



## Lezione n.4

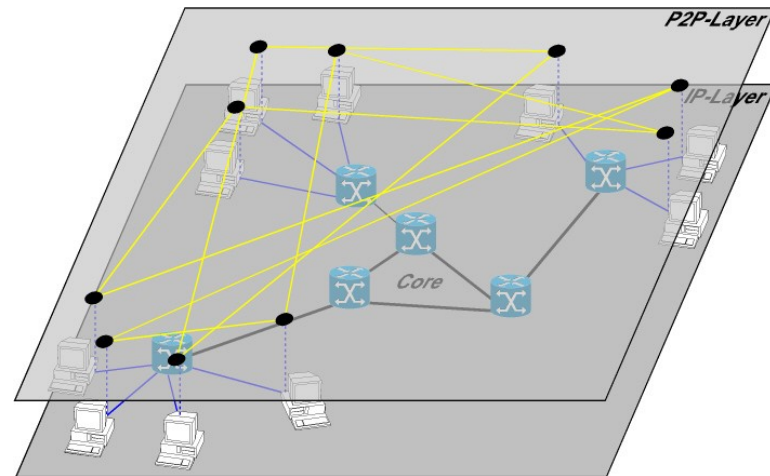
# Sistemi P2P di Prima Generazione: Sistemi Centralizzati

Laura Ricci

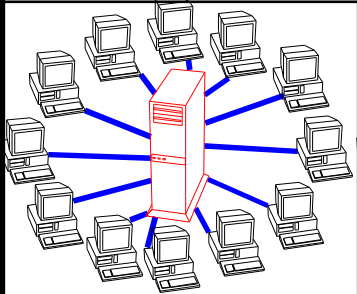
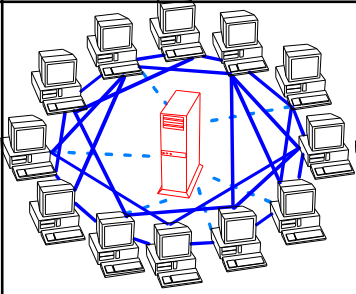
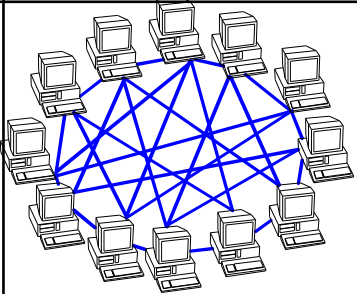
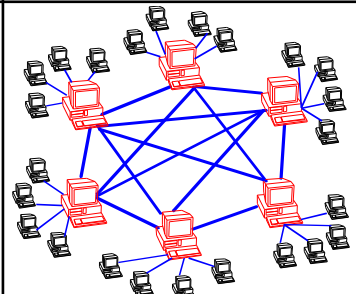
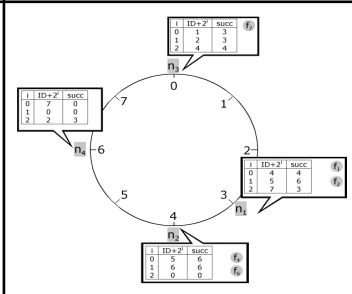
Peer to Peer systems and Applications  
Capitolo 5

# P2P: Sistemi di Prima e di Seconda Generazione

1. **Caratteristiche dei primi sistemi P2P**
2. **Reti Peer-to-Peer Centralizzate**
  1. Caratteristiche base
  2. Protocolli
  3. Discussione
3. **Reti Peer-to-Peer Completamente Decentralizzate**
  1. Caratteristiche Base
  2. Protocolli
  3. Discussione
4. **Reti Peer-to-Peer Ibride**
  1. Caratteristiche Base
  2. Protocolli
  3. Discussione



# P2P: Classificazione Generale

<b>Client-Server</b>	<b>Peer-to-Peer: Caratteristiche Generali</b>			
<ol style="list-style-type: none"> <li>Il Server è l'unica entità in grado di memorizzare l'informazione condivisa → La rete è gestita dal Server in modo centralizzato</li> <li>Il Server è il Sistema con la maggior capacità di calcolo</li> <li>I client posseggono minor capacità di calcolo</li> </ol> <p><b>Esempio: WWW</b></p>	<ol style="list-style-type: none"> <li>Le risorse sono condivise tra i peers.</li> <li>Le risorse offerte da un peer possono essere accedute direttamente da altri peer</li> <li>Il peer fornisce e richiede risorse (Servent)</li> </ol>			
	<b>P2P Non strutturati</b>			<b>P2P Strutturati</b>
	<i>Centralized P2P</i>	<i>Pure P2P</i>	<i>P2P Ibridi</i>	<i>DHT-Based</i>
	<ul style="list-style-type: none"> <li>Soddisfano 1.,2.,3.</li> <li>Esiste una entità centralizzata</li> <li>Tale entità centralizzata svolge solo compiti di indicizzazione delle risorse</li> </ul> <p><b>Esempio: Napster</b></p>	<ol style="list-style-type: none"> <li>Soddisfano 1.,2.,3.</li> <li>Non esiste alcuna entità centralizzata</li> <li>La funzionalità del sistema non viene compromessa dall'eliminazione di un peer</li> </ol> <p><b>Esempi: Gnutella 0.4,</b></p>	<ol style="list-style-type: none"> <li>Soddisfano 1.,2.,3.</li> <li>Esistono alcuni peer (superpeer) che hanno anche funzioni di indicizzazione</li> <li>I superpeer sono determinati dinamicamente</li> </ol> <p><b>Esempi: Gnutella 0.6, JXTA</b></p>	<ol style="list-style-type: none"> <li>Soddisfano 1.,2.,3.</li> <li>Non esistono entità centralizzate</li> <li>La overlay network risulta strutturata</li> <li>Overlay network strutturata</li> </ol> <p><b>Esempi: Chord, CAN</b></p>
	<b>1<sup>st</sup> Gen.</b>		<b>2<sup>nd</sup> Gen.</b>	
				

# P2P: Sistemi di Prima e di Seconda Generazione

- L'informazione condivisa:

E' distribuita sulla rete in *modo non deterministico*, possono esistere *diverse repliche* di un'informazione.

in generale viene memorizzata sui nodi che la mettono a disposizione della rete

- I peer definiscono una *overlay network*, o *signaling network*, utilizzata per scambiare messaggi di controllo, ma non per lo scambio dei dati

Inserimento/cancellazione di peer

Verifica periodica di altri peers (keep-alive)

Ricerca di dati

- Il trasferimento dei dati:

