Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica

Sistemi operativi e reti

A.A. 2016-17

Pietro Frasca

Parte II: Reti di calcolatori

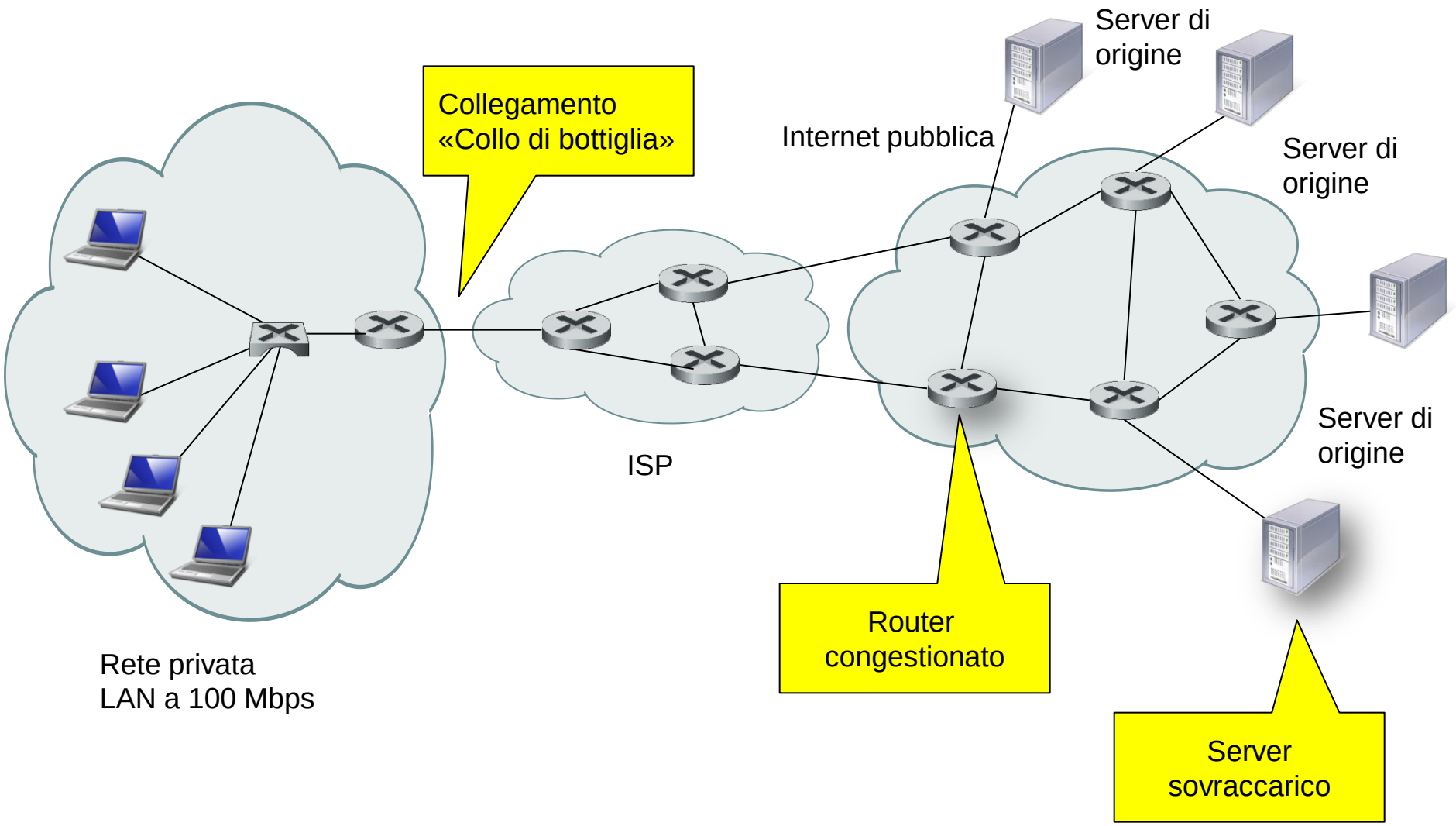
Lezione 9 (33)

Martedì 4-04-2017

Distribuzione di contenuti

- Con i protocolli client/server fino ad ora descritti, un utente può scaricare file di contenuto vario. E' possibile però, che il tempo impiegato per il trasferimento possa essere troppo lungo per vari motivi, tra i quali:
  - client e server sono connessi da un percorso in cui è presente uno o più collegamenti a bassa velocità;
  - tra client e server c'è almeno un router congestionato, che causa elevati ritardi di coda e perdita di pacchetti.
  - Al server, dove sono presenti i contenuti desiderati, arrivano contemporaneamente richieste di molti client, che rendono il server sovraccarico e il collegamento dello stesso un collo di bottiglia per ciascuna connessione.

- Per ridurre i suddetti ritardi, una possibile soluzione consiste nel copiare **il contenuto** di un server su altri server e di indirizzare i client su uno dei server che più velocemente può trasmettere le informazioni richieste.

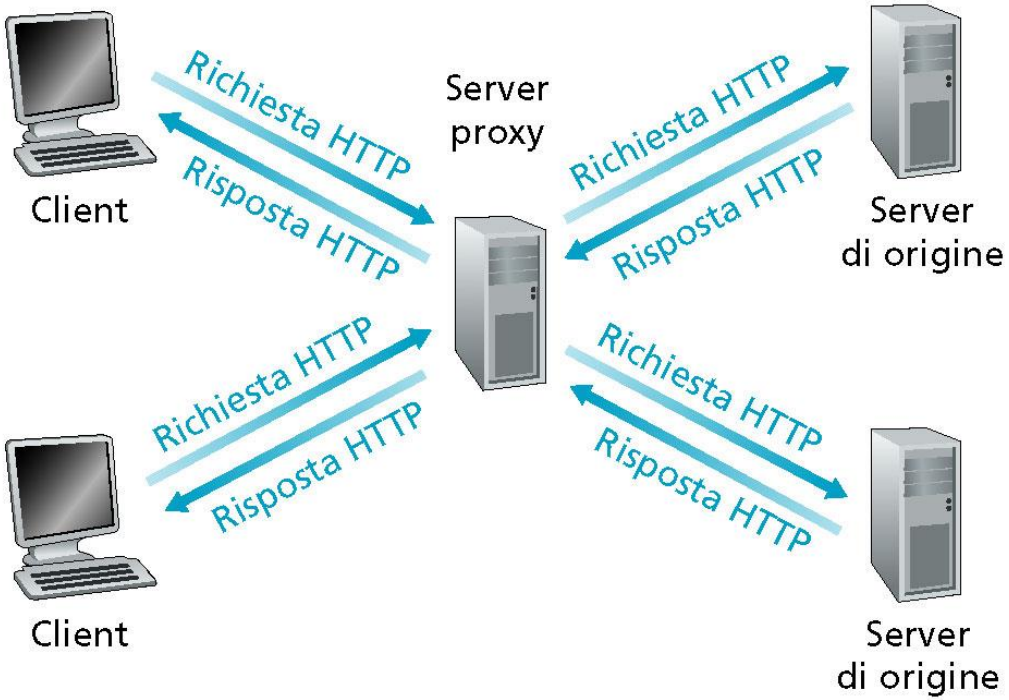

Server di origine
Collegamento «Collo di bottiglia»
Internet pubblica
Server di origine
Server di origine
ISP
Router congestionato
Rete privata
LAN a 100 Mbps
Server sovraccarico

- Con il termine **distribuzione di contenuti** si intende l'insieme di tecnologie, sia software che hardware, che consentono di copiare file su molti server e di permettere ai client di accedere più velocemente alle copie dei file contenute in tali server.
- Diffuse tecnologie per la distribuzione di contenuti, con caratteristiche e obiettivi diversi, sono:
  - **Server proxy**;
  - **reti per la distribuzione di contenuti (CDN, *Content Distribution Networks)***

- Da un certo punto di vista le applicazioni di rete basate su architettura P2P possono essere considerate una tecnologia di distribuzione di contenuti.

