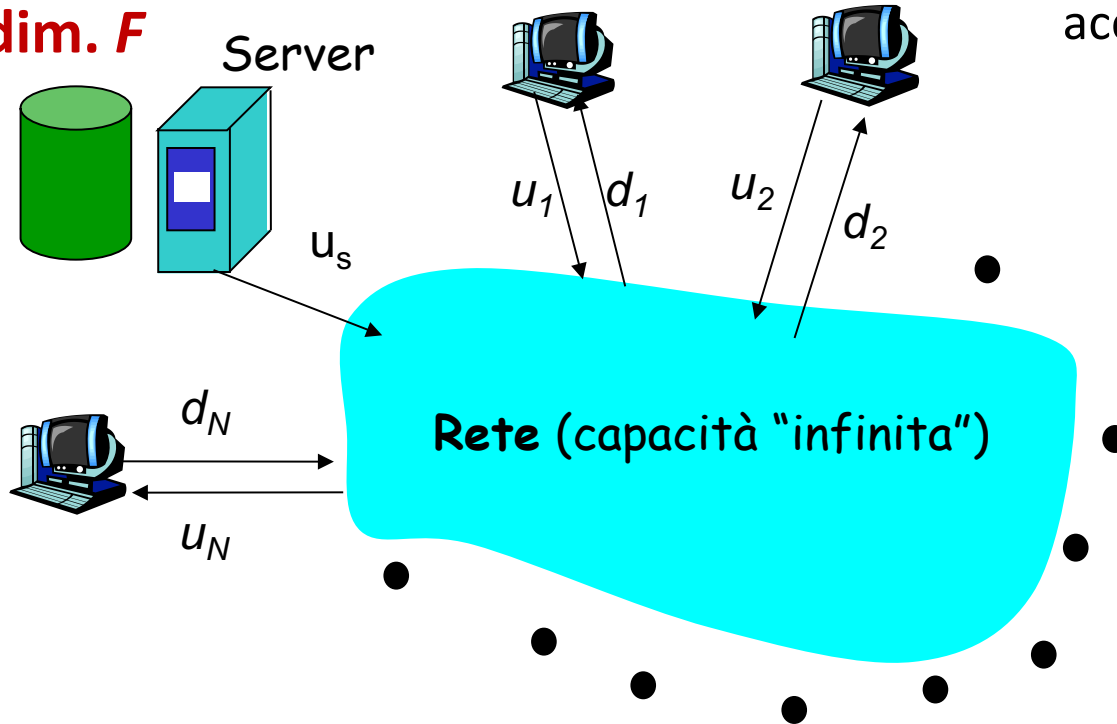




# Distribuzione di file: confronto tra Server-Client e P2P

Domanda : Quanto tempo ci vuole per distribuire file da un server a  $N$  host?

**File,  
dim.  $F$**

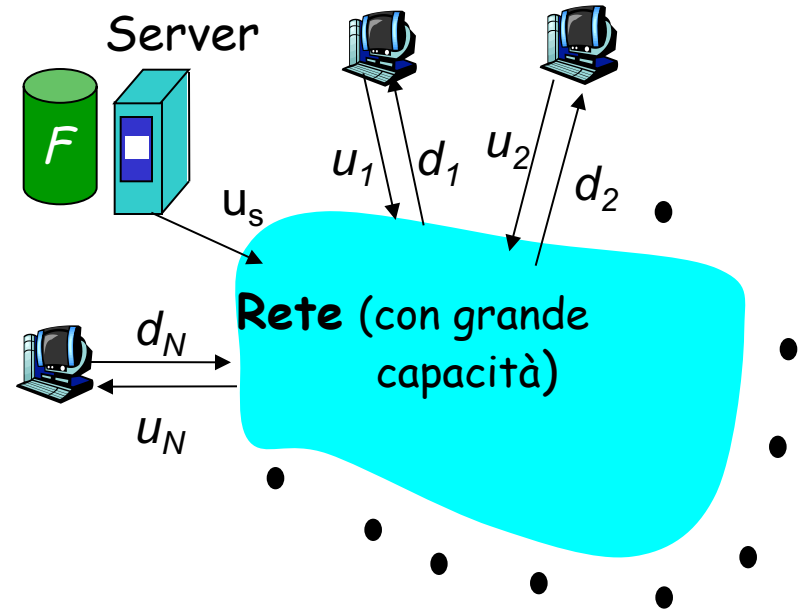


$u_s$ : capacità di upload  
del collegamento di  
accesso del server

$u_i$ : capacità di upload del  
collegamento di accesso  
dell' $i$ -esimo host

$d_i$ : capacità di  
download del  
collegamento di  
accesso dell' $i$ -esimo  
host

- Il server invia in sequenza  $N$  copie:
  - ❖  $Tempo = N (F/u_s)$
- Il client  $i$  impiega il tempo  $F/d_i$  per scaricare