Avviene su connessioni distinte da quelle definite dalla *overlay network*

In genere mediante il protocollo HTTP

- Connessioni basate su TCP/IP

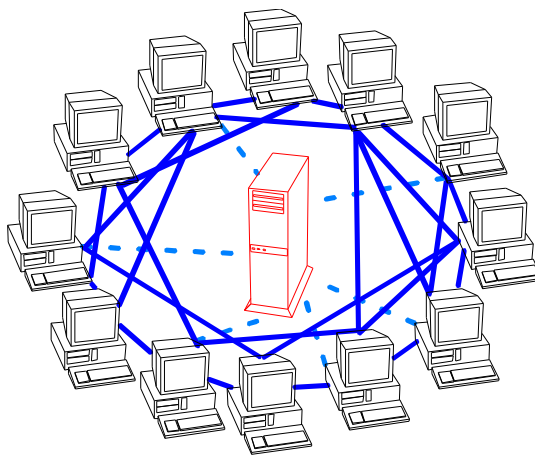
- Applicazione tipica: condivisione dei files

# P2P:Sistemi di Prima e Seconda Generazione

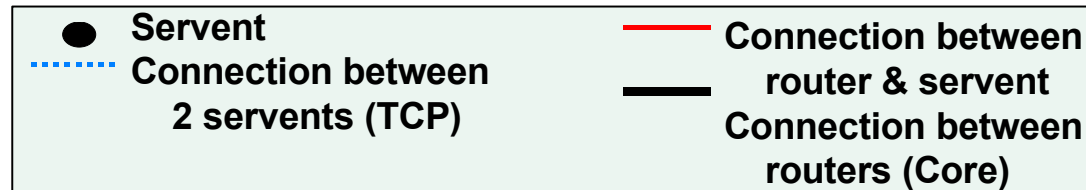
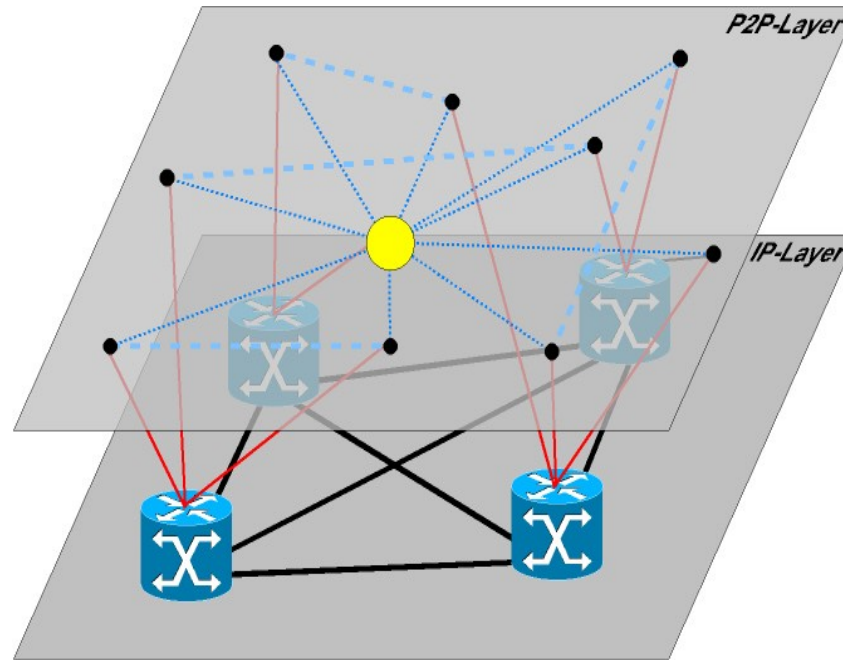
- **Overlay Topology:** rete virtuale stabilita tra i peer mediante connessioni TCP per lo scambio di informazione di controllo
- **Caratteristiche della overlay network**
  - completamente indipendente dalla rete fisica, grazie al livello di astrazione TCP/IP può essere strutturata in modo gerarchico (Gnutella Superpeers, JXTA rendez-vous peers)
  - può includere un server centralizzato (look up server di Napster)
- **I peer partecipano attivamente alla overlay network**
  - Forniscono informazione
  - Ricercano informazione
  - Svolgono funzioni di routing

# Sistemi P2P Centralizzati:Napster

- Tutti i peer si connettono ad un server centralizzato (broker). La topologia dell'overlay network risultante è a stella (*star network*)
- Le connessioni tra i peers sono stabilite dinamicamente 'on demand' per lo scambio dei dati



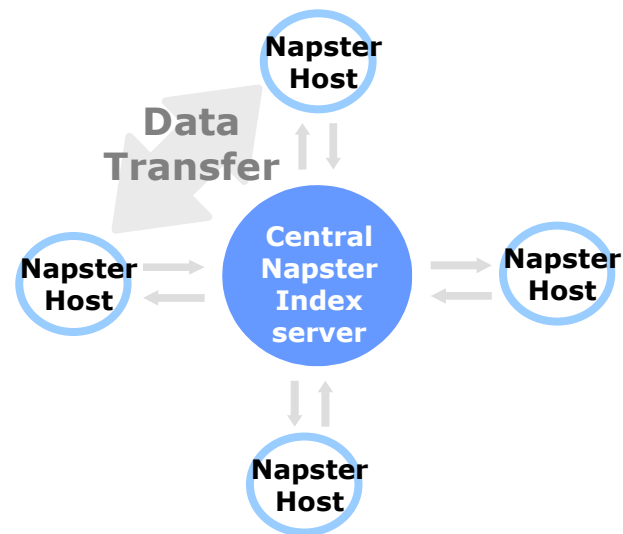
# Topologia di Napster



# Napster: Principi di Funzionamento

## Comportamento del peer (ad alto livello)

- Si connette al Server Napster
- Invia al Server (push) le informazioni relative ai files musicali che intende condividere (metadata). Il server memorizza queste informazioni in un database centralizzato
- Ricerca un file, fornendo una lista di keywords inviate al server
- Riceve un insieme di coppie (file-peer) e sceglie una coppia in base a qualche criterio (bitrate, frequency, tipo di collegamento)
- Si connette al peer prescelto e scarica il file (connessione HTTP)





# Napster: Principi di Funzionamento

- Altre funzionalità

  - Instant messaging

  - Chats tra peers

  - Hot lists

- A causa della popolarità di NAPSTER, un unico server centralizzato si è rilevato presto insufficiente. Architetture alternative

  - definizione di un server farm. Si introduce un insieme di brokers. L'interfaccia verso l'esterno rimane comunque unico (single point of failure)

  - introduzione di un insieme di broker Napster. Ogni broker gestisce un proprio database