Server proxy

- Un browser può essere configurato in modo tale che tutte le richieste HTTP e/o FTP siano ridirette a un server proxy.
- Un **server proxy** è un computer su cui gira un opportuno software che memorizza nel suo file system copie di file man mano che i client navigano su internet.
- Il funzionamento è illustrato nella figura seguente.



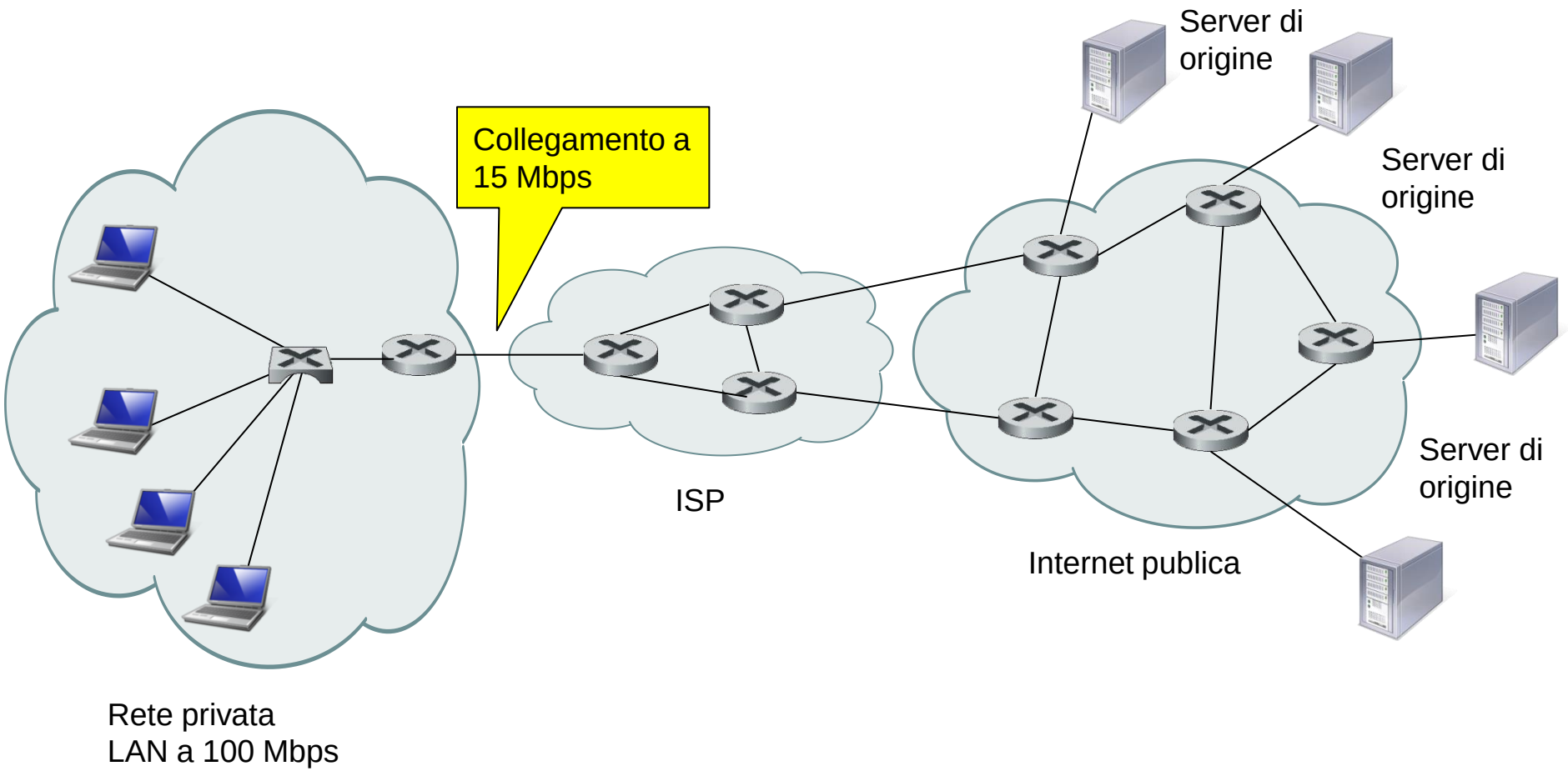
- Il browser ridirige una richiesta HTTP destinata ad un server web (che indicheremo con **server di origine**) al proxy.
- Il proxy verifica se possiede una copia del file richiesto.
  - Se ha il file, lo invia al browser.
  - Se non ha il file, lo richiede al server di origine. Quando il proxy riceve il file, ne salva una copia nel suo filesystem e ne invia una copia al browser.
- Per esempio, una società privata potrebbe installare un proxy nella sua rete e configurare tutti i browser installati sui suoi host in modo che tutte le richieste HTTP passino attraverso il proxy.
- I proxy consentono una forma di distribuzione di contenuti in quanto copiano le informazioni richieste dai client nei loro file system.
- È da notare che i fornitori di contenuti non si devono occupare della distribuzione delle informazioni che avviene automaticamente ogni volta che gli utenti visitano i siti di quei fornitori.

- I proxy, ai fini della distribuzione dei contenuti, producono i seguenti vantaggi:
  - Un proxy può ridurre notevolmente il tempo di risposta a una richiesta del client, nel caso in cui la larghezza di banda tra il client e il proxy è superiore a quella tra il client e il server di origine.
  - il proxy può ridurre sensibilmente il traffico su un link di accesso a Internet di un'azienda. Inoltre, a livello globale, l'uso dei proxy può ridurre in modo consistente il traffico dell'intera Internet.
  - L'installazione di molti proxy fornisce un'infrastruttura per una veloce distribuzione di contenuti. L'installazione di molti proxy favorisce soprattutto quei siti che pubblicano contenuti di grande interesse collettivo ma hanno limitate risorse hardware.

Esempio di uso di proxy

- Consideriamo lo schema della figura seguente, in cui una rete privata è connessa a Internet mediante un ISP con un link a 15 Mbit/s.
- Supponiamo che gli utenti della rete privata richiedano pagine web di dimensione media di 1 Mbit con una frequenza media di 15 richieste al secondo.
- Trascuriamo il traffico generato da un messaggio di richiesta HTTP per via della sua piccola dimensione.
- Supponiamo inoltre che il "**ritardo internet**", consistente nel tempo trascorso da quando il router del lato Internet invia una richiesta HTTP a quando riceve la corrispondente risposta, sia mediamente di **due secondi**.
- Il **tempo totale di risposta** è dato dalla somma del ritardo della LAN, del ritardo di accesso (il ritardo fra i due router) e del ritardo Internet.

tempo_tot_risp = R_LAN + R_Accesso + R_Internet


Server di origine
Server di origine
Server di origine
Collegamento a 15 Mbps
ISP
Internet publica
Rete privata
LAN a 100 Mbps

- Facciamo ora un calcolo semplificato per stimare questo ritardo.
  L'intensità del traffico è data dalla relazione:

