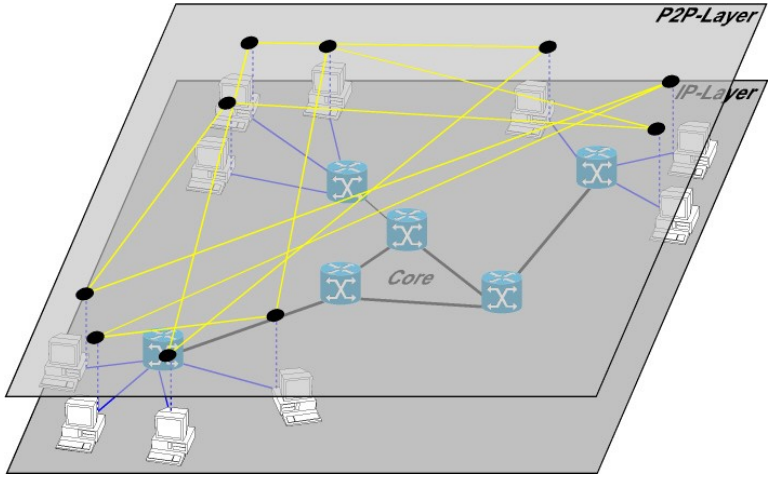
Lezione n.4
Sistemi P2P di Prima Generazione: Sistemi Centralizzati

Laura Ricci

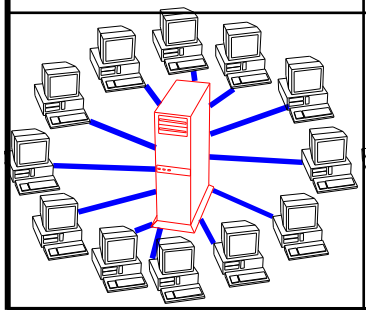
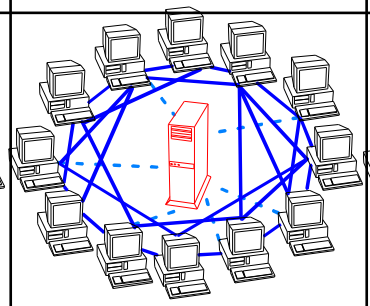
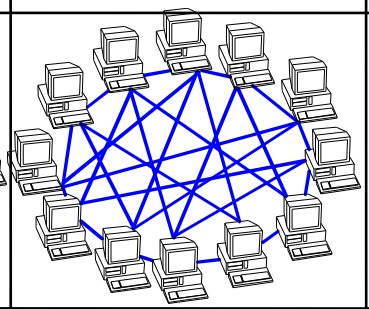
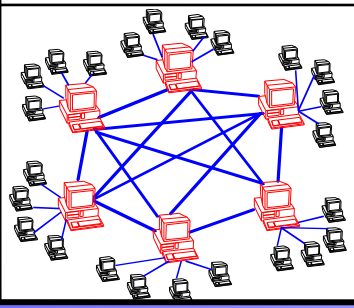
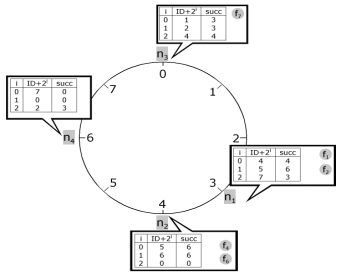
Peer to Peer systems and Applications
Capitolo 5

INTRODUZIONE

1. Arhitetture di 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

Client-Server	***Peer-to-Peer: Caratteristiche Generali***			
1. Il Server è l'unica entità in grado di memorizzare l'informazione condivisa → La rete è gestita dal Server in modo centralizzato 2. Il Server è il Sistema con la maggior capacità di calcolo 3. I client posseggono minor capacità di calcolo **Esempio:** WWW	1. Le risorse sono condivise tra i peers. 2. Le risorse offerte da un peer possono essere accedute direttamente da altri peer 3. Il peer fornisce e richiede risorse (Servent)			
	P2P Non strutturati			***P2P Strutturati***
	P2P Centralizzato	*P2P Puro*	*P2P Ibridp*	*P2P basato su DHT*
	• Soddisfano 1.,2.,3. • Esiste una entità centralizzata • Tale entità centralizzata svolge solo compiti di indicizzazione delle risorse **Esempio:** Napster	1. Soddisfano 1.,2.,3. 2. Non esiste alcuna entità centralizzata 3. La funzionalità del sistema non viene compromessa dall' eliminazione di un peer **Esempi:** Gnutella 0.4,	1. Soddisfano 1,.2.,3. 2. Esistono alcuni peer (superpeer) chehanno anche funzioni di indicizzazione 3. I superpeer sono determinati dinamicamente **Esempi:** Gnutella 0.6, JXTA	1. Soddisfano 1.,2.,3. 2. Non esistono entità centralizzate 3. La overlay network risulta strutturata 4. Overlay network strutturata **Esemnpi:** Chord, CAN,Pastry
		1st Gen.	**2nd Gen.**	
				

P2P: CARATTERISTICHE GENERALI

- 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:CARATTERISTICHE GENERALI

- Overlay Topology: rete virtuale stabilita tra i peer mediante connessioni TCP per lo scambio di informazione di controllo

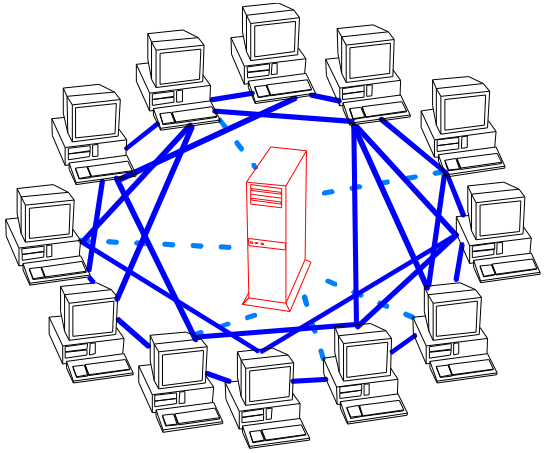
- Caratteristiche della overlay network
  - completamente indipendente dalla rete fisica, grazie al livello di astrazione TCP/IP
  - può essere strutturata in modo gerarchico (Gnutella Superpeers, JXTA rendez-vouz peers)
  - può includere un server centralizzato (look up server di Napster)

- **I peer partecipano attivamente alla overlay network**
  - Forniscono informazione
  - Ricercano informazione
  - Svolgono funzioni di routing