    - Non c'è condivisione di dati tra i diversi database

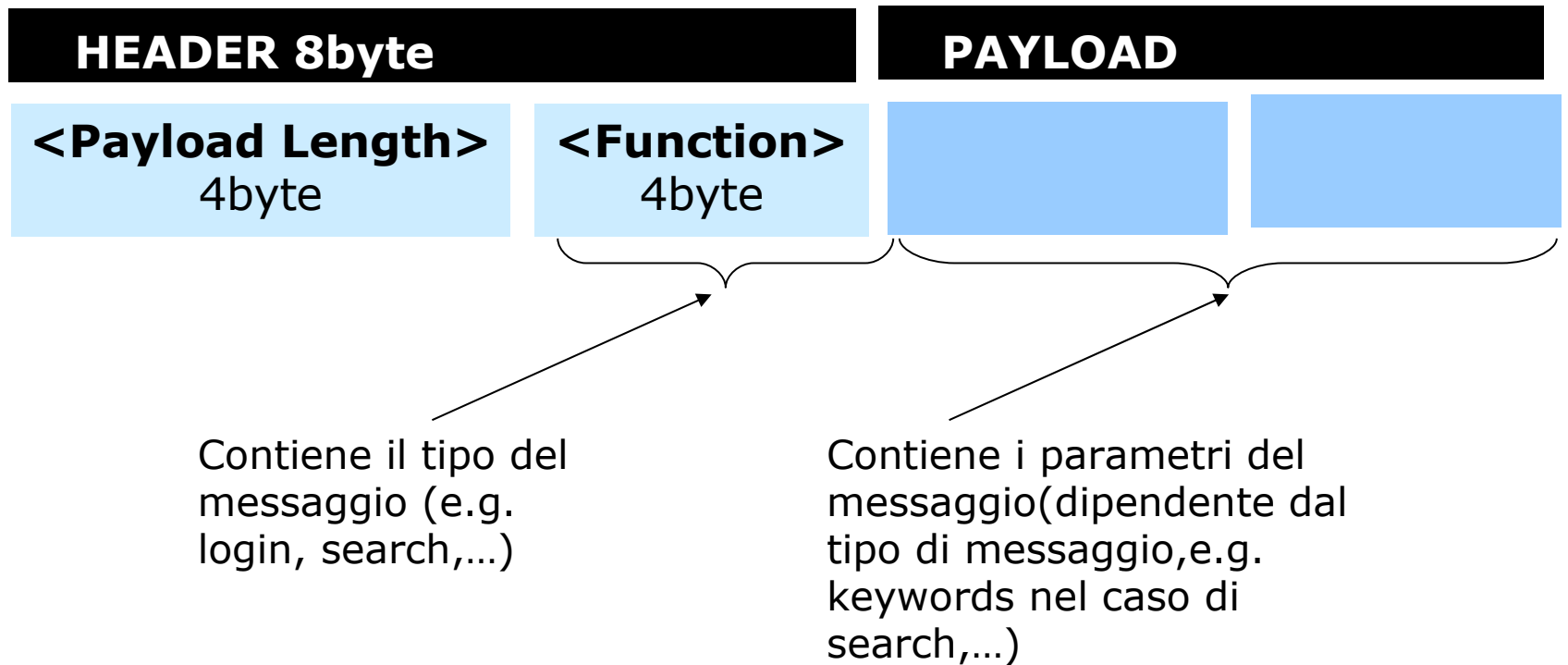
    - Ogni broker gestisce una diversa rete Napster (più reti centralizzate)

    - Bilanciamento del carico effettuato da un insieme di look-up servers

  - Coordinamento dei diversi brokers (verso le reti P2P ibride)

# NAPSTER: Struttura dell'Header

Messaggi scambiati tra il peer ed il Server NAPSTER: header



# Napster: Inizializzazione

## Client/Server Service

### 1: LOGIN (Function:0x02)

<Nick> <Password> <Port> <Client-Info> <Link-type>

### 2: LOGIN ACK (Function: 0x03)

### 3: NOTIFICATION OF SHARED FILE (0x64)

„<Filename>“ <MD5> <Size> <Bitrate> <Freq> <Time>



# Napster: Inizializzazione

## Client/Server Service

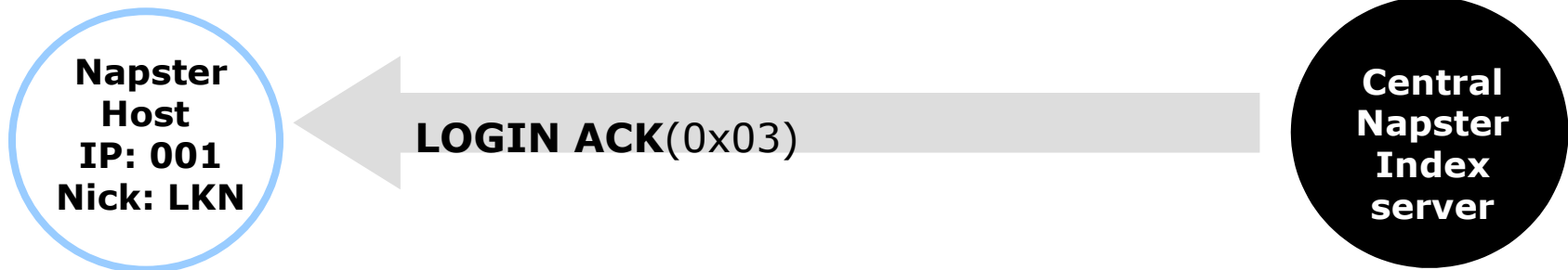
### 1: LOGIN (Function:0x02)

<Nick> <Password> <Port> <Client-Info> <Link-type>

### 2: LOGIN ACK (Function: 0x03)

### 3: NOTIFICATION OF SHARED FILE (0x64)

„<Filename>“ <MD5> <Size> <Bitrate> <Freq> <Time>



# Napster: Inizializzazione

## Client/Server Service

### 1: LOGIN (Function:0x02)

<Nick> <Password> <Port> <Client-Info> <Link-type>

### 2: LOGIN ACK (Function: 0x03)

### 3: NOTIFICATION OF SHARED FILE (0x64)

„<Filename>“ <MD5> <Size> <Bitrate> <Freq> <Time>



### NOTIFICATION(0x64)

„hang-up.mp3“ 3f3a3... 5674544  
128 44100 342



**Central Napster Index server**

# Napster:Ricerca

## 1: SEARCH (Function: 0xC8)

[FILENAME CONTAINS „Search Criteria“] [MAX\_RESULT <Max>]

[LINESPEED <Compare> <Link-Type>]

[BITRATE <Compare> „<Bitrate>“] [FREQ <Compare> „<Freq>“]