  intensitàTraffico = L·a / R

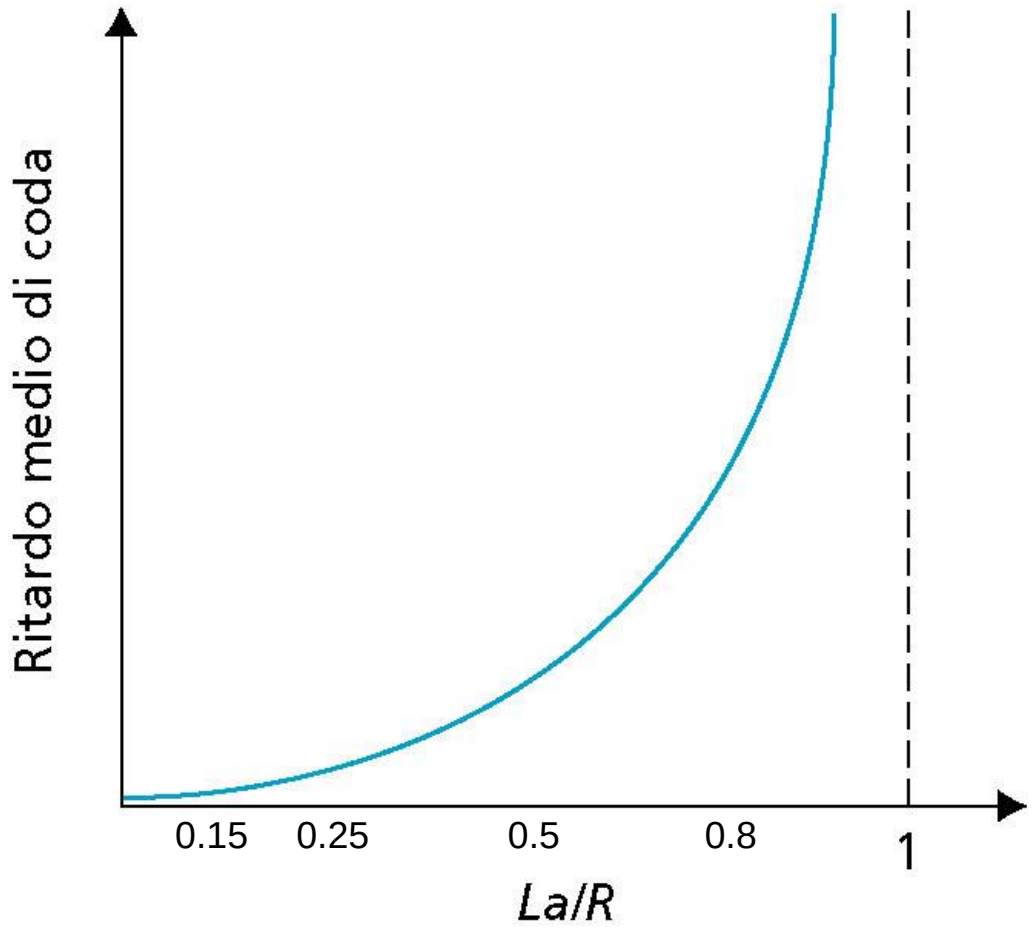
  quindi, sulla LAN l'intensità del traffico è:

  $(10^6 \cdot 15)/ 10^8 = 15 \cdot 10^{-2} = 0.15$

  mentre l'intensità del traffico sul link di accesso (dal router Internet al router della rete privata) è

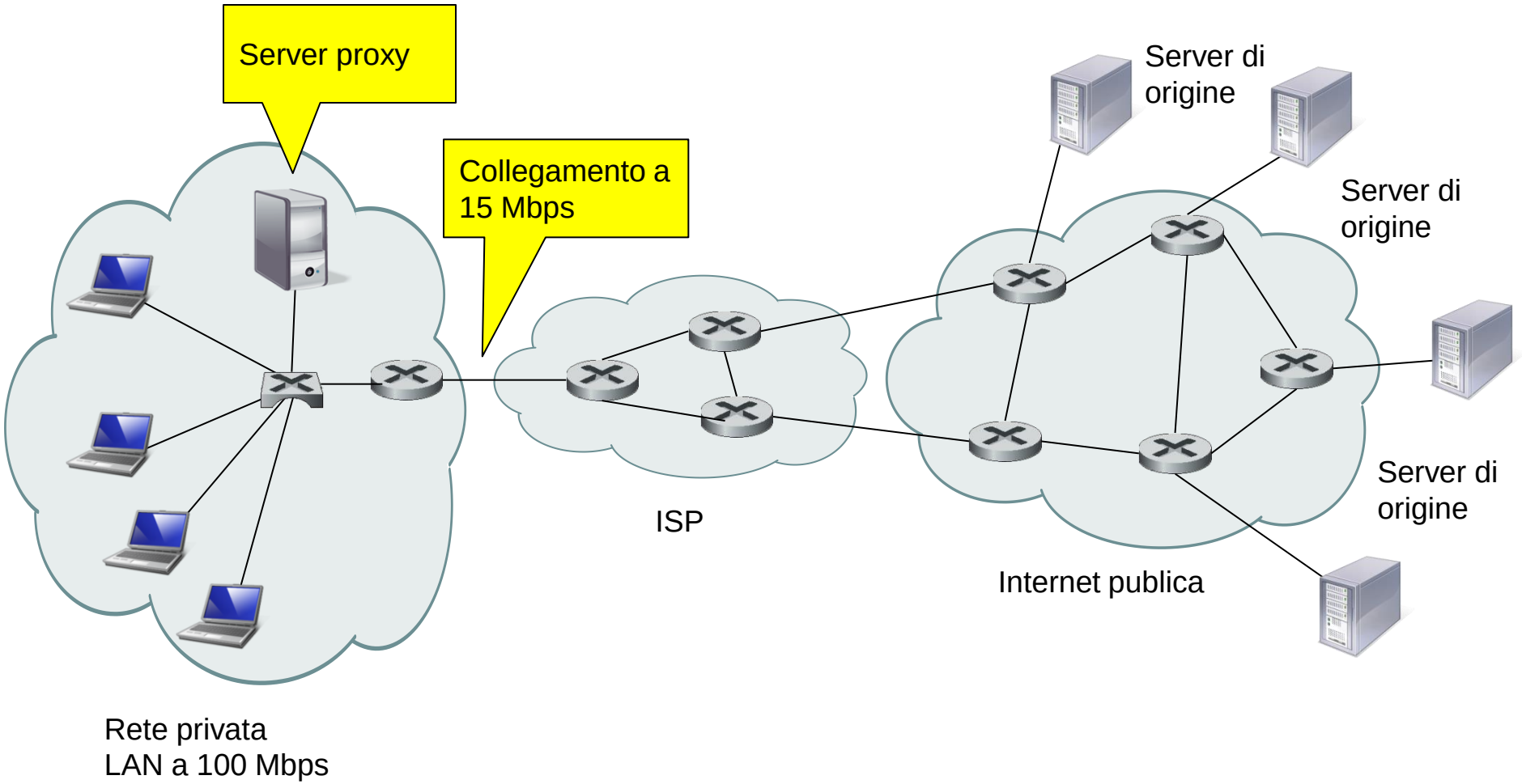
  $(10^6 \cdot 15)/ (15 \cdot 10^6) = 1$

  Un'intensità di traffico di 0.15 su una LAN tipicamente produce un trascurabile **ritardo di coda di pochi millisecond**i.
  Tuttavia, come abbiamo visto in una precedente lezione, quando l'intensità del traffico è prossima a 1, come nel caso del link di accesso, il ritardo di coda diventa molto grande.



Ritardo medio di coda in funzione dell'intensità del traffico

- Quindi, il tempo di risposta medio per soddisfare le richieste sarà dell'ordine dei minuti, che è un ritardo inaccettabile per gli utenti.
- Per risolvere questo problema si potrebbe aumentare la velocità del link di accesso da 15 Mbit/s a, per esempio, 100 Mbit/s. Questa soluzione ridurrà l'intensità di traffico sul link di accesso da 1 a 0,15, che porta ad un ritardo, tra i router, trascurabile. Con questa soluzione, il **tempo totale di risposta** sarà di **circa 2 secondi**, cioè, poco più del ritardo Internet. Tuttavia, tale soluzione ha un costo molto elevato per portare il link di accesso da 15 Mbit/s a 100 Mbit/s.
- Consideriamo ora una soluzione che prevede di non aumentare la velocità del collegamento di accesso ma di **installare un server proxy nella rete privata**.
- Supponiamo che per questo esempio il proxy possa soddisfare il 40% delle richieste. Tipicamente le percentuali di successo (hit rate) variano tra 0,2 e 0,7. Poiché i client e il proxy sono connessi alla stessa rete LAN, il 40% delle richieste saranno ottenute dal proxy in circa 10 millisecondi.