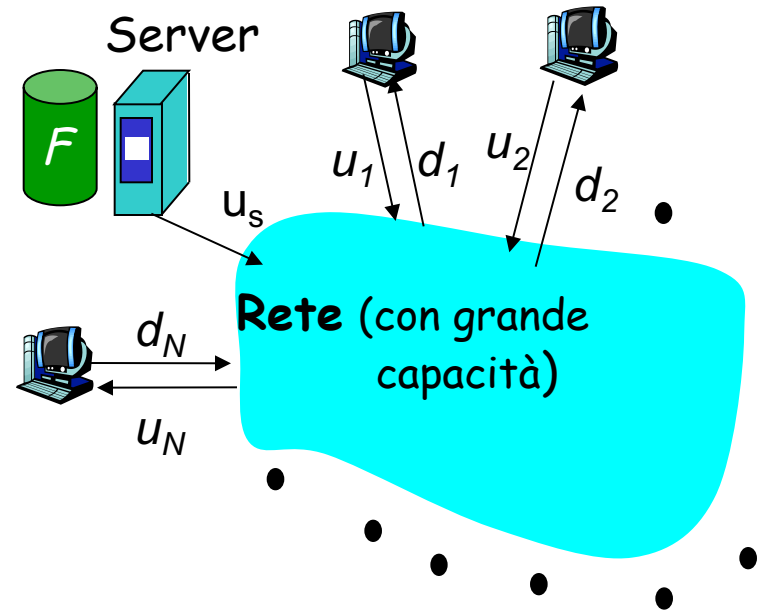


Tempo per distribuire  $F$   
a  $N$  client usando l'approccio client/server

$$= d_{cs} = \max \left\{ \frac{NF}{u_s}, \frac{F}{\min_i(d_i)} \right\}$$

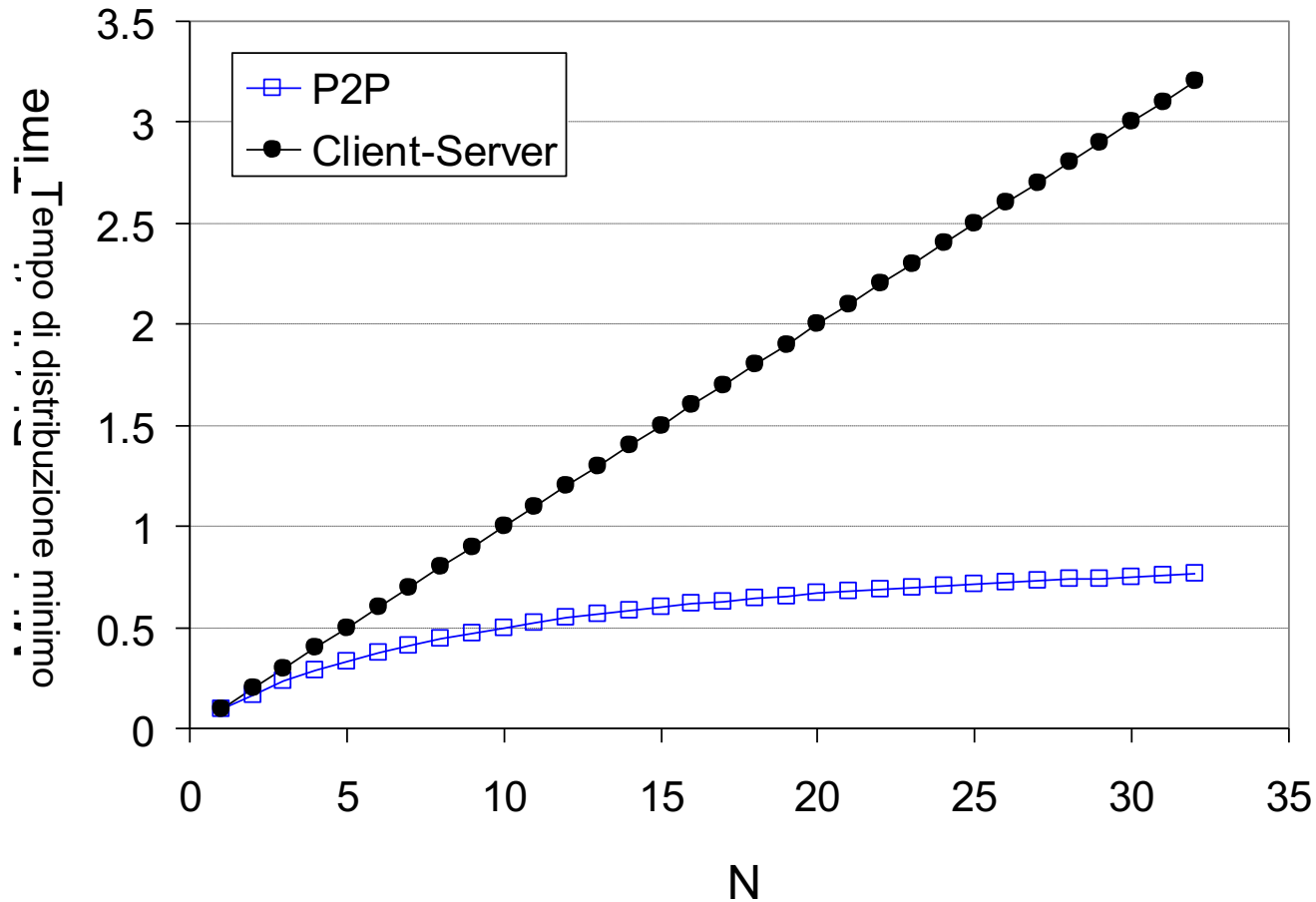
**aumenta linearmente con  $N$**

- ✗ il server deve inviare una copia nel tempo  $F/u_s$
- ✗ il client  $i$  impiega il tempo  $F/d_i$  per il download
- ✗ Devono essere scaricati  $NF$  bit
- ✗ Il più veloce tasso di upload è:  $u_s + \sum u_i$



$$d_{P2P} = \max \{ F/u_s, F/\min_i (d_i), NF/(u_s + \sum u_i) \}$$

Capacità upload del client =  $u$ ,  $F/u = 1$  ora,  $u_s = 10u$ ,  $d_{\min} \geq u_s$