## 2: SEARCH RESPONSE (Function: 0xC9)

„<Filename>“ <MD5> <Size> <Bitrate> <Freq>

<Time> <Nick> <IP> <Link-Type>



### SEARCH(0xC8)

FILENAME CONTAINS „greendays“ MAX\_RESULTS 100

LINESPEED „AT LEAST“ 6 BITRATE „AT LEAST“ „128“

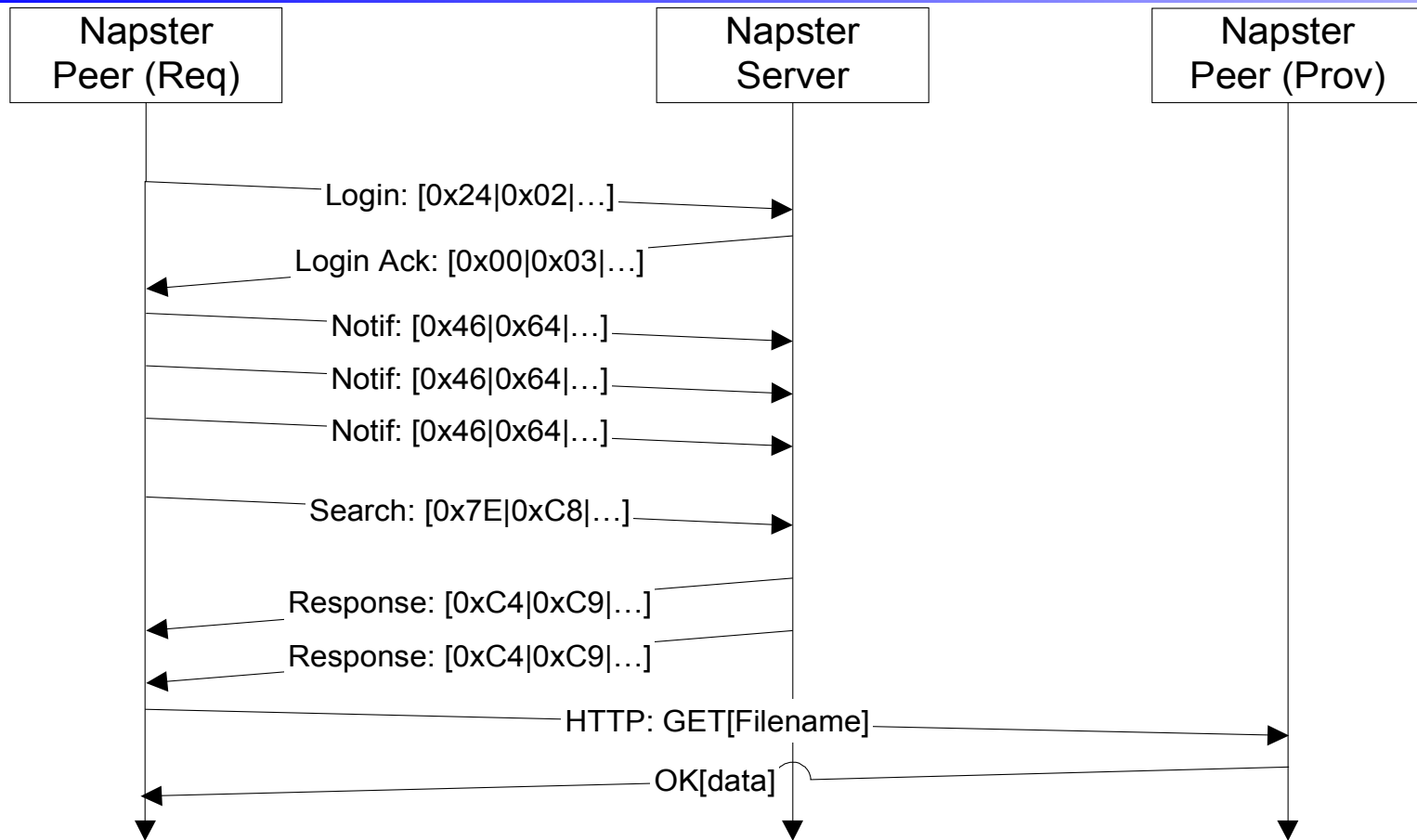
FREQ „EQUAL TO“ „44100“



## Altri sistemi P2P basati su server centralizzato

- Audiogalaxy
- WinMX
- BiTorrent

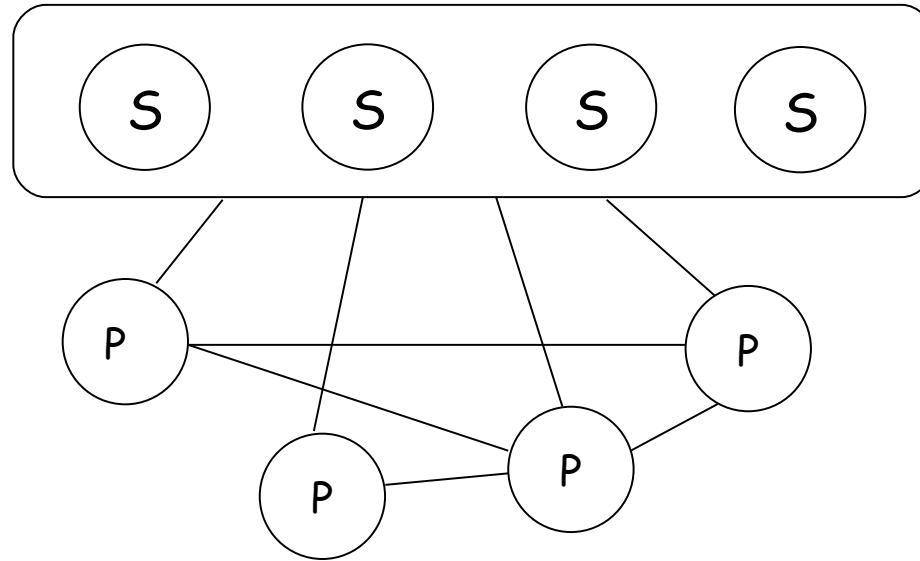
# Napster: il protocollo semplificato



Sequenza semplificata (**rispetto al protocollo reale**) di messaggi di una rete Napster con due peer. Il peer req richiede un file al Peer Prov