Server proxy
Collegamento a
15 Mbps
Server di
origine
Server di
origine
Server di
origine
ISP
Internet publica
Rete privata
LAN a 100 Mbps

- Tuttavia, il restante 60% delle richieste sarà ottenuto dai server di origine. Ma con solo il 60% degli oggetti richiesti da passare attraverso il link di accesso, l'intensità del traffico su questo link si riduce da 1,0 a 0,6.
- Generalmente, un'intensità del traffico inferiore allo 0,8 corrisponde a un piccolo ritardo di decine di millisecondi. Fatte queste considerazioni, il ritardo medio è quindi:

$$0.4 \cdot 10^{-2} + 0.6 \cdot 2.01 = 0.004 + 1.206 = 1.21 \text{ secondi}$$

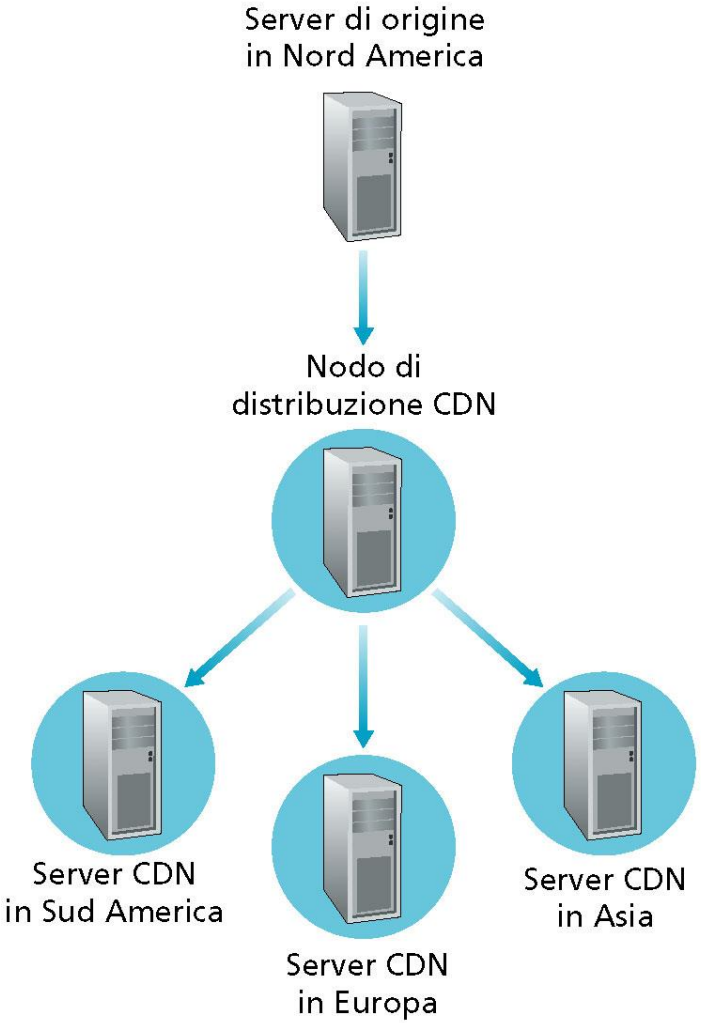
Vediamo che, questa seconda soluzione fornisce un tempo di risposta addirittura inferiore a quello della prima e non richiede di aumentare la larghezza di banda del link di connessione a Internet. E' necessario solo installare un server proxy: una soluzione a basso costo, in quanto molti proxy usano software di pubblico dominio che girano anche su PC.

Reti per la distribuzione di contenuti

- In questi ultimi anni si sono diffuse in Internet **le reti per la distribuzione di contenuti (CDN, *Content Distribution Networks).***
- I clienti di una CDN sono i fornitori di contenuti.
- Un fornitore di contenuti, ad esempio Yahoo, si serve di una società fornitrice di CDN, ad esempio Akamai, per trasmettere i suoi contenuti agli utenti con ritardi più bassi possibile.
- Un'azienda fornitrice di CDN tipicamente fornisce il servizio di distribuzione dei contenuti nel seguente modo:
  - L'azienda fornitrice di CDN installa centinaia di server CDN in edifici detti **centri di Internet hosting**.
  - La CDN copia nei suoi server i contenuti dei suoi clienti. Quando un cliente aggiorna i suoi contenuti, per esempio, modifica una pagina Web, la CDN ridistribuisce il nuovo contenuto nei server CDN.

- La ditta fornitrice di CDN rende disponibile un meccanismo tale che quando un utente richiede un contenuto, il contenuto viene fornito dal server CDN che lo può inviare all'utente in tempi più brevi. Questo server può essere il server CDN più vicino all'utente.
- In genere, le CDN utilizzano funzioni avanzate *(redirection)* del DNS per fornire ai browser l'indirizzo del server migliore.

La figura seguente mostra l'interazione tra il fornitore di contenuti e l'azienda fornitrice di *CDN.*



- Il cliente può chiedere alla CDN di distribuire parzialmente i suoi contenuti, ad esempio quelli di tipo multimediale, e distribuire i rimanenti tipi per conto proprio, senza l'intervento della CDN.
- Il cliente invia il contenuto da distribuire a un nodo CDN, che a sua volta lo copia in tutti i suoi server CDN.
- L'azienda fornitrice di CDN può possedere una rete privata per trasferire il contenuto dal nodo CDN ai server CDN.
- Ogni server CDN contiene file di molti diversi fornitori di contenuti.

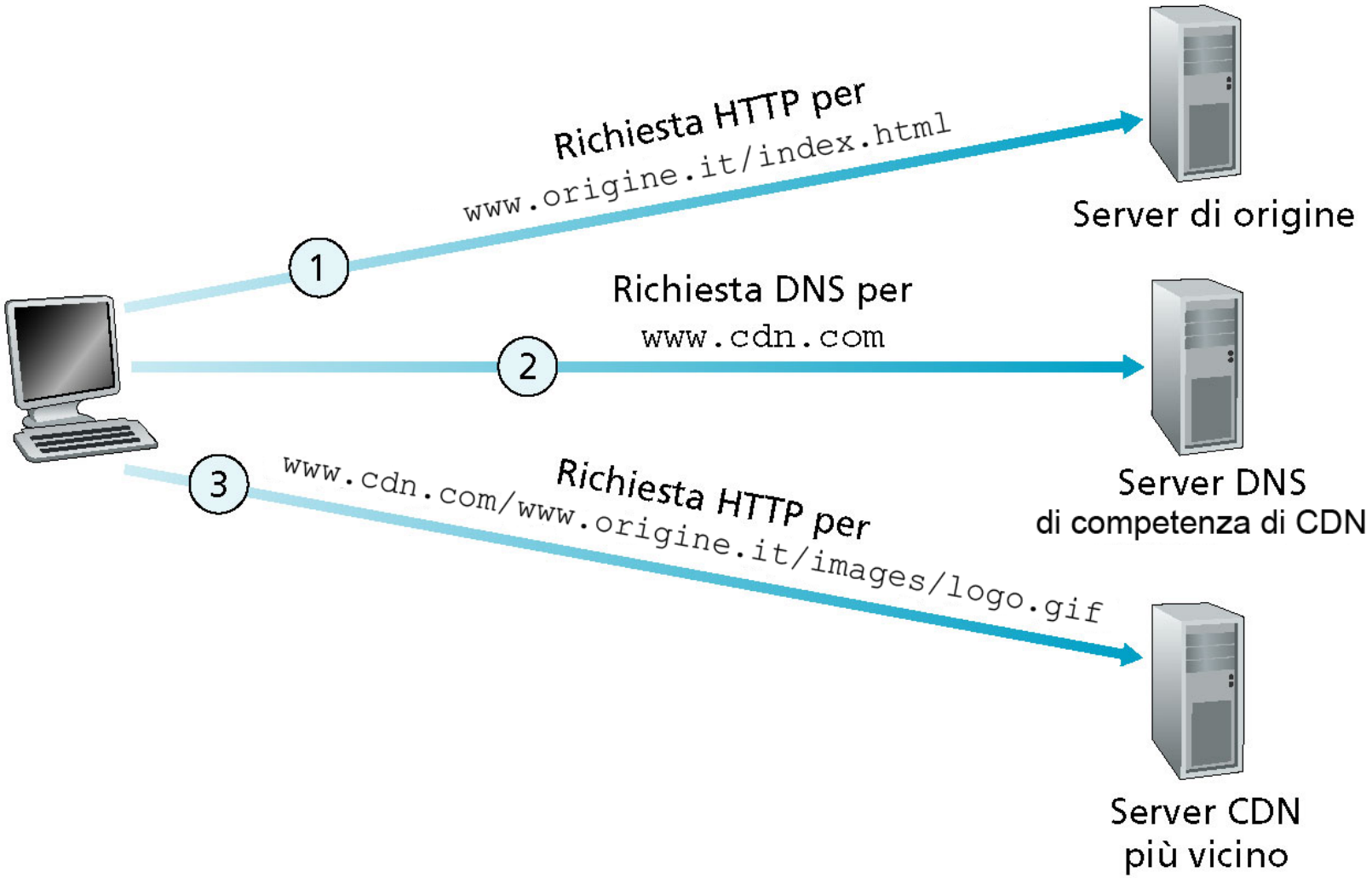
- Per esempio, supponiamo che l'hostname del cliente sia **www.origine.it**. Supponiamo che il nome dell'azienda fornitrice di CDN sia **www.cdn.com**. Supponiamo inoltre che il cliente voglia che solo i file multimediali siano distribuiti dalla CDN mentre tutti gli altri file, compresi le pagine base HTML, sono distribuite direttamente dal cliente stesso.
- Per ottenere questo, il cliente modifica tutti i riferimenti a file multimediali nel suo server di origine in modo che gli URL dei file multimediali abbiano il prefisso **http://www.cdn.com**.
- In questo modo, se un file HTML del cliente aveva originariamente un riferimento