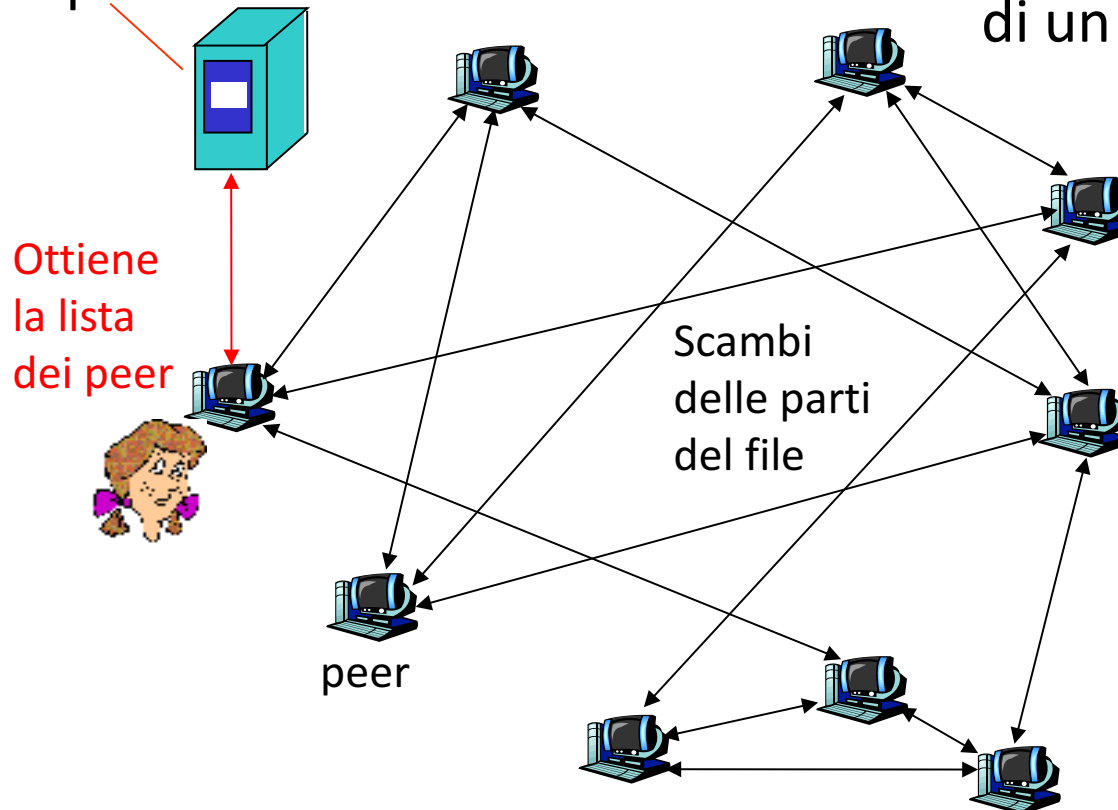




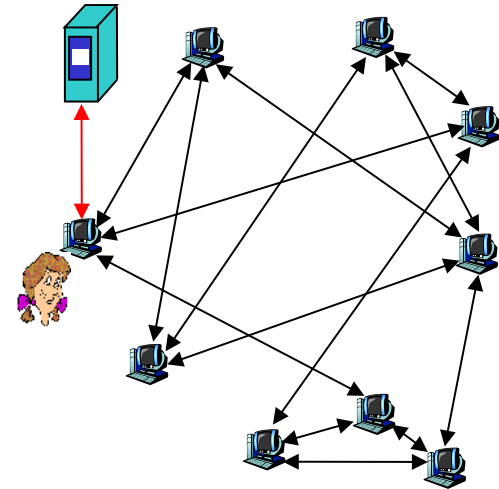
## □ Distribuzione di file P2P

tracker: tiene traccia dei peer che partecipano

torrent: gruppo di peer che si scambiano parti di un file



- ❑ Il file viene diviso in parti (*chunk*) da 256 Kb
- ❑ Quando un peer entra a far parte del torrent:
  - ❖ si registra presso il tracker per avere la lista dei peer, e si collega ad un sottoinsieme di peer vicini (“neighbors”)
  - ❖ non possiede nessuna parte del file, ma le accumula col passare del tempo
- ❑ Mentre effettua il download, il peer carica le sue parti su altri peer
- ❑ I peer possono entrare e uscire a piacimento dal torrent
- ❑ Una volta ottenuto l’intero file, il peer può lasciare il torrent (egoisticamente) o (altruisticamente) rimanere collegato