# Napster: Architettura

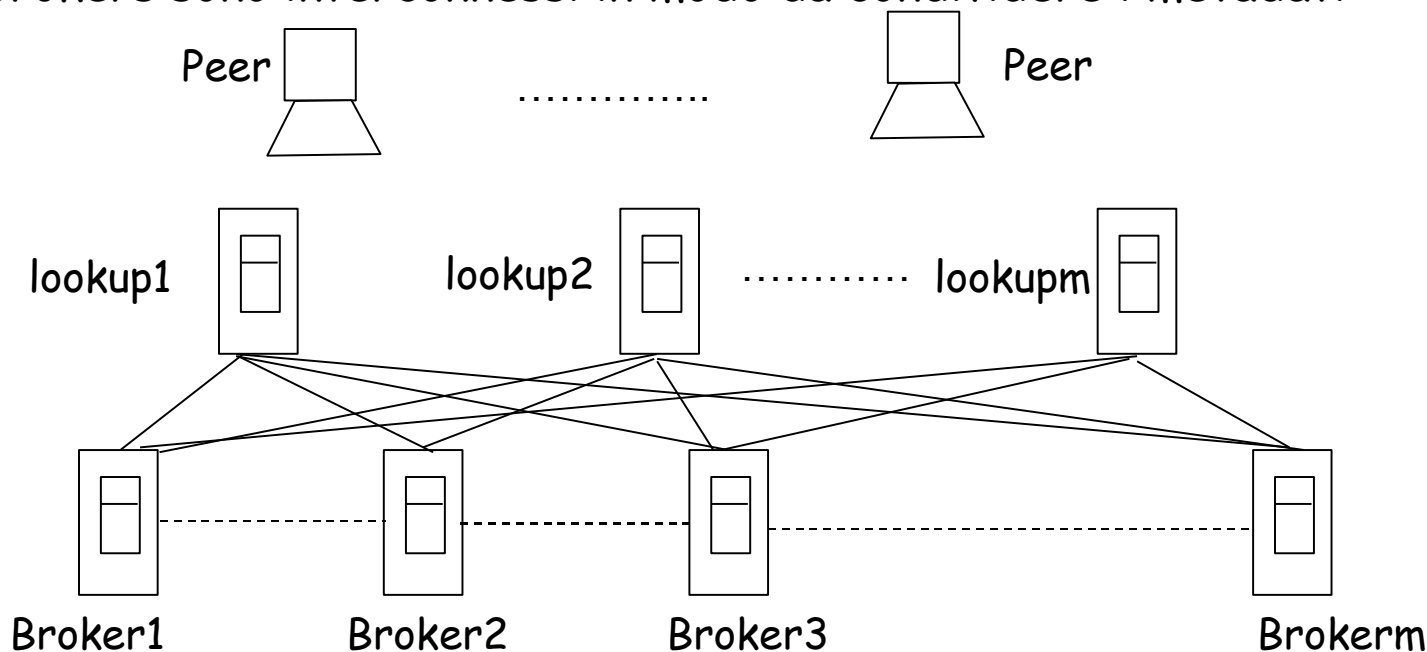


- Definizione di un cluster di server (server farm) per aumentare la scalabilità del sistema
- I server sono localizzati in **un'unica locazione geografica**
- Esiste comunque un unico punto di accesso al sistema (esempio i router che collega il cluster alla rete)
- i server condividono il database contenente le meta-informazioni

# Napster: Architettura

Bilanciamento del carico: vengono definiti  $m$  lookup servers ed  $n$  brokers

- Brokers: gestiscono i metadata (informazioni sui files)
- Look-up servers: ricevono le richieste dagli utenti e smistano le richieste ai broker in modo da bilanciare il carico. Ogni look-up server è collegato ad ognuno dei broker per conoscere il suo carico
- I brokers sono interconnessi in modo da condividere i metadata

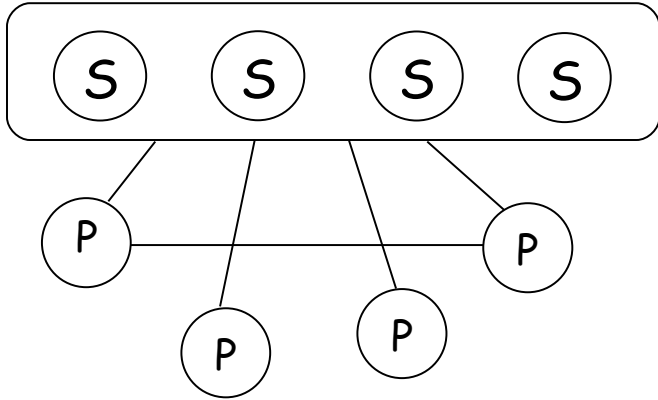


# Napster: Architettura

## Bilanciamento del carico

- L'intero sistema può essere acceduto mediante un unico punto di entrata, rappresentato mediante un unico nome simbolico (es: server.napster.com, porta 8875)
- Il DNS locale può effettuare il bilanciamento del carico, distribuendo le richieste ai diversi look-up servers
- Tecnica utilizzata per server che devono gestire un alto tasso di richieste (es: cnn.com, java.sun.com,...)
  - Si definisce un server farm
  - Il DNS associa al nome simbolico dell'host un insieme di indirizzi IP
  - Il DNS restituisce tutto l'insieme di indirizzi IP al richiedente, ma ruota l'ordine degli indirizzi. Poiché il client sceglie in genere il primo indirizzo IP, si ottiene un bilanciamento del carico sui diversi

# P2P ibrido vs. Server Farm

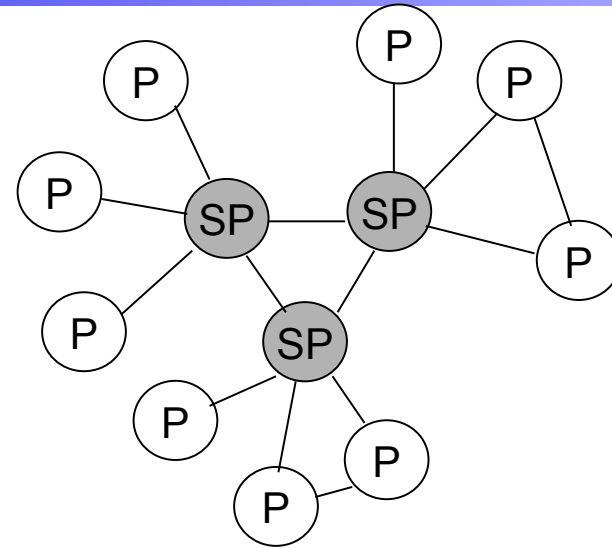


- i servers sono collocati in un'unica locazione geografica

⇒

maggiore individuabilità

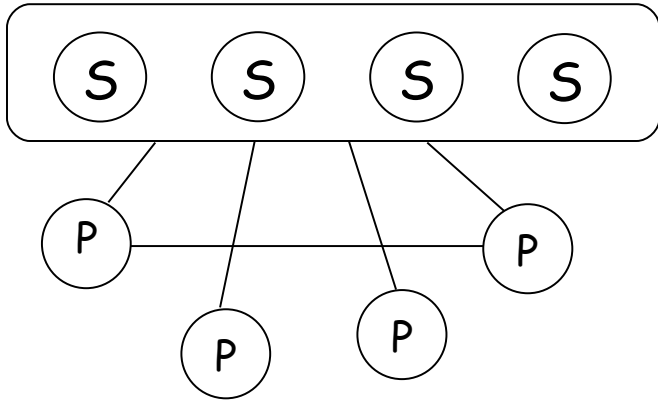
- un server ha solo funzionalità di coordinamento (full time activity)