  <img src= http://www.origine.it/images/logo.gif>,

  il cliente scriverà il riferimento nel file HTML con l'URL:

  <img src= http://www.cdn.com/www.origine.com/images/logo.gif >

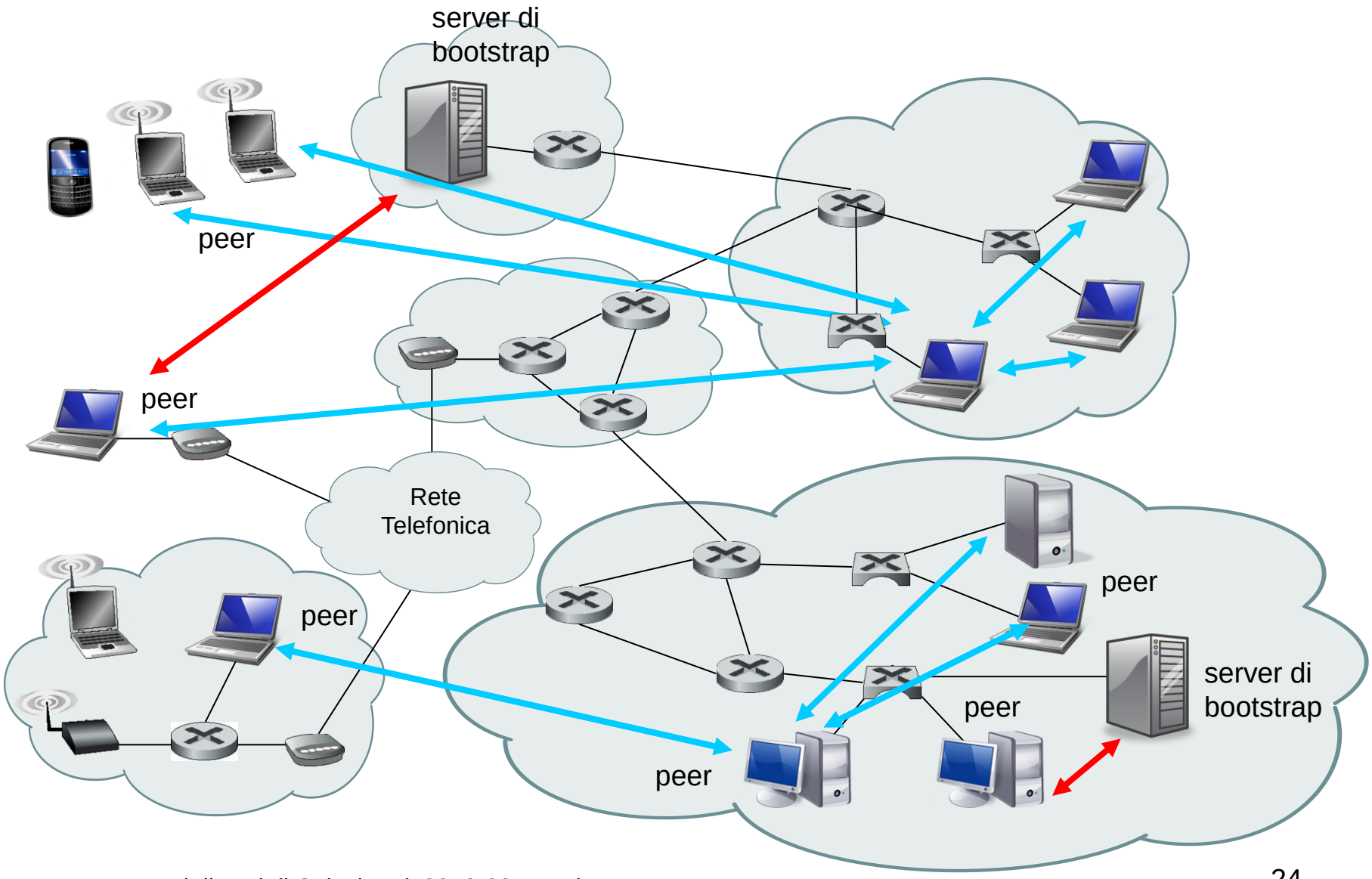
- Quando un browser richiede una pagina Web, ad esempio **index.html**, contenente l'immagine **logo.gif**, avvengono le seguenti azioni:
  - Il browser richiede la pagina index.html al server di origine **www.origine.it** il quale la invia al browser. Il browser esamina il file HTML index.html e trova il riferimento a **http://www.cdn.com/www.origine.it/images/logo.gif**
  - Il browser allora invia un messaggio di richiesta **DNS** per **www.cdn.com**. Quando la richiesta giunge al server DNS di competenza di **www.cdn.com**, esso utilizzando **una mappa interna della rete che ha costruito per l'intera Internet**, restituisce l'indirizzo IP del server CDN che è il migliore per il browser richiedente che generalmente è il server CDN più vicino al browser richiedente.
  - Il browser richiedente riceve una risposta DNS con l'indirizzo IP del server CDN migliore e invia a questo la sua richiesta HTTP relativa al file logo.gif a questo server CDN.
- Per le richieste successive a www.cdn.com il client continua a usare lo stesso server CDN, dato che l'indirizzo IP ottenuto per www.cdn.com è nella cache del DNS locale del client.


Richiesta HTTP per
www.origine.it/index.html
1
Server di origine
Richiesta DNS per
www.cdn.com
2
Server DNS
di competenza di CDN
3
www.cdn.com/www.origine.it/images/logo.gif
Richiesta HTTP per
Server CDN
più vicino

Applicazioni P2P (da pari a pari)

