



Linguaggi ed Applicazioni multimediali

05.01- Architectures
05.02- Motivation
05.03 – Main P2P system

Sistemi peer to peer



P2P - Caratteristiche

Perché un sistema si possa definire peer to peer deve presentare almeno 4 caratteristiche:

- 1. PEER DISCOVERY:** i peer devono poter scoprire altri peer sulla rete
- 2. RESOURCE SHARING:** i peer devono poter condividere le risorse con gli altri peer
- 3. RESOURCE DISCOVERY:** i peer devono poter effettuare ricerche ed interrogazioni per scoprire le risorse messe a disposizione da altri peer
- 4. DINAMICITÀ:** il sistema deve funzionare anche se i peer si disconnettono e riconnettono più o meno frequentemente