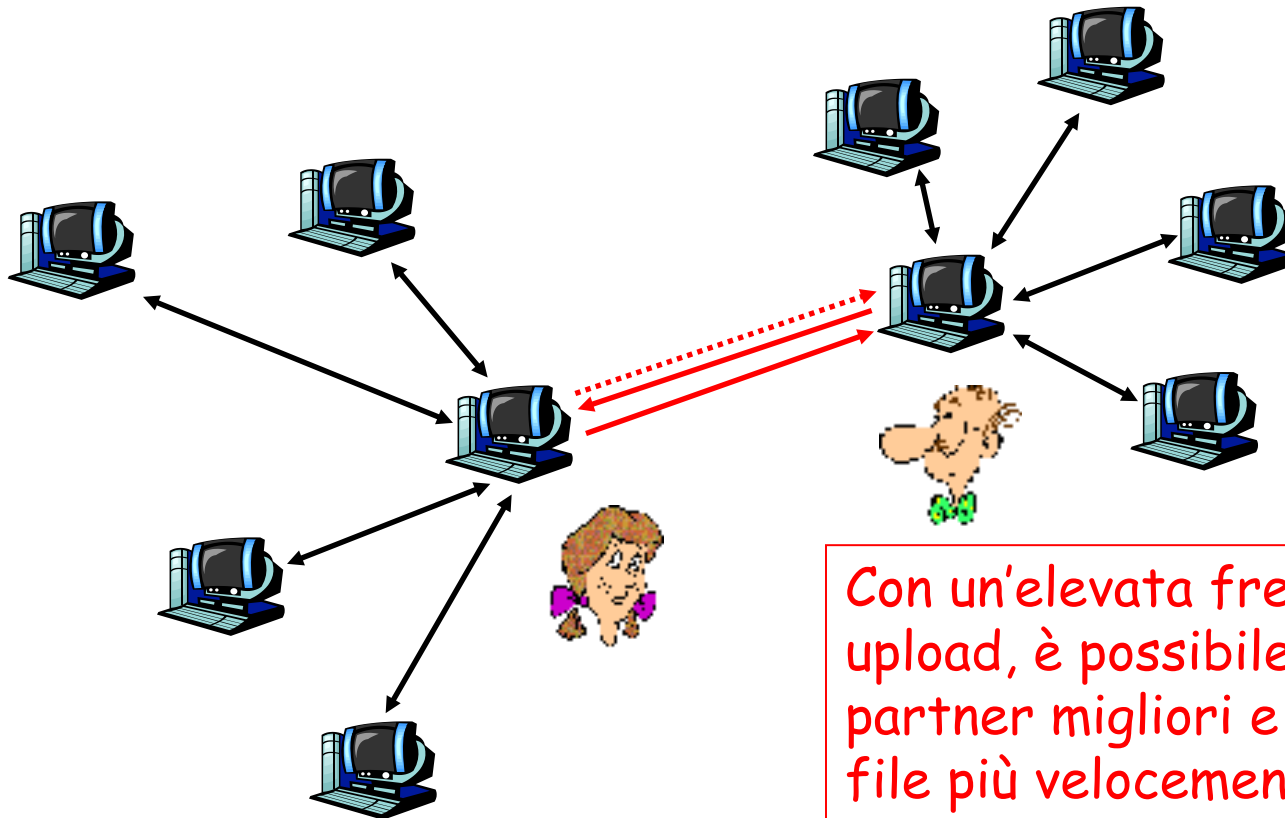




- ❑ In un dato istante, peer diversi hanno differenti sottoinsiemi del file
- ❑ periodicamente, un peer (Alice) chiede a ciascun vicino la lista dei chunk che possiede
- ❑ Alice invia le richieste per le sue parti mancanti:
  - ❖ Adotta la tecnica del *rarest first*
- ❑ Alice invia le sue parti a quattro vicini, quelli che attualmente le stanno inviando i propri chunk alla frequenza più alta
  - ❖ i 4 favoriti vengono rivalutati ogni 10 secondi
- ❑ ogni 30 secondi seleziona casualmente un altro peer, e inizia a inviargli chunk
  - ❖ Il peer appena scelto può entrare a far parte dei top 4
  - ❖ A parte i “top 4” e il “nuovo entrato”, gli altri peer sono “soffocati”, cioè non ricevono nulla



- (1) Alice casualmente sceglie Roberto
- (2) Alice diventa uno dei quattro fornitori preferiti di Roberto; Roberto ricambia
- (3) Roberto diventa uno dei quattro fornitori preferiti di Alice





Indice nei sistemi P2P: corrispondenza tra le informazioni e la loro posizione negli host

## File sharing (es. e-mule)

- ❑ L'indice tiene traccia dinamicamente della posizione dei file che i peer condividono.
- ❑ I peer comunicano all'indice ciò che possiedono.
- ❑ I peer consultano l'indice per determinare dove trovare i file

## Messaggistica istantanea

- ❑ L'indice crea la corrispondenza tra utenti e posizione.
- ❑ Quando l'utente lancia l'applicazione, informa l'indice della sua posizione
- ❑ I peer consultano l'indice per determinare l'indirizzo IP dell'utente.