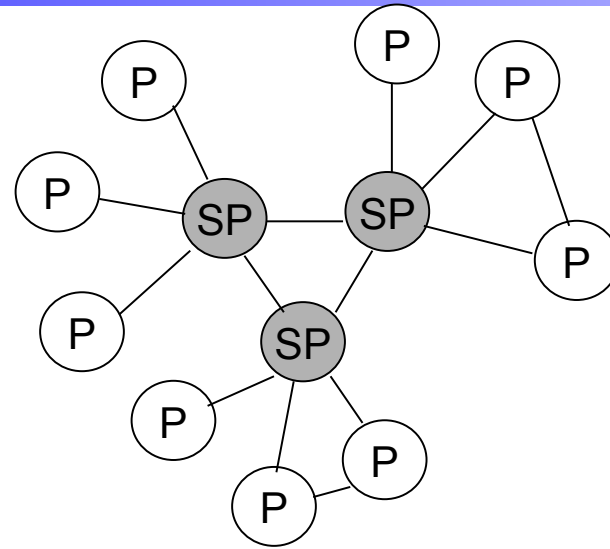
- L'insieme di servers è geograficamente distribuito

- Il super peer è un peer che si assume anche responsabilità di coordinamento (part time activity)

# P2P ibrido vs. Server Farm



- L' insieme dei servers è definito **staticamente**
- Unico punto di accesso al sistema  
⇒  
scarsa tolleranza ai guasti

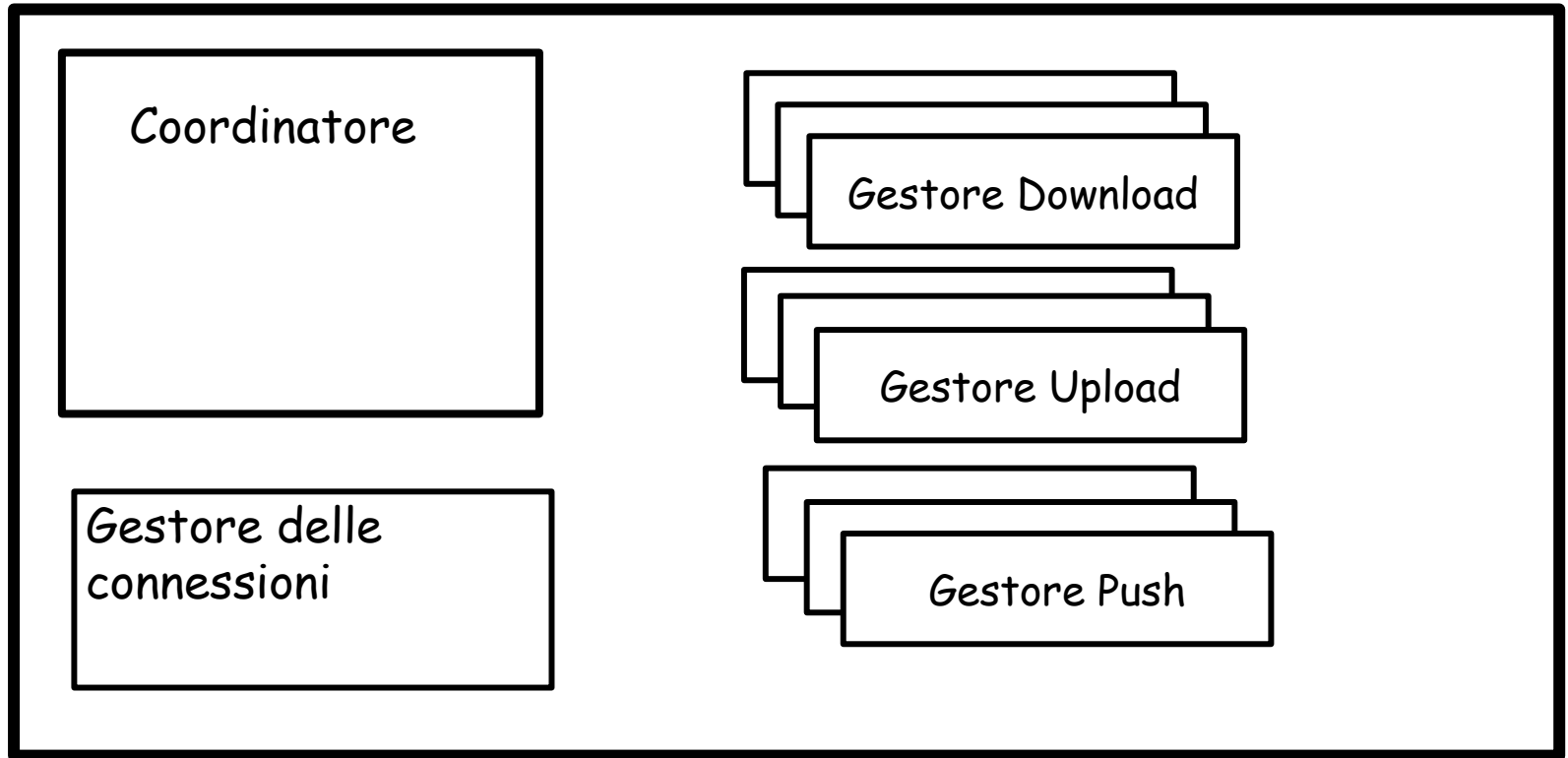


- Super peer sono **eletti dinamicamente**
- Nel caso di fault di un superpeer il sistema si riconfigura eleggendo altri superpeers

# Napster: il protocollo completo

- Un peer  $P$  riceve da un broker Napster una lista di peer che condividono una canzone ricercata.  $P$  sceglie un peer  $S$  dalla lista.
- La creazione della connessione tra  $P$  ed  $S$  viene coordinata dal broker  $B$ . Questo consente ai peer di accettare connessioni solo se "certificate" da  $B$ .
  - $P$  notifica a  $B$  che intende scaricare un file da  $S$
  - $B$  notifica ad  $S$  la richiesta di  $P$
  - $S$  notifica a  $B$  la propria disponibilità ad accettare la connessione
  - $B$  notifica a  $P$  la risposta di  $S$
  - A questo punto  $P$  può aprire una connessione con  $S$  per il trasferimento dei dati
- Supponiamo che  $S$  si trovi a monte di un firewall.
  - $S$  notifica a  $B$  la propria indisponibilità ad accettare connessioni
  - $S$  effettua un **push** del file verso  $P$  =  $S$  apre una connessione verso  $P$  e gli invia il file