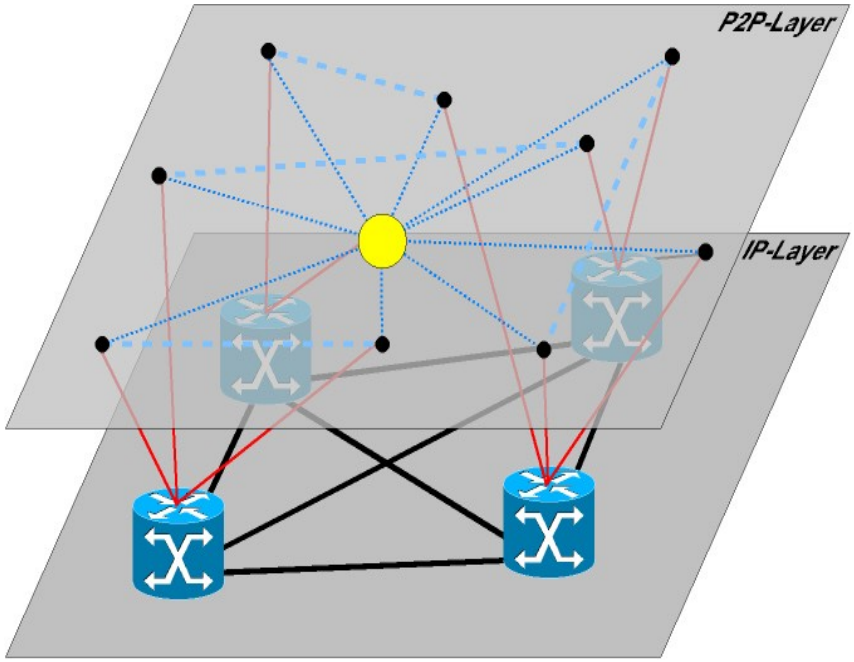
SISTEMI P2P CENTRALIZZATI: NAPSTER

- Tutti i peer si connettono ad un server centralizzato (broker). La topologia dell'overlay network risultante è a stella: star overlay network.

- Le connessioni tra i peers sono stabilite dinamicamente 'on demand' per lo scambio dei i dati



Topologia di Napster



Servent
Connection between 2 servents (TCP)
Connection between router & servent
Connection between routers (Core)

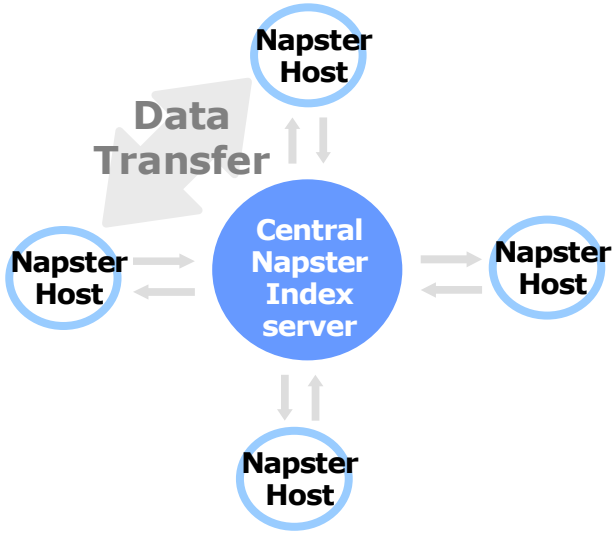
NAPSTER:PRINCIPI DI FUNZIONAMENTO

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 qulche criterio (bitrate, frequency, tipo di collegamento)
- Si connette al peer prescelto e scarica il file(connessione HTTP)

Ricerca : client server

Download: P2P

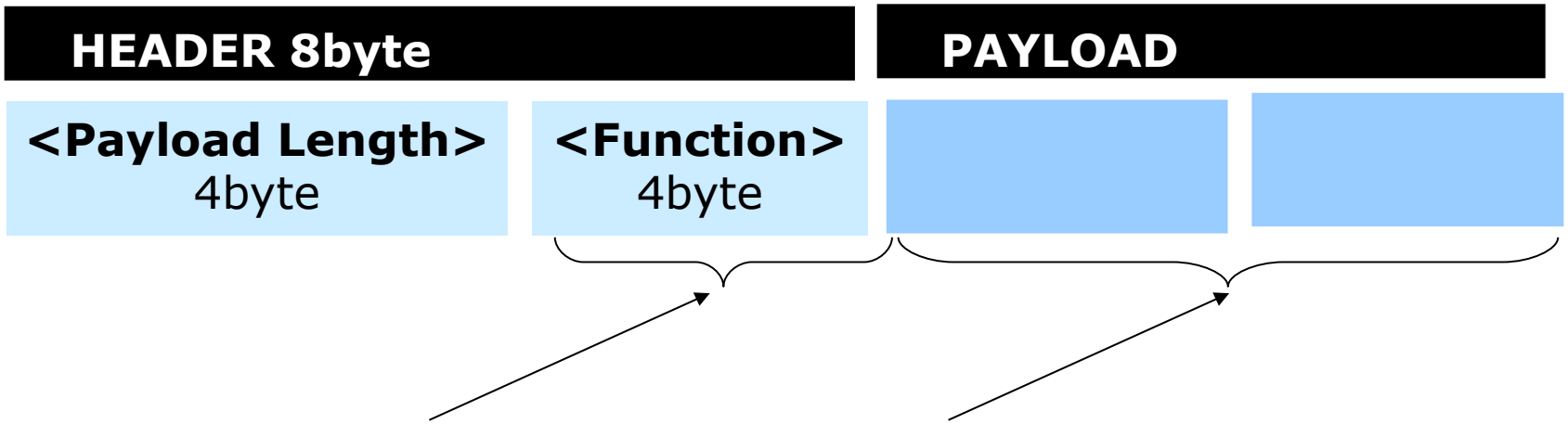


NAPSTER: PRINCIPI DI FUNZIONAMENTO

- 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'e' 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



Contiene il tipo del messaggio (e.g. login, search,…)

Contiene i parametri del messaggio(dipendente dal tipo di messaggio,e.g. keywords nel caso di search,…)