## Progetto originale di "Napster"

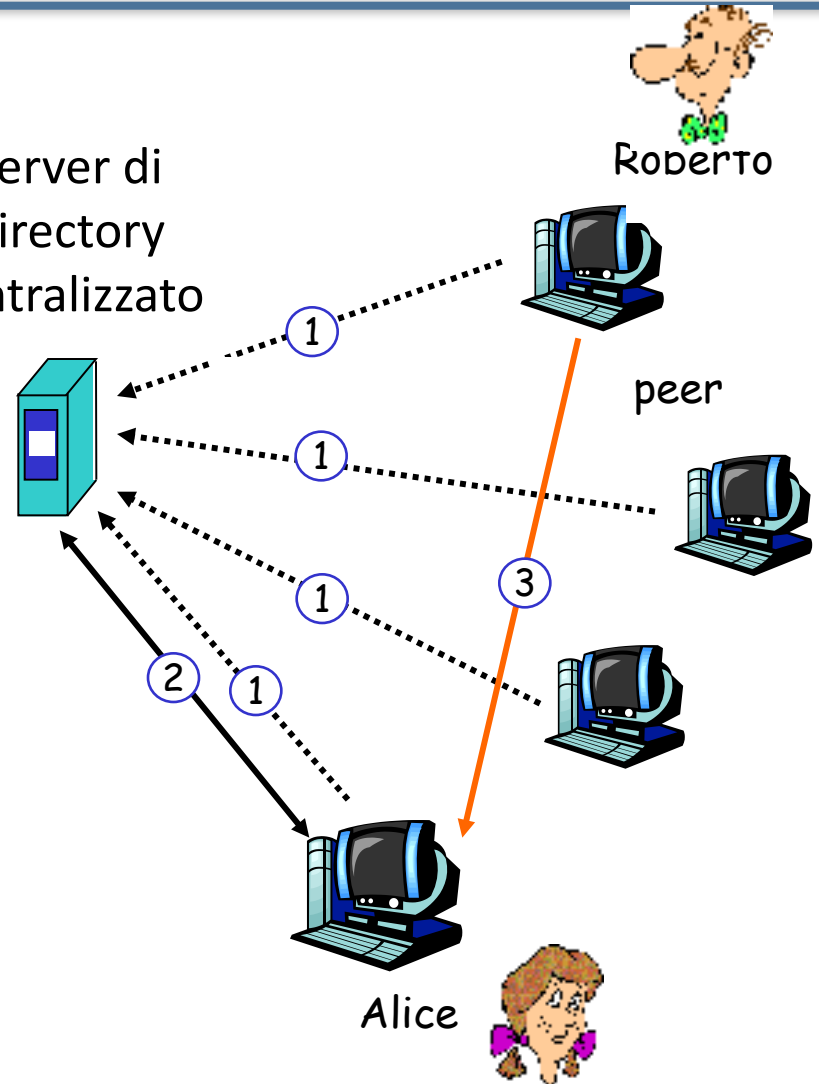
1) quando il peer si collega, informa il server centrale:

- ❖ indirizzo IP
- ❖ contenuto

2) Alice cerca la canzone "Hey Jude"

3) Alice richiede il file a Roberto

Server di directory centralizzato





Il trasferimento dei file è distribuito, ma il processo di localizzazione è fortemente centralizzato

- ❑ Unico punto di guasto
- ❑ Collo di bottiglia per le prestazioni
- ❑ Violazione del diritto d'autore
  - ❑ Non cambia, ma chi ospita la directory è facile da colpire e ... tradisce gli altri



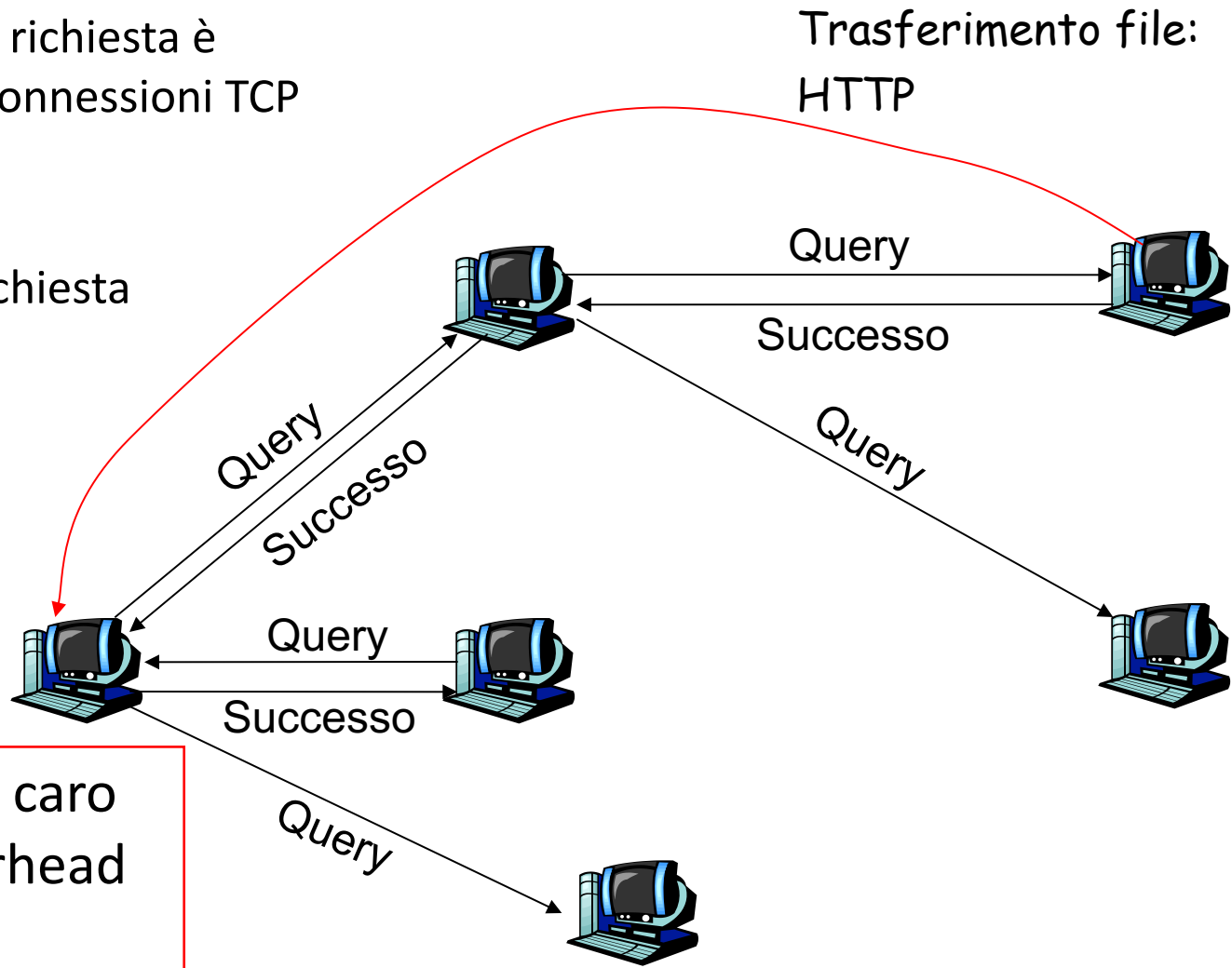
- ❑ Completamente distribuito
  - ❖ nessun server centrale
- ❑ Protocollo di pubblico dominio usato da Gnutella
- ❑ Ciascun peer indicizza i file che rende disponibili per la condivisione (e nessun altro)