# Napster: Struttura di un Peer



Struttura del Peer Napster

# Napster: Struttura di un Peer

- Il **coordinatore** gestisce:
  - le connessioni e le comunicazioni con il look-up server e quindi con il Broker
  - l'interazione con l'utente: ricerca di files, download, rimozione di files, etc.
- Il **gestore delle connessioni** si occupa di gestire tutte le richieste di connessione provenienti dagli altri peers (richieste di connessione per upload o per push)
- I gestori di **download, upload, push**, gestiscono lo scambio diretto dei dati tra i peers
  - Possono esistere più istanze di questi gestori. Ogni istanza si occupa di un singolo trasferimento
  - Push: utilizzato per peer che si trovano a monte del firewall.



# Napster: Struttura di un Peer

- Tutte le comunicazioni avvengono tramite TCP/IP
- Per unirsi alla rete Napster il peer

Aprire una connessione con un look-up server (porta 8875)

Il look-up server restituisce l'indirizzo di uno dei brokers (porta 6699)

Si connette al broker ed invia l'informazione relativa ai dati che vuole condividere (metadata)

Attende messaggi dal broker, oppure connessioni da altri peer (porta 6699) o eventi generati dall'utente locale

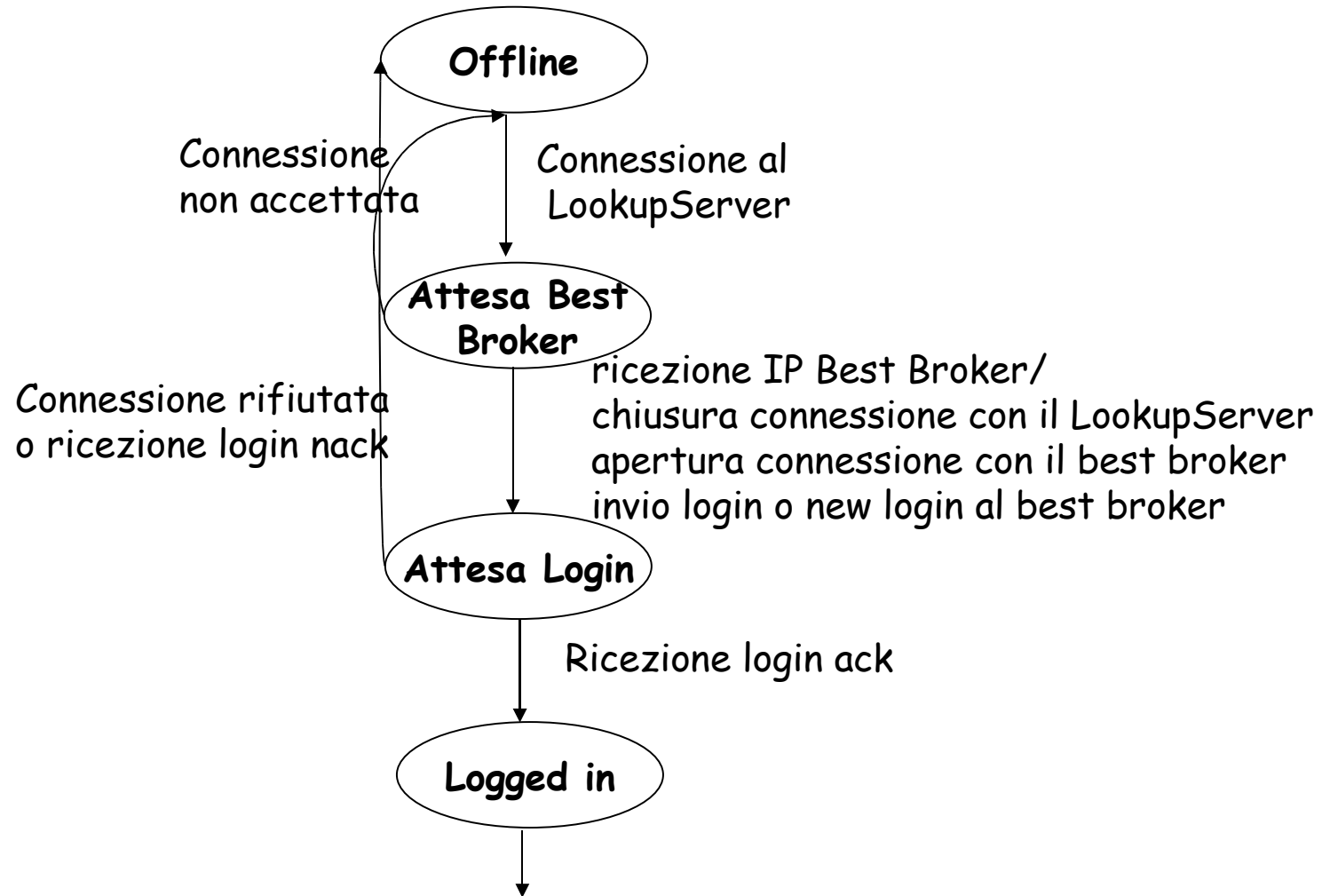
# Napster: Struttura di un Peer

- In Napster il download peer-to-peer di un file viene coordinato dal broker in modo centralizzato
- Download di un file: P1 vuole scaricare un file dal peer P2
  - P1 invia al proprio broker la richiesta di download (**download request**)
  - Il broker invia a P2 una richiesta di upload (**upload request**)
  - P2 invia al broker l'ack (**upload accept+porta su cui effettuare la connessione**)
  - Il broker invia a P1 l'ack (**download accept**)
  - P1 apre una connessione verso P2
  - Il file viene spedito da P2 sulla connessione aperta (**upload**)
- Il coordinamento centralizzato consente di introdurre un livello di sicurezza
  - ⇒
  - un peer non accetta una connessione se prima non ha ricevuto una richiesta di upload