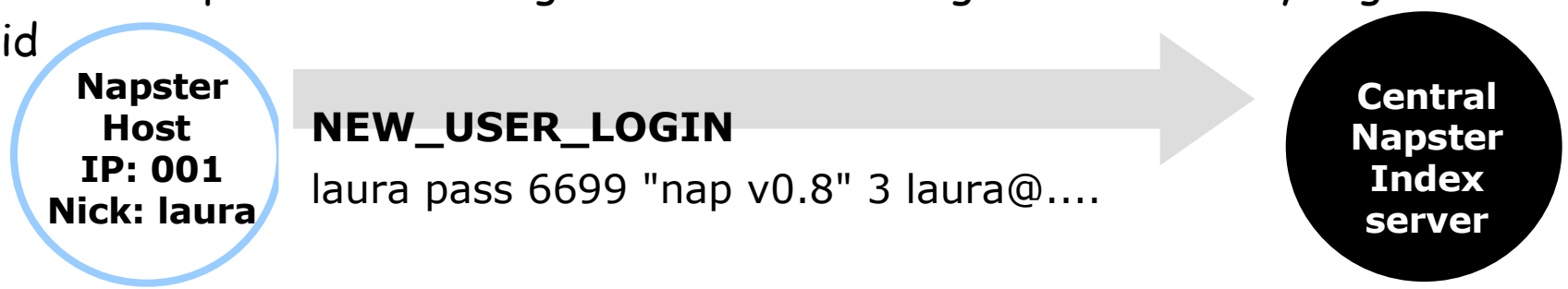
NAPSTER:REGISTRAZIONE

Client/Server Service

1: NEW USER LOGIN

<Nick>	<Password>	<Port>	<Speed>	<e-mail>

- Utilizzato quando l'utente si collega per la prima volta
- L'utente puo' utilizzare un messaggio di nick check per verificare l'unicità del nickname
- Il server risponde con un msg di nickname not registered/already registered/ invalid



NAPSTER:INIZIALIZZAZIONE

Client/Server Service

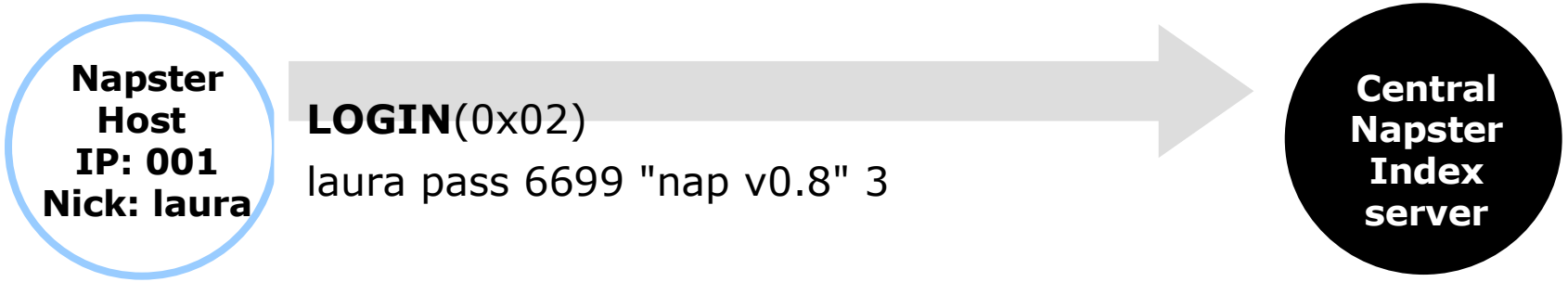
1: LOGIN (Function:0x02)

<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 LOGIN

Client/Server Service

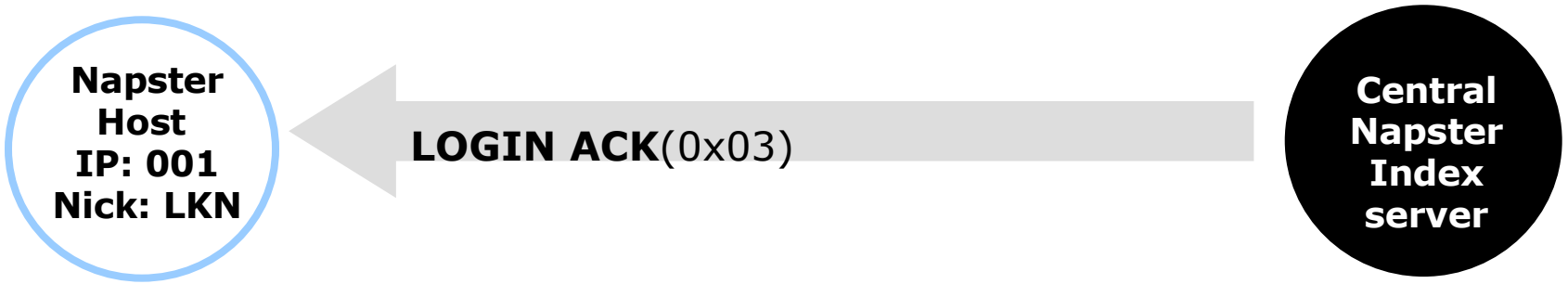
1: LOGIN (Function:0x02)

<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 LOGIN

Client/Server Service

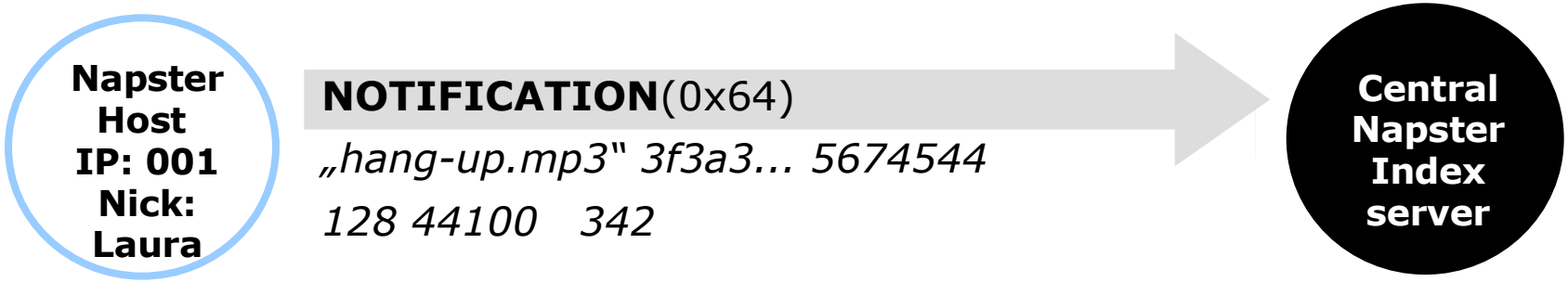
1: LOGIN (Function:0x02)

<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 LOGIN

- MD5 (acronimo di Message Digest algorithm 5) è un algoritmo per la crittografia dei dati
- MD5 prende in input una stringa di lunghezza arbitraria e produce in output una stringa a 128 bit (32 valori esadecimali, indipendentemente dalla stringa di input).
- l'output (noto anche come "MD5 Checksum" o "MD5 Hash") restituito è con alta probabilità univoco
- È praticamente impossibile ottenere date due diverse stringhe in input la medesima stringa in output
- È estremamente complesso risalire alla stringa di input partendo dalla stringa di output (la gamma di possibili valori in output è pari a 16 alla 32esima potenza).