## Rete di copertura: grafo

- ❑ Arco tra i peer X e Y se c'è una connessione TCP
- ❑ Tutti i peer attivi e gli archi formano la rete di copertura
- ❑ Un arco è un collegamento virtuale e *non* fisico
- ❑ Un dato peer sarà solitamente connesso con meno di 10 peer vicini nella rete di copertura

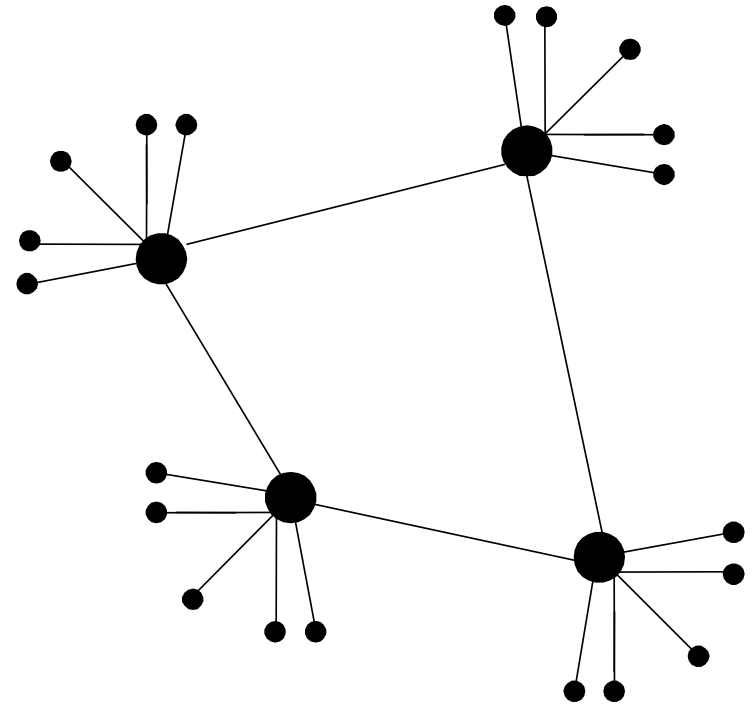


- ❑ Il messaggio di richiesta è trasmesso sulle connessioni TCP esistenti
- ❑ Il peer inoltra il messaggio di richiesta
- ❑ Il messaggio di successo è trasmesso sul percorso inverso



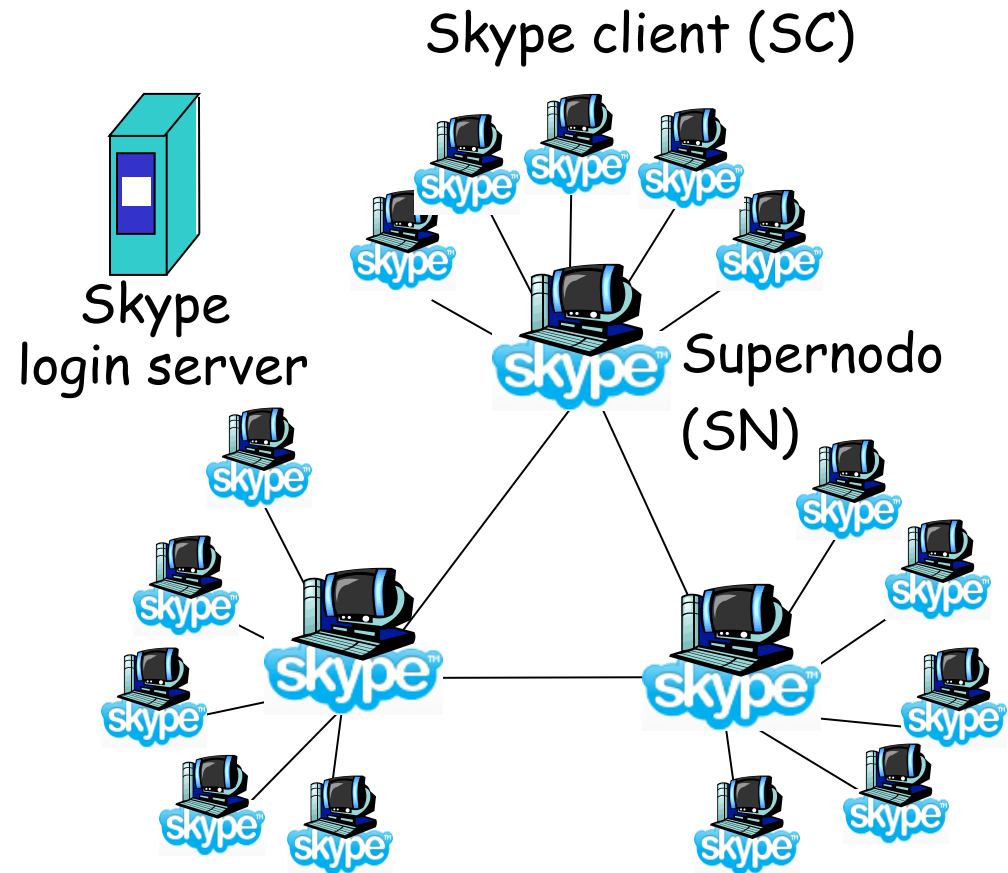
Flooding è molto caro  
in termini di overhead  
Soluzione: DHT