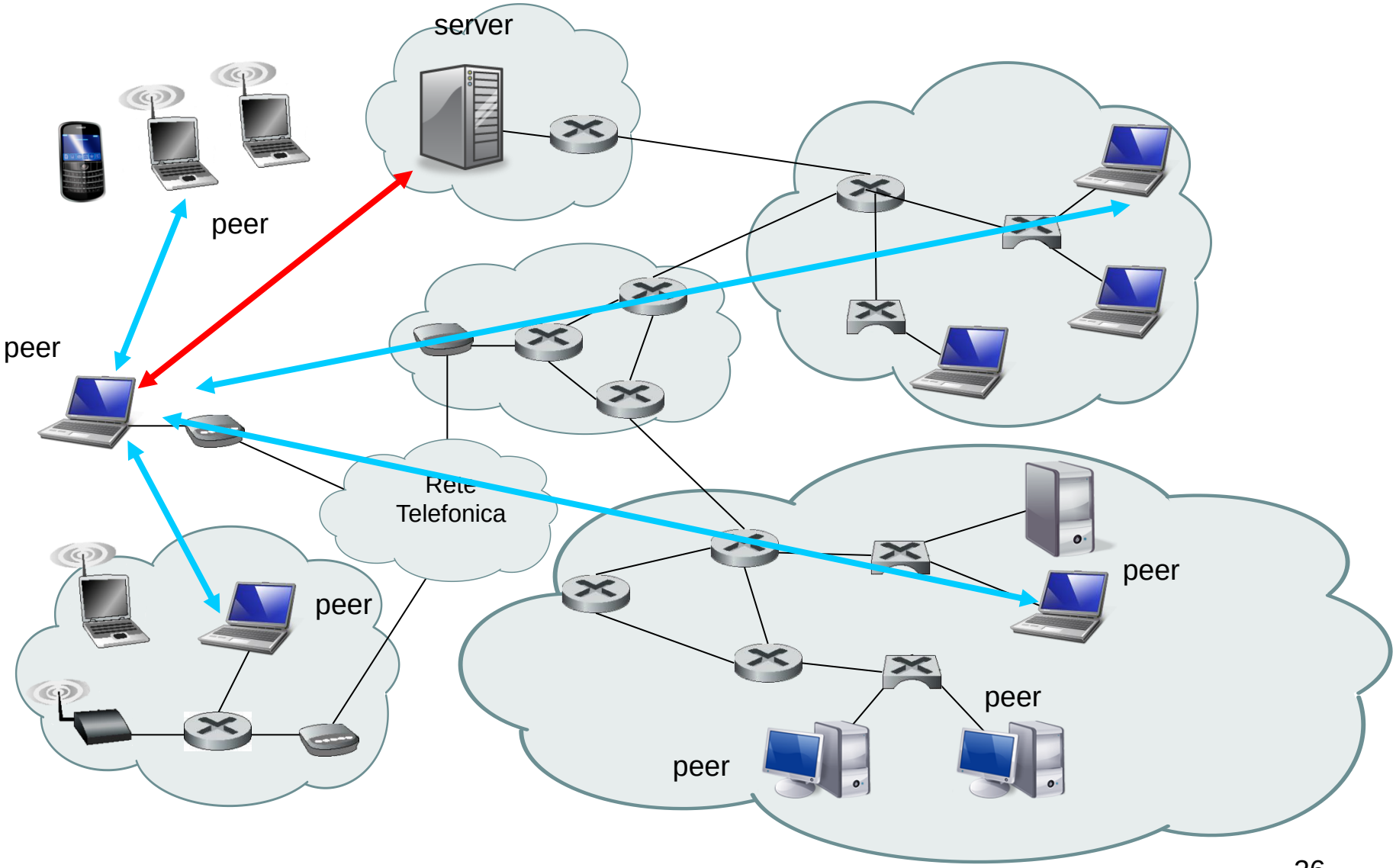
- I protocolli e le applicazioni fin ora descritti sono stati progettati con architettura client/server.
- Un altro modello, più complesso, per la progettazione di applicazioni di rete è il Peer to Peer (P2P).
- Questa architettura consente agli host di scambiarsi file direttamente tra loro, senza passare attraverso server intermediari.
- Gli host in questa architettura sono detti **pari**.
- In ogni pari è implementato sia il lato client sia il lato server del protocollo di trasferimento di file che generalmente è l'HTTP (a volte FTP) poiché un pari è sia un distributore che un fruitore di contenuti.
- Nel P2P, ci sono anche aspetti importanti legati alla violazione del copyright, della privacy, e della sicurezza.
- In un sistema P2P in ogni istante sono connessi un gran numero di pari, e ogni pari può avere molti file da condividere.

- Un importante servizio che un'applicazione P2P deve fornire agli utenti è la **ricerca dei contenuti**. Se un pari **P** vuole ottenere un particolare file, ad esempio un brano musicale, allora quel pari **P** deve conoscere gli indirizzi IP dei pari connessi che hanno copie del file desiderato.
- Vedremo tre architetture per ricercare i file, ciascuna delle quali è stata usata da diversi sistemi di condivisione di file P2P.
- Nella figura seguente è mostrata l'architettura P2P.


server di
bootstrap
peer
peer
Rete
Telefonica
peer
peer
peer
peer
server di
bootstrap

Database centralizzato

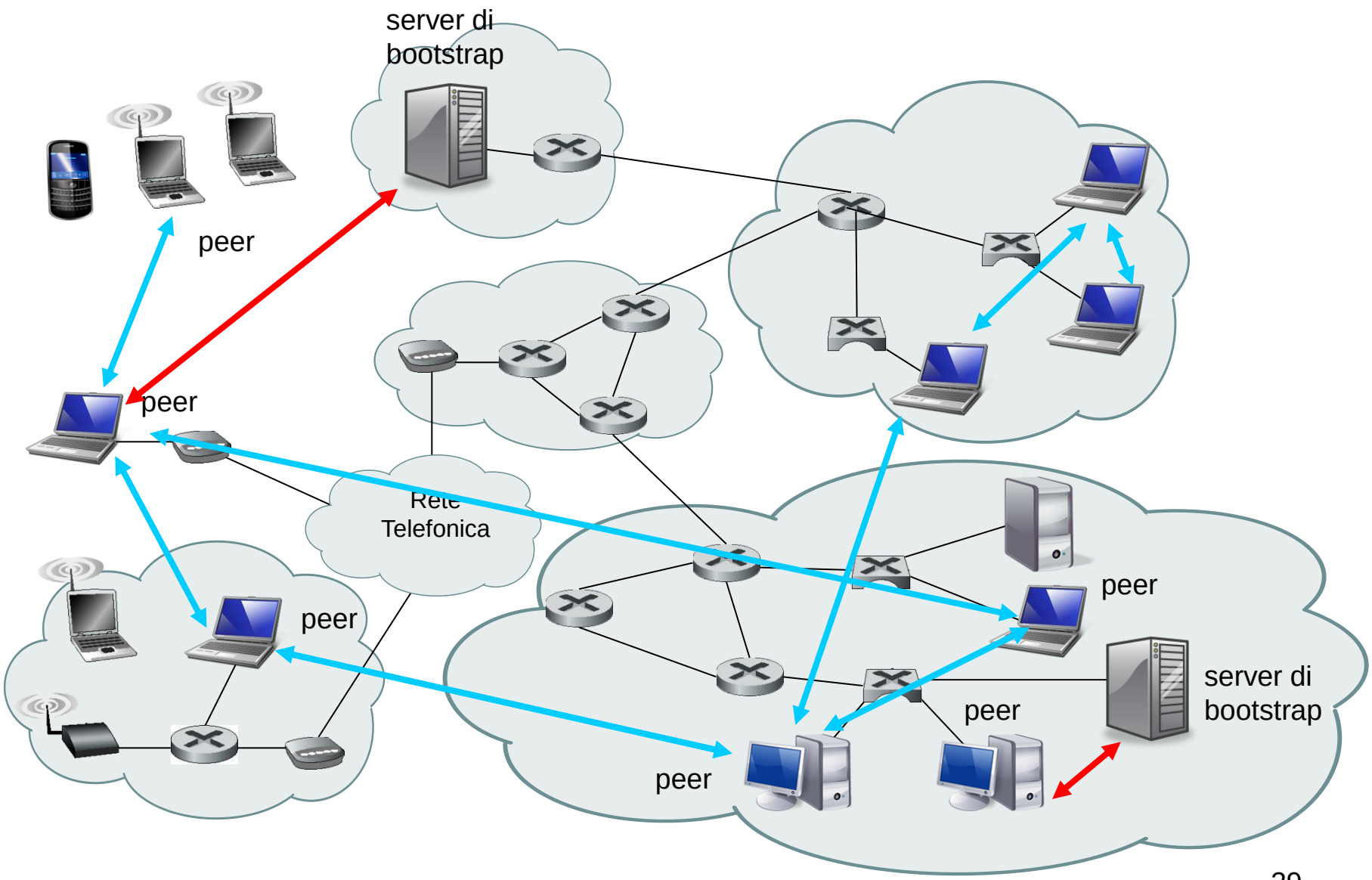
- Una delle soluzioni più semplici per ricercare i file da scaricare si ottiene mediante un server che gestisce un database centralizzato, come nel caso di Napster.
- Come è mostrato nella figura, al momento dell'avvio, l'applicazione P2P si connette a un server inviando ad esso i nomi dei file contenuti nel sua directory locale condivisa.
- Il server memorizza le informazioni provenienti da tutti i pari che si connettono al sistema, creando in tal modo un database centralizzato, dinamico che associa ogni nome di file con un insieme di indirizzi IP.
- Quando un pari scarica o aggiunge un nuovo file nella cartella condivisa, o rimuove un file, informa il server in modo che esso possa aggiornare il suo database.