NAPSTER LOGIN

- MD5 (fingerprint) – Introdotto originariamente per
  - identificare univocamente un file MP3 e facilitarne il reperimento
  - tenere traccia dei files duplicati nel sistema
  - facilitare il download del file da utenti diversi
- Ma....gli oppositori di NAPSTER sostennero che le canzoni scaricate illegalmente sul sistema (ad esempio i CD appena lanciati dalle case discografiche) potevano essere individuati facilmente tramite il loro MD5
- Risultato: questo campo è stato tolto in seguito alla causa legale sostenuta dalle case discografiche contro NAPSTER
- Altri campi:

  Size – Dimensione in byte dl file

  BitRate, Frequency – indicano la qualità dell'MP3

  Time - Durata della canzone

NAPSTER LOGIN

Bitrate

- File mp3 è un flusso di bit suddiviso in frames (simili ai fotogrammi di un film).
- Il bitrate è il valore che indica quanti bit vengono usati per codificare un secondo di musica .
- Si esprime in kilobit per secondo (kbps) e
- Maggiore sarà la quantità di bit utilizzati migliore sarà la resa
- Normalmente, la qualità in ascolto è proporzionale al bitrate, dunque bitrate sempre più alti garantiscono sicuramente qualità superiore.

Frequenza

- Frequenza di campionamento del suono

NAPSTER: RICERCA

1: SEARCH (Function: 0xC8)

[FILENAME CONTAINS „Search Criteria“] [MAX_RESULT <Max>]

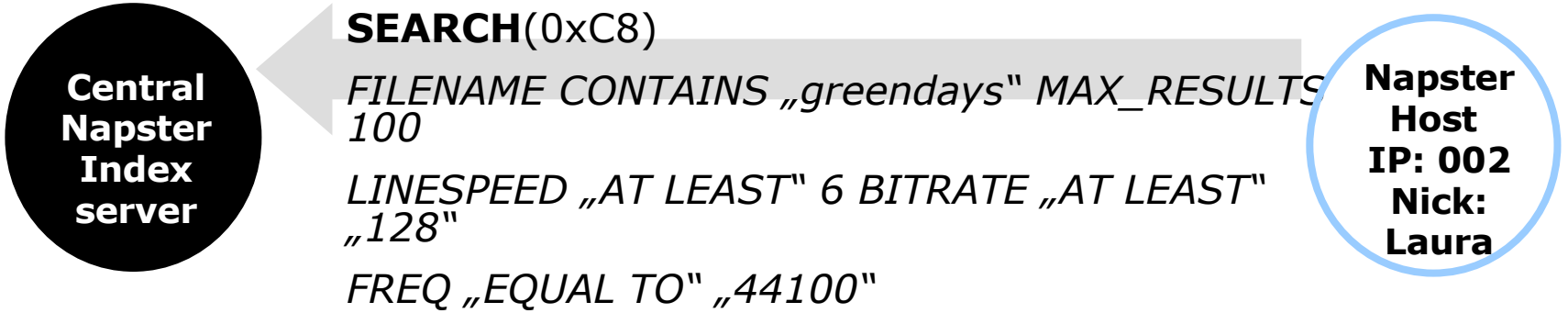
[LINESPEED <Compare> <Link-Type>]

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

2: SEARCH RESPONSE (Function: 0xC9)