- La copertura gerarchica combina le migliori caratteristiche di indice centralizzato e query flooding
- Ogni peer è un leader di gruppo o è assegnato a un leader di gruppo
  - ❖ Connessione TCP tra peer e il suo leader di gruppo
  - ❖ Connessioni TCP tra qualche coppia di leader di gruppo
- Il leader di gruppo tiene traccia del contenuto di tutti i suoi figli.



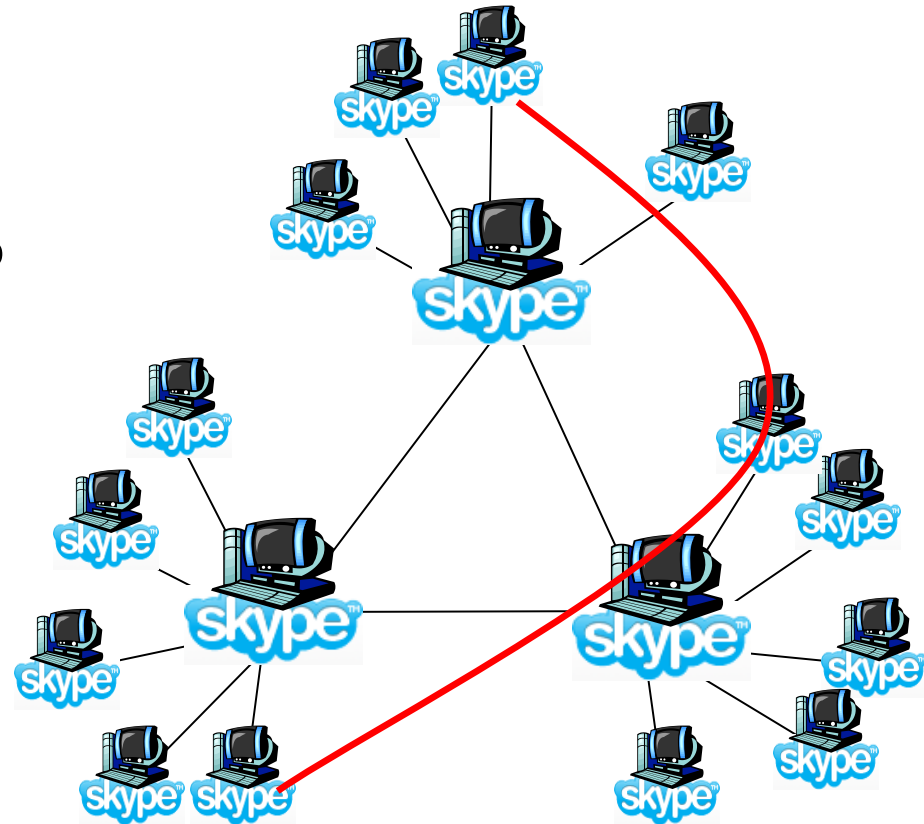
- Peer ordinario
- Peer leader di gruppo
- Relazioni di adiacenza nella rete di copertura

- ❑ intrinsecamente P2P: coppie di utenti comunicano tra loro
- ❑ Protocollo proprietario (dedotto mediante reverse engineering)
- ❑ Copertura gerarchica con i supernodi
- ❑ L'indice crea corrispondenza tra nomi utente e indirizzi IP
- ❑ Autenticazione e gestione degli account centralizzata



## Peer e relay

- Si pone un problema quando sia Alice che Roberto hanno NAT.
  - ❖ NAT evita che un host al di fuori della rete domestica crei una connessione con un host all'interno di questa
- Soluzione
  - ❖ Usando il supernodo di Alice e Roberto, si sceglie un relay
  - ❖ Ciascun peer inizia la sessione con il relay.
  - ❖ I peer ora comunicano con NAT attraverso il relay