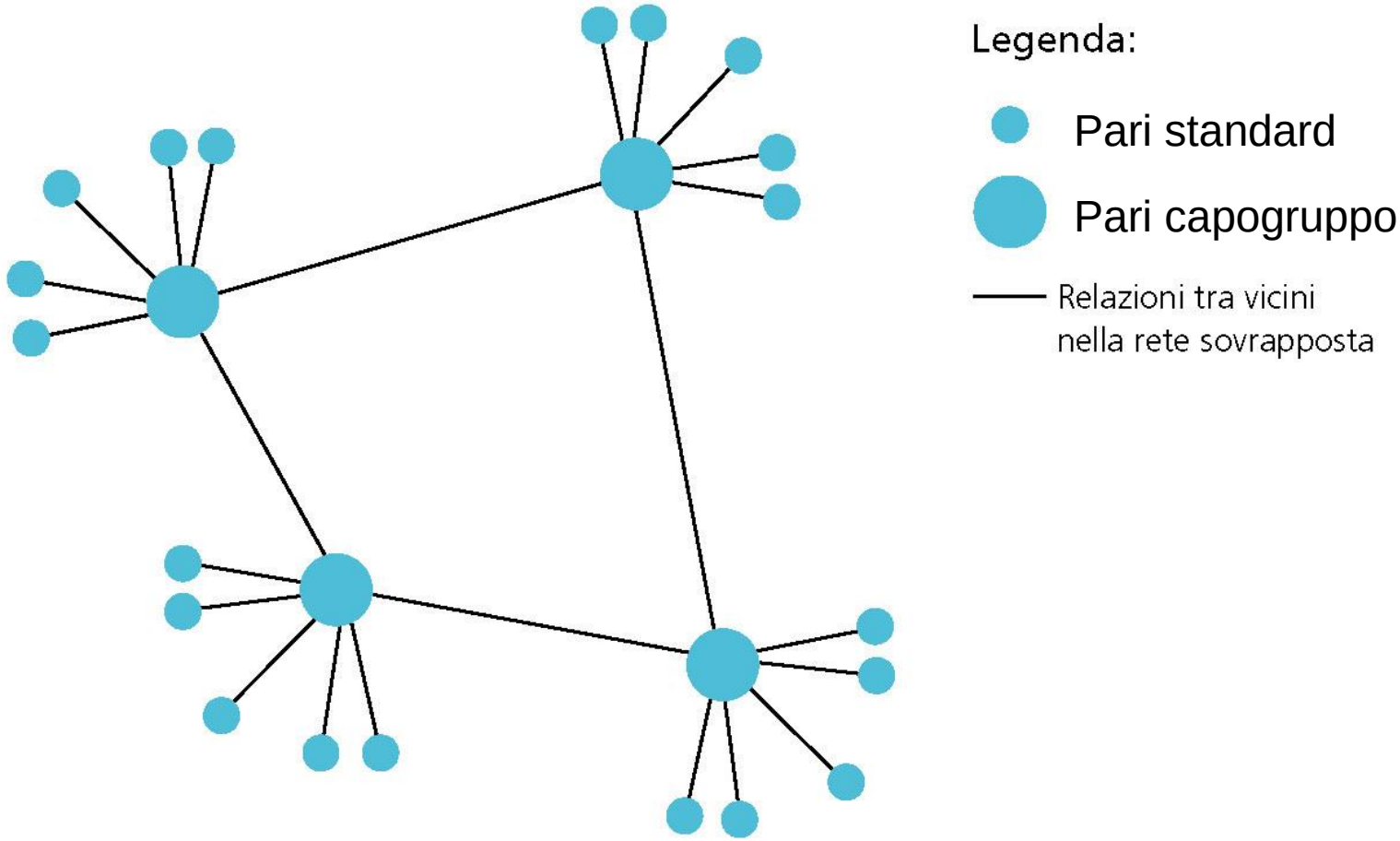

server
peer
peer
Rete
Telefonica
peer
peer
peer
peer

- Per mantenere aggiornato il database il server deve sapere quando un pari si disconnette dalla rete.
- Un modo per conoscere quali pari sono connessi è di inviare periodicamente messaggi ai pari per verificare se sono connessi. Se il server rileva che un pari non è più connesso, rimuove il suo indirizzo IP dal database.
- Usare una directory centralizzata per localizzare i file ha una serie di svantaggi:
  - ***Unico punto di guasto***. Se il server non funziona, allora si blocca l'intera applicazione P2P.
  - ***Collo di bottiglia per le prestazioni***. In un grande sistema P2P, con centinaia di migliaia di utenti connessi, un server centralizzato deve mantenere aggiornato un enorme database e deve rispondere a migliaia di richieste al secondo, pertanto le prestazioni del servizio sono limitate da problemi di sovraccarico e di traffico.

Database distribuito

- Il database per la ricerca dei contenuti è distribuito tra un insieme di pari detti **capogruppo** (esempio FastTrack).
- Un pari quando si connette alla rete P2P si connette con qualche altro pari già appartenente alla rete.
- Pertanto, il sistema P2P deve avere uno o più **server di bootstrap** sempre accesi, che svolgono il compito di gestire e mantenere la topologia della rete P2P.
- Quando un pari si connette alla rete P2P, prima contatta un server **di bootstrap** il quale può assegnargli il ruolo di capogruppo, oppure inviargli l'indirizzo IP di uno dei capogruppo già esistenti, e quindi il pari crea una connessione con quel capogruppo.
- Se il pari è stato nominato capogruppo, il server di bootstrap invia ad esso gli indirizzi IP di alcuni (o tutti) gli altri capigruppo.
- Il capogruppo gestisce, per tutti i pari del suo gruppo, un database che associa i file con gli indirizzi IP.
- Con questa architettura, la rete è costituita da un insieme di sistemi P2P con database centralizzato, in cui ogni capogruppo ha un ruolo di server, ma è anche un pari.

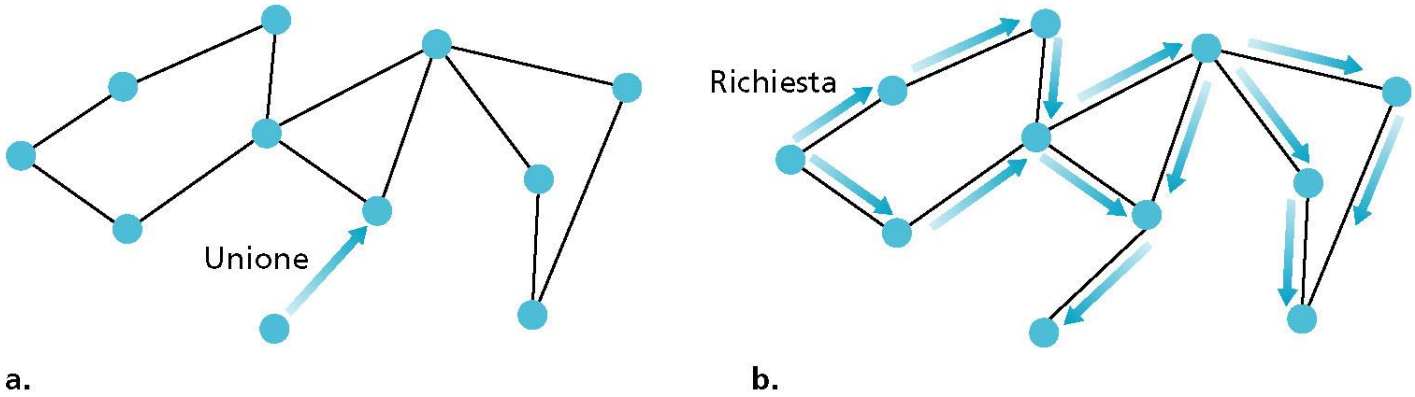

server di
bootstrap
peer
peer
Rete
Telefonica
peer
peer
peer
peer
server di
bootstrap