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

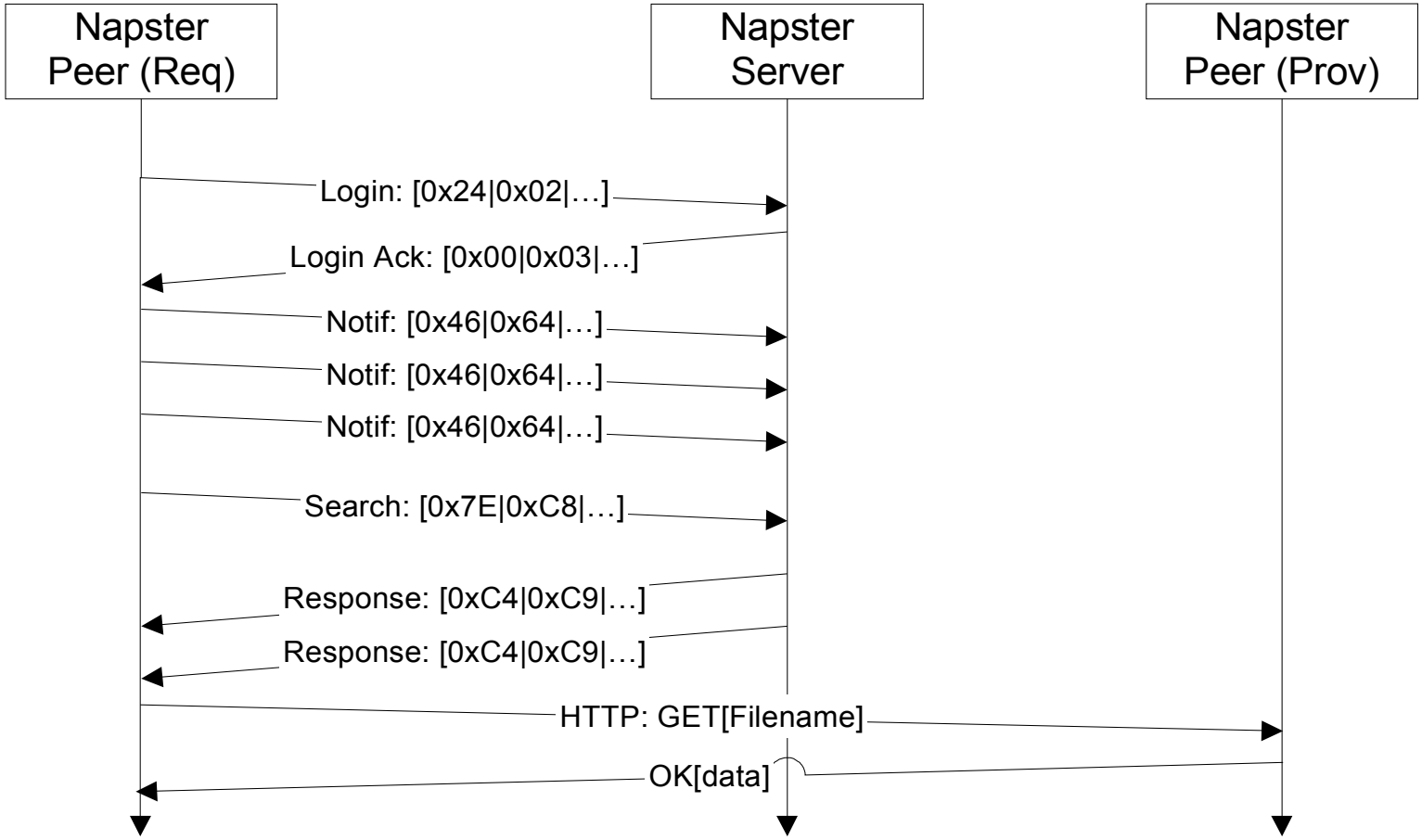
<Time> <Nick> <IP> <Link-Type>



NAPSTER: RICERCA

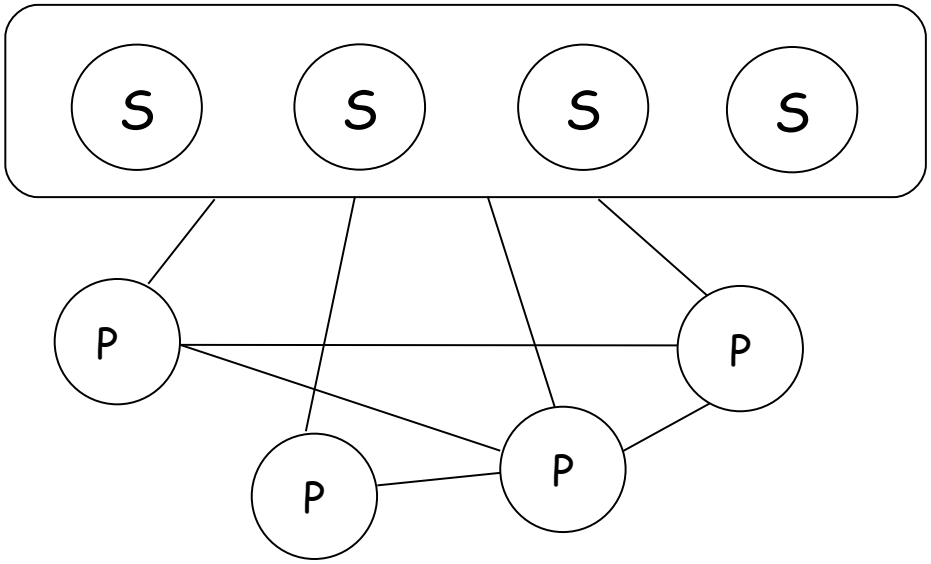
- Quando il server riceve un messaggio di SEARCH, ricerca nel proprio database un file che soddisfi i parametri della ricerca.
- Il database è creato con i dati ricevuti dai peer
- Se la query è soddisfatta, ilserver risponde con almeno un messaggio di serach response
- Il messaggio di query response contiene:
  - Il nome completo del file
  - Le sue caratteristiche. Viene spedito anche l'MD5 del file. Questo verrà utilizzato per controllare l'integrità del file scaricato
  - L'indirizzo IP dell'host che lo possiede
- Il file viene scaricato utilizzando il protocollo HTTP

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: IL PROBLEMA DELLA SCALABILITA'



Definizione di un cluster di server (server farm) per aumentare la scalabilità del sistema

- I server sono localizzati in un'unica locazione geografica

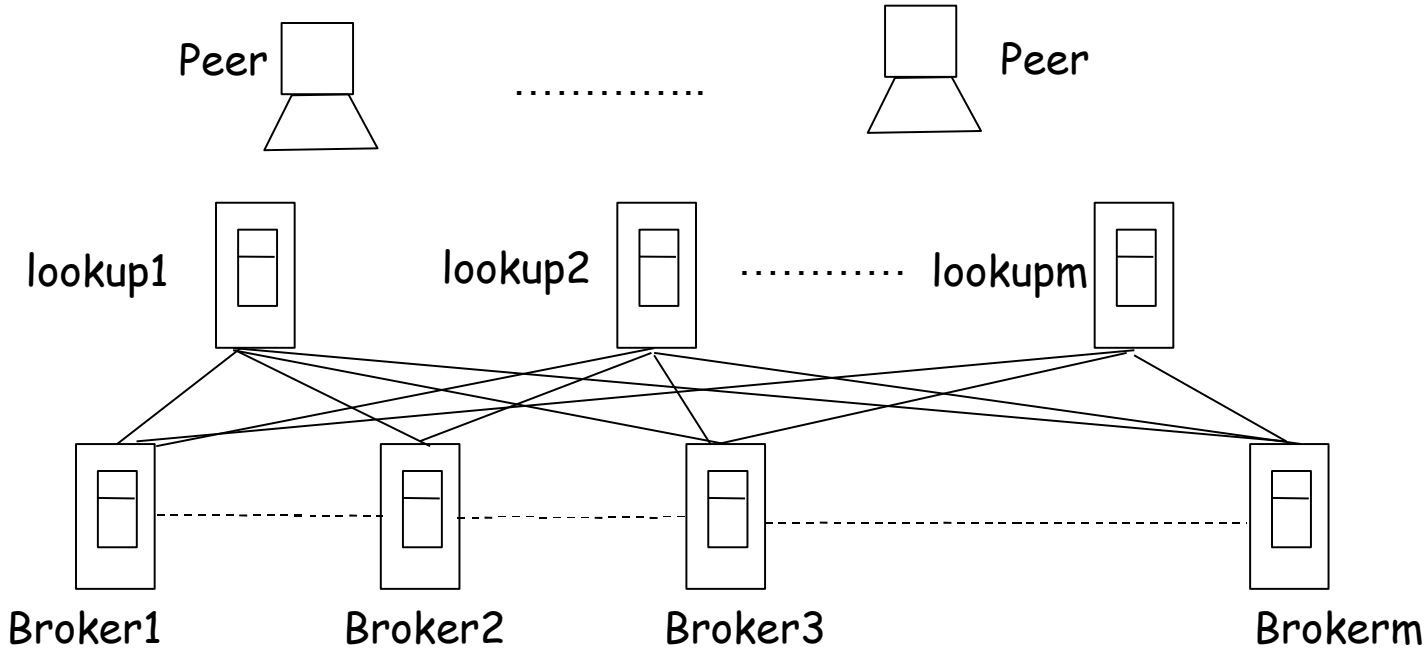
Esiste comunque un unico punto di accesso al sistema (esempio i router che collega il cluster alla rete)