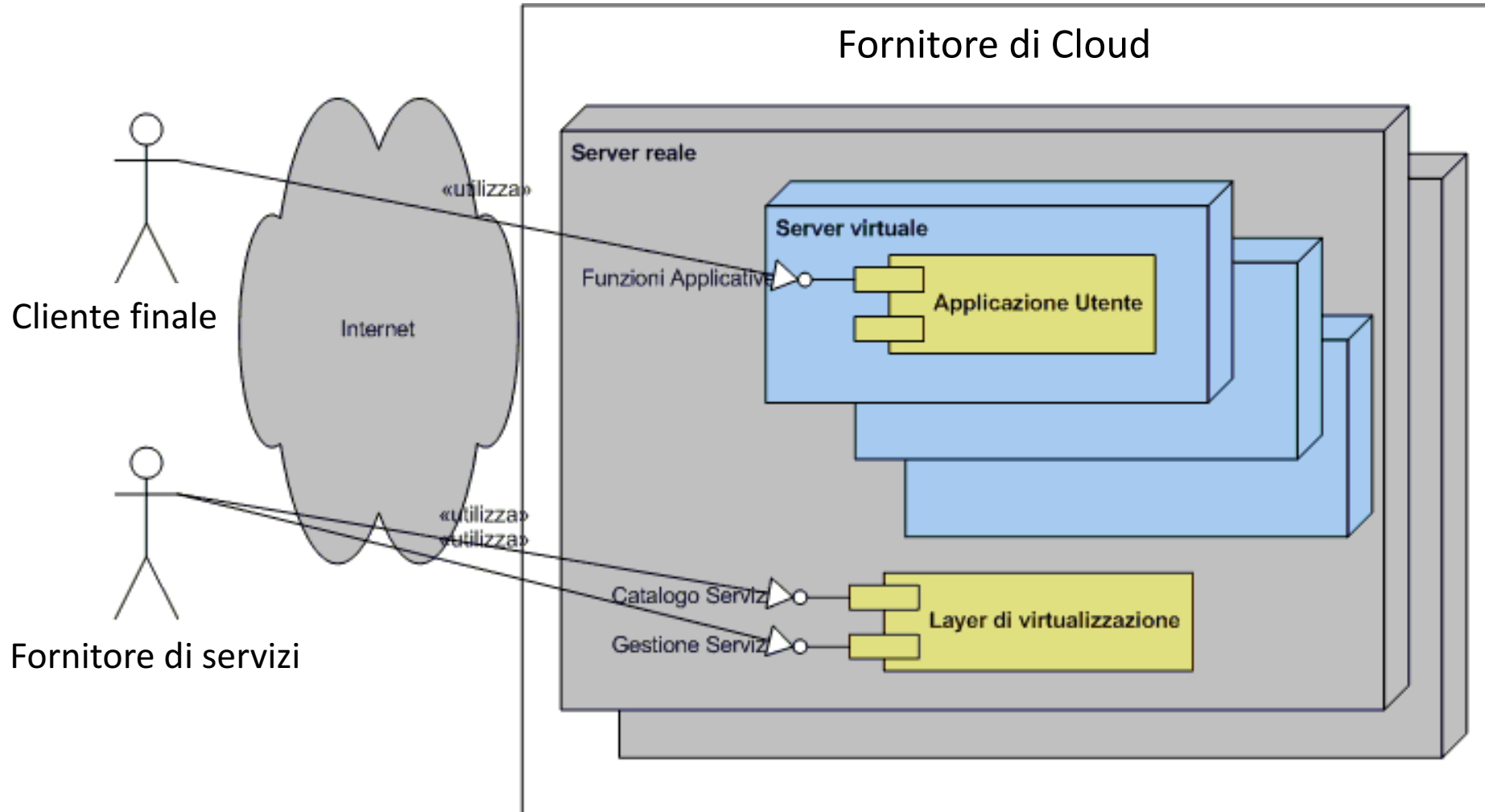




□ 2.6 Condivisione di file P2P

□ **2.7 Cloud computing**

- ❑ Virtualizzazione dell'infrastruttura
- ❑ Il fornitore di servizi “compra” l'infrastruttura virtuale dal fornitore della “cloud” anziché gestire l'hardware in proprio
- ❑ Il fornitore cloud espone delle interfacce per elencare e gestire la propria offerta di piattaforma virtuale
- ❑ Il fornitore di servizi utilizza tali interfacce per selezionare il servizio richiesto (ad es. un server virtuale completo oppure solo storage) e per amministrarlo (configurazione, attivazione, disattivazione)
- ❑ Il cliente finale utilizza il servizio configurato dal fornitore di servizio





- ❑ Le caratteristiche fisiche dell'implementazione (server reale, localizzazione del data center) sono irrilevanti (quasi)
  
- ❑ Criticità:
  - ❑ Sicurezza e privacy
  - ❑ Problemi internazionali di natura economica e politica
  - ❑ Continuità del servizio offerto
  - ❑ Difficoltà di migrazione dei dati





- YouTube serve più di 100 milioni di video al giorno

### Come fa?

- Ogni video è ospitato su di un mini-cluster → qualsiasi video è gestito da più di una macchina (più velocità, backup online)
- Utilizzano il server lighttpd[\*] per i video (più veloce e scalabile di altri server come Apache o TomCat) su TCP ... quindi affidabile, ma non real-time

[\*] <http://www.lighttpd.net/> ... forse o forse no, o dipende dal momento, ma è del tutto irrilevante essendo un dettaglio di implementazione del servizio



- I contenuti più popolari vengono spostati sulla Content Delivery Network (CDN):
  - Replica dei contenuti in differenti posizioni in modo da “avvicinarle” agli utenti (meno hop, rete meno congestionata)
- I contenuti meno popolari (<1-20 views/day) restano sui server YouTube