Legenda:
Pari standard
Pari capogruppo
Relazioni tra vicini
nella rete sovrapposta

- Pertanto, le dimensioni dei database sono più piccole e ciascun capogruppo sarà meno sovraccaricato di richieste e il traffico sarà più distribuito.
- L'insieme di pari e delle loro connessioni logiche, formano una rete logica, detta **rete di copertura (o sovrapposta)** *(overlay network)*.
- Poiché sia i pari che i capogruppo si connettono e disconnettono continuamente, la rete di copertura cambia dinamicamente.
- Questa architettura richiede di realizzare un protocollo molto complesso per gestire la rete di copertura. Ad esempio, quando un capogruppo si disconnette, è necessario assegnare tutti i pari del capogruppo ad altri capigruppo.
- Nella fase di ricerca di un file, il pari invia la richiesta al proprio capogruppo il quale a sua volta la rinvia agli altri capogruppo per ottenere un elenco completo dei pari che hanno il file richiesto. In questo modo il pari richiedente ottiene molte liste di pari che hanno il file richiesto.

Inondazione di richieste (query flooding)

- L'applicazione P2P open source Gnutella usa un approccio completamente distribuito per realizzare la ricerca dei file.
- In Gnutella, non esiste una struttura gerarchica con pari e capigruppo ma i pari si uniscono in una rete sovrapposta senza gerarchia.
- Anche Gnutella utilizza nodi di bootstrap sempre accesi.



- Un pari si unisce alla rete P2P, **connettendosi a un server di bootstrap** che comunica al pari gli indirizzi IP di un certo numero di pari appartenenti alla rete di copertura.
- Ogni pari conosce solo i pari suoi vicini. Anche per la realizzazione di questa architettura è richiesto un protocollo complesso per mantenere aggiornata la rete sovrapposta, poiché i pari si connettono e disconnettono continuamente.
- Per la ricerca dei file, Gnutella usa la tecnica detta **inondazione di richieste *(query flooding).***
- Con tale tecnica, quando un pari ricerca un file, invia una richiesta a tutti i suoi vicini, ciascuno dei quali rinvia la richiesta a tutti i pari suoi vicini, meno che al richiedente. In questo modo, la richiesta arriva ad ogni pari appartenente alla rete sovrapposta. Se un pari che riceve la richiesta ha una copia del file desiderato, invia una risposta al pari richiedente, indicando che possiede una copia del file.
- La tecnica del **flooding** crea una elevato traffico di richieste, per limitare il quale, si stabilisce un limite massimo di livello di propagazione delle richieste.

- Ciò si realizza con un campo ***contatore di nodo*** presente nel messaggio di richiesta, che inizialmente è impostato ad un valore predefinito, ad esempio 9. Ogni volta che il messaggio di richiesta arriva ad un pari, il campo contatore viene decrementato di uno prima che il messaggio di richiesta sia rinviato al suoi vicini. Quando un pari riceve una richiesta con il campo contatore uguale a zero, non la inoltra.

BitTorrent

- BitTorrent è il protocollo P2P per la distribuzione di file attualmente più diffuso. In BitTorrent, l'insieme di tutti i pari che partecipano alla distribuzione di un determinato file è detto **torrent** *(torrente).*
- I pari appartenenti a un torrent si scambiano parti *(chunk)* del file tra loro. La dimensione tipica dei chunk è 256 kbyte.
- Quando un pari si associa a un torrent per la prima volta, non ha parti del file. Col passare del tempo, memorizza sempre più parti che, mentre scarica, invia agli altri pari.
- Un pari può disconnettersi dal torrent in qualsiasi momento con solo alcune parti del file e riconnettersi al torrent successivamente.
- Ciascun torrent è gestito da un nodo di infrastruttura detto **tracker**.
- Quando un pari entra a far parte di un torrent, si registra presso il tracker e periodicamente invia messaggi al tracker per comunicare la propria appartenenza al torrent.

- In questo modo, il tracker tiene traccia dei pari che appartengono al torrent. In un dato istante, un torrent può avere migliaia di pari.
- Quando un nuovo pari **P**, entra a far parte di un torrent, il tracker seleziona in modo casuale un sottoinsieme di pari (ad esempio 50) dall'insieme dei pari che appartengono a quel torrent, e invia l'indirizzo IP di questi 50 pari al nuovo pari. Avendo la lista dei pari, il nuovo pari **P** cerca di stabilire delle connessioni TCP con tutti i pari della lista. Chiamiamo i pari con i quali il nuovo pari riesce a stabilire una connessione TCP ***"pari vicini"***.
- Col passare del tempo, alcuni di questi pari possono disconnettersi dal torrent, mentre altri possono cercare di stabilire una connessione TCP con **P**. Quindi i pari vicini a uno specifico pari cambiano nel tempo.
- In un certo istante, ciascun pari avrà un sottoinsieme delle parti di un file e pari diversi avranno differenti sotto insiemi.

- Periodicamente il nuovo pari chiederà a ciascuno dei suoi vicini, sulle connessioni TCP, la lista delle parti del file in loro possesso.
- Tramite questa conoscenza, il pari P invierà richieste, di nuovo sulle connessioni TCP, per le parti del file che ancora non possiede.
- Pertanto, in un dato istante il pari P avrà alcune parti del file e saprà quali parti hanno i suoi vicini. Con queste informazioni, il pari P deve prendere due importanti decisioni: primo, quali parti deve richiedere per prime ai suoi vicini e, secondo, a quali vicini dovrebbe inviare le parti richieste. Nella decisione, P adotta la tecnica «il più raro per primo». L'idea è determinare, tra le parti che ancora non ha, quelle che sono più rare; cioè le parti che sono meno duplicate tra i suoi vicini, e richiederle per prime. In questo modo, le parti più rare sono redistribuite più velocemente, cercando di rendere uguale il numero di copie di ciascuna parte nel torrent,

Per determinare a quali richieste P debba rispondere, BitTorrent usa un intelligente algoritmo scambio il quale stabilisce che P assegni una priorità più alta ai vicini che, in un dato istante, inviano dati alla velocità più alta. Pe fare questo, P misura continuamente la velocità di trasferimento per ciascuno dei suoi vicini, e individua i quattro pari da cui riceve dati alla velocità più alta. P poi contraccambia inviando le parti del file a quegli stessi quattro pari. Ogni 10 secondi ricalcola le velocità ed eventualmente modifica l'insieme dei quattro pari. Un punto importante è che ogni 30 secondi sceglie casualmente un vicino in più e invia ad esso le parti. Indichiamo con Q il nuovo pari scelto casualmente. Dato che P sta inviando dati a Q, potrebbe diventare uno dei quattro "uploader" principali di Q, nel qual caso Q inizierebbe a inviare dati a P. Se la velocità alla quale Q manda i dati a P è abbastanza alta, Q, a sua volta, potrebbe diventare uno dei quattro "uploader" principali di P.

In breve, ogni 30 secondi P sceglie casualmente un nuovo vicino e inizia a scambiare parti con esso. Se lo scambio di dati è soddisfacente, ciascun pari metterà l'altro nella propria lista dei quattro "uploader" principali e continueranno a scambiarsi parti tra loro fino a che uno non trova un partner migliore. La selezione casuale dei vicini consente anche a nuovi pari di avere le parti del file, in modo che abbiano qualcosa da scambiare. Tutti gli altri pari, a parte quei cinque (i quattro pari "top" e un pari di prova), non ricevono alcune parti da P.

- Nella condivisione di file P2P, un problema importante è il free-riding (sfruttamento), nel quale un pari scarica file dal sistema di condivisione, senza inviarne. L'algoritmo di scambio di BitTorrent elimina virtualmente il problema del free-riding, in quanto, se P vuole scaricare bit da Q a una velocità soddisfacente per un periodo di tempo esteso, deve nello stesso tempo inviare bit a Q a una velocità decente.