- i server condividono il database contenente le meta-informazioni

NAPSTER: IL PROBLEMA DELLA SCALABILITA'

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

- Brookers: 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 metadati



NAPSTER: ARCHITETTURA

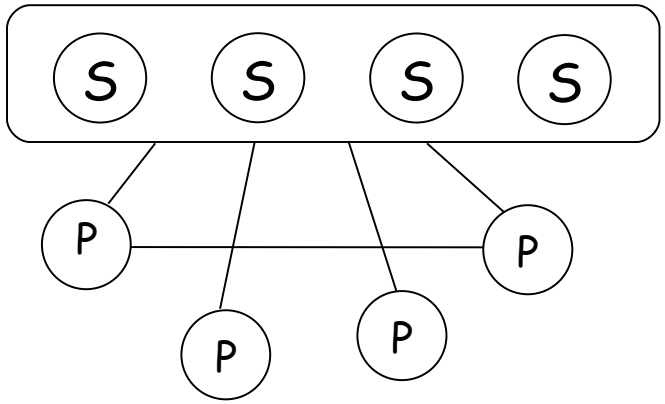
Bilanciamento del carico

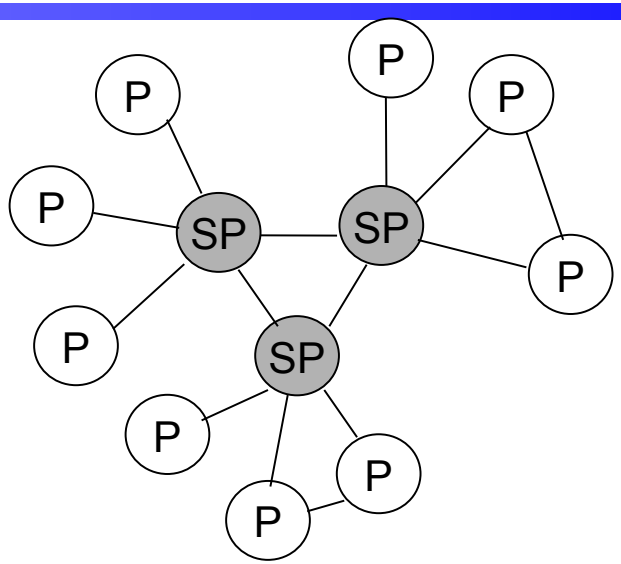
- L'intero sistema può essere acceduto mediante un unico punto di entrata, rappresentato mediante un unico nome simbolico (es: server.napster.com, porta 8875)

- 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 indirirzzi. 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

⇒

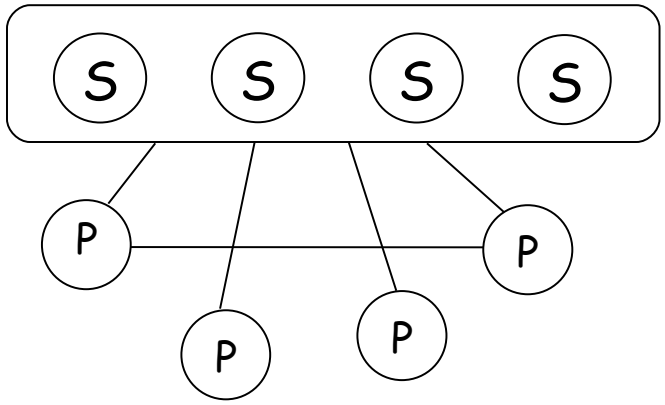
maggiore individuabilità

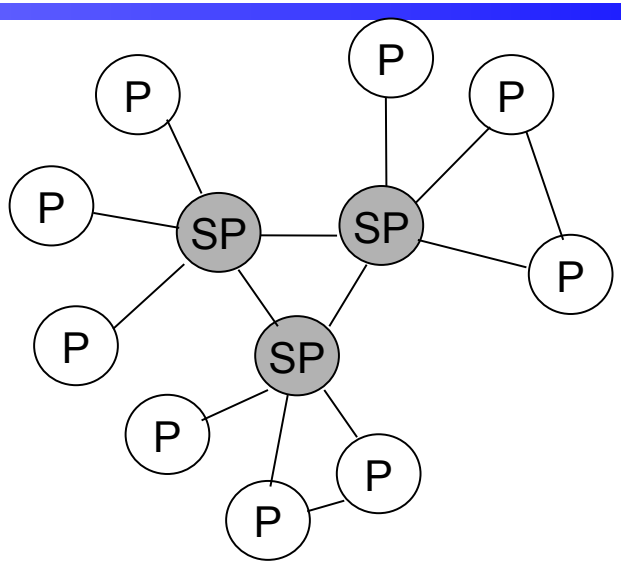
- un server ha solo funzionalità di coordinamento (full time activity)

- L'insieme di servers è geograficamente distribuito

- Il super peer è un peer che si assume anche funzionalità 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