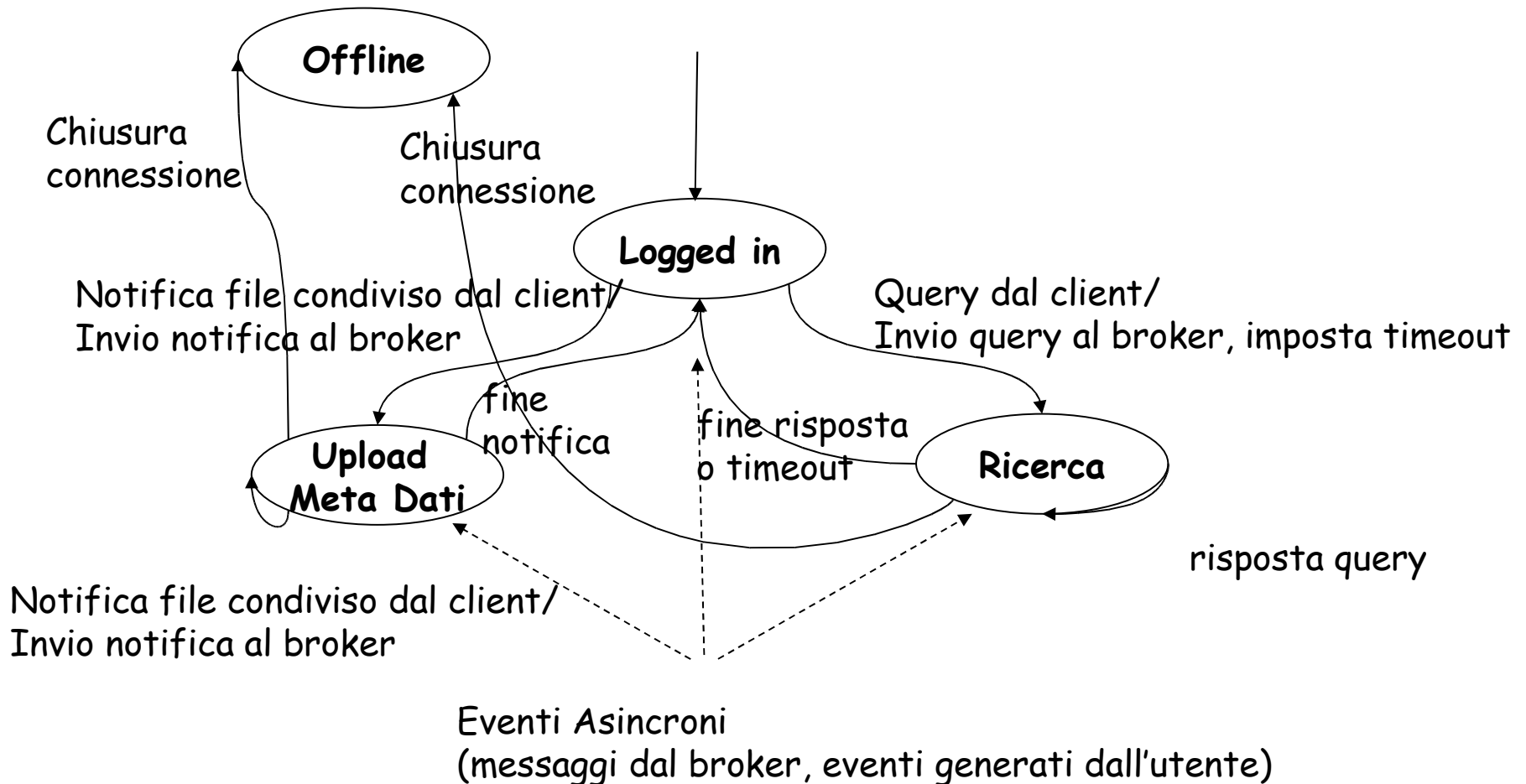
# Napster: Struttura di un peer

- Alcuni peer possono trovarsi a monte di un firewall
  - In questo caso, in generale, il peer non può accettare connessioni dall'esterno
  - Supponiamo che il peer possa aprire connessioni verso l'esterno
  - Almeno uno dei due peer deve accettare connessioni dall'esterno
- P1 vuole scaricare un file da un peer P2 che si trova a monte di un firewall
  - P1 invia al proprio broker la richiesta di download (**download request**)
  - Il broker invia a P2 una richiesta di upload (**upload request**)
  - P2 invia al broker l'ack indicando la disponibilità di effettuare l'upload, ma l'impossibilità di accettare connessioni dall'esterno (**upload accept**, porta = 0)
  - Il broker invia a P1 l'ack (**download accept**, porta = 0)
  - P1 dichiara di accettare la connessione da P2 (**alternate download request**)
  - Il broker invia a P2 l'ack (**alternate download ack**)
  - P2 apre una connessione verso P1
  - Il file viene spedito da P2 sulla connessione aperta (**push**)

# Napster: il Coordinatore



# Napster: il Coordinatore



# Napster: il Coordinatore

## Eventi Asincroni

- Ricezione di **statistiche** da parte del server
- Richiesta di **cancellazione di un file** (dall'utente) / invio meta dati al broker
- Richiesta di **download** da parte dell'utente / invio **download request** al server
- Richiesta di **upload** da parte del broker / invio (**upload accept + porta**, porta  $\neq 0$  se posso accettare connessione, porta = 0 se non posso accettare connessioni, perché a monte di un firewall) oppure **accept failed**
- Ricezione di **download ack** con **porta > 0** / apertura connessione con il peer remoto, attivazione di un'istanza del Gestore di Download
- Ricezione di **download ack** con **porta = 0** / invio di **alternate-download request** al server o rifiuto ad accettare connessione (se il peer si trova a monte di un firewall)
- Ricezione di **alternate download ack** / attivazione di un'istanza del Gestore di Push

# Napster: Gestore delle Connessioni



**Richiesta Push**/attivazione di  
una istanza del *Gestore*  
di Download

**Richiesta Upload**/  
attivazione di una istanza del  
*Gestore* di Upload

# Napster: struttura generale del Peer

- *Gestore del Download* = riceve il file dal peer remoto e lo memorizza sul disco (esegue una sequenza di receive)
- *Gestore Upload* = spedisce il file al peer remoto che lo ha richiesto (esegue una sequenza di send)
- *Gestore Push* = apre una connessione verso il peer remoto e spedisce il file richiesto (come sopra, ma apre anche la connessione)



# Napster: Vantaggi e Svantaggi

- **Svantaggi: Server centralizzato**

  - Collo di bottiglia

  - Limitata scalabilità

  - Poco affidabile (single point of failure)

- **Vantaggi**

  - Ricerca veloce (one hop lookup)

  - Ricerca completa =assicura che, se una informazione esiste nel sistema, essa viene individuata

  - Central managing/trust authority

  - Minimizzazione dei messaggi spediti sulla rete. Non richiesti messaggi per la riconfigurazione dinamica della rete (vedi keep-alive in Gnutella)

- **Aree di applicazione**

  - VoIP

  - Auctioning (Ebay)