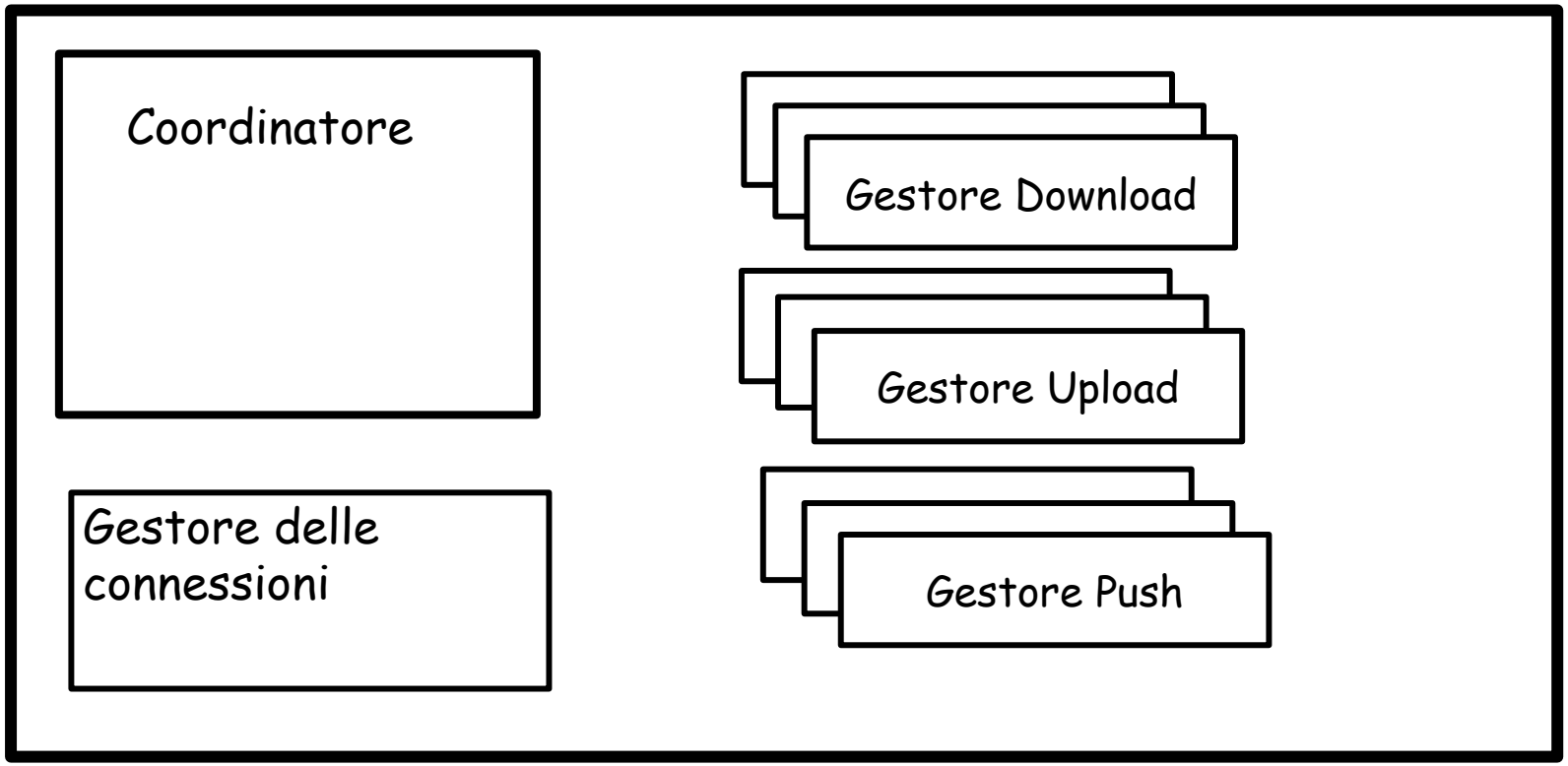
NAPSTER: IL PROTOCOLLO COMPLETO

- Un peer P riceva 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
  - Apre 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

  $\Rightarrow$

  un peer non accetta una connessione se prima non ha ricevuto una richiesta di upload

NAPSTER: STRUTTURA DI UN PEER

- Se il peer P1 che possiede il file F non si trova a monte di un firewall, il peer P2 che vuole scaricare F apre una connessione sulla porta specificata dal server
- P2 accetta la connessione
- P1 invia una GET, poi invia un messaggio

  <mynick> "<filename>" <offset>

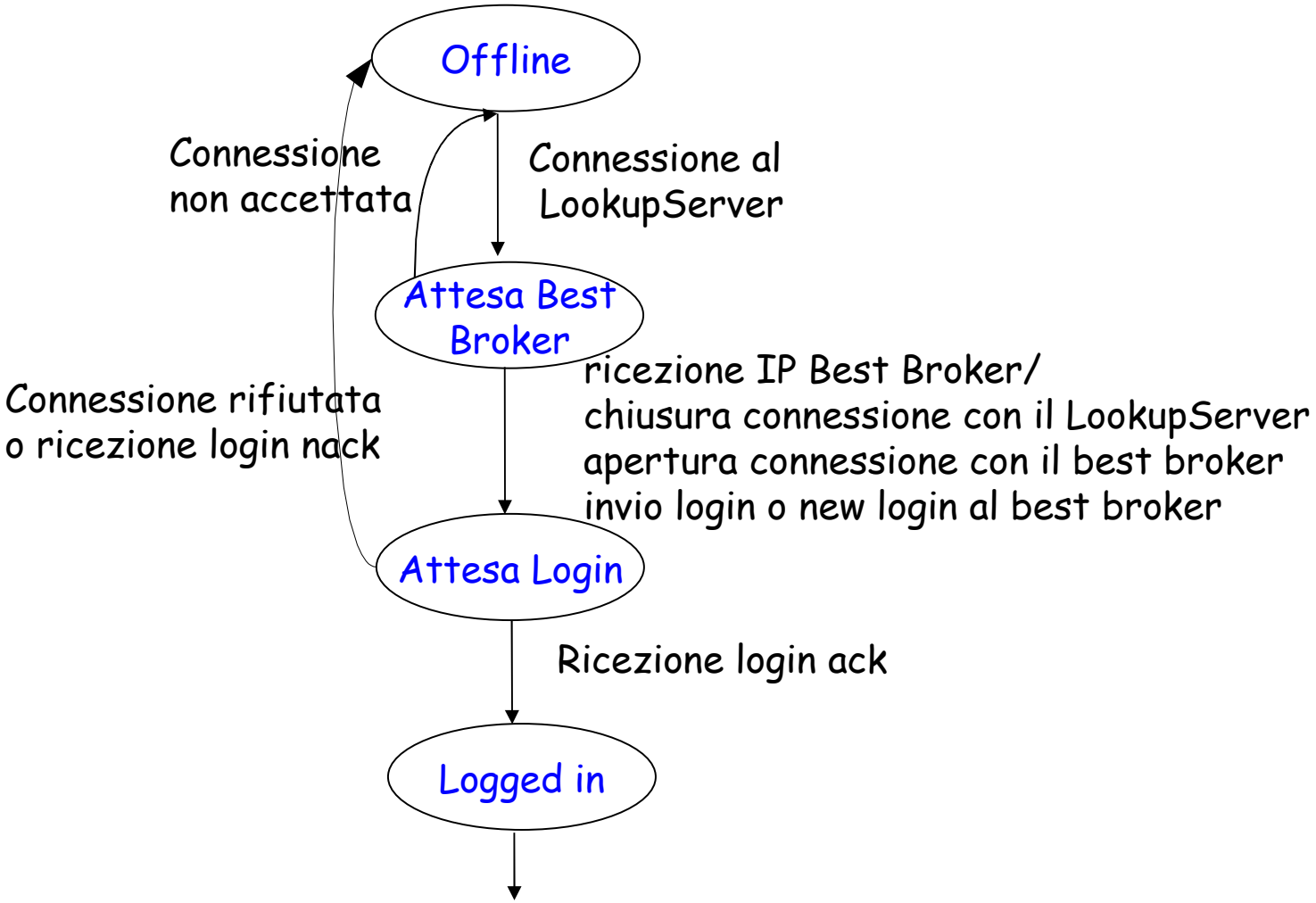
  <offset> è l'offset in F che indica il primo byte che si intende scaricare (0 la prima volta)
- P2 restituisce la lunghezza del file, oppure un messaggio di errore
- Se non ci sono errori P2 inizia ad inviare il file

NAPSTER: STRUTTURA DI UN PEER

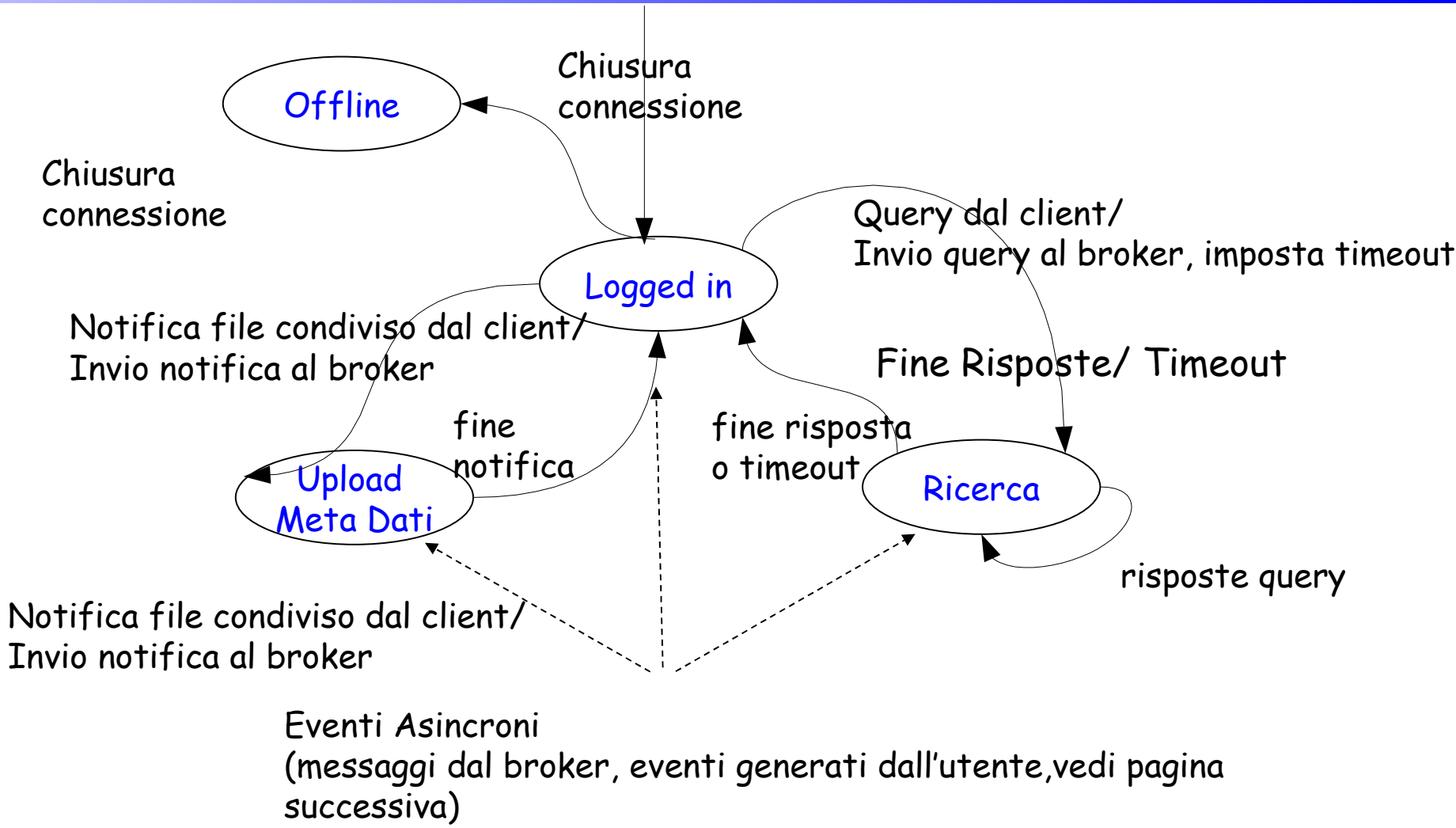
- Alcuni peer possono trovarsi a monte di un firewall
  - In questo caso, in generale, il peer non può accettare connessioni dall'esterno
  - Supponiamo che il peer possa aprire connessioni verso l'esterno
  - Almeno uno dei due peer deve accettare connessioni dall'esterno

- 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)

STRUTTURA DI UN PEER:IL COORDINATORE



STRUTTURA DI UN PEER: IL COORDINATORE



STRUTTURA DI UN PEER: 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 ≠ 0 se posso accettare connessione, porta = 0 se non posso accettare connessioni, perché a monte di un firewall) oppure accept failed
- Ricezione di dowload 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: GESTIONE DELLE CONNESSIONI

Gestore delle connessioni: schema generale

Richiesta SEND/
attivazione di
una istanza del Gestore
di Download

Richiesta GET/
attivazione di una istanza del
Gestore di Upload

Una connessione può essere aperta da
un peer che richiede il download (GET): invio i dati, attivo una istanza del gestore di upload
un peer che esegue la push (SEND): ricevo i dati

NAPSTER: STRUTTURA GENERALE DEL PEER

- Gestore 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: VALUTAZIONE DEL PROTOCOLLO

- Consideriamo un utente che condivide 10 files e richiede un file che è condiviso da altri 20 utenti
- Traffico sulla rete generato dal protocollo (solo messaggi per l'implementazione del protocollo, non download

(login + login-ack) + 10* notif + 1 * serch + 10 * response

login = 40 bytes, login_ack= 4 bytes, notif = 74 bytes,

search = 130 bytes, response = 200 bytes

40 + 4 + 10* 74 + 130 + 10* 200 = 2914 bytes

- 2914 bytes per la gestione del protocollo

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
  - L'entità centralizzata garantisce funzionalitàm di controllo/sicurezza
  - Minimizzazione dei messaggi spediti sulla rete. Non richiesti messaggi per la riconfigurazione dinamica della rete (vedi keep-alive in Gnutella)
- Idee di base ripresa da
  - BitTorrent
  - E-mule- E-donkey
  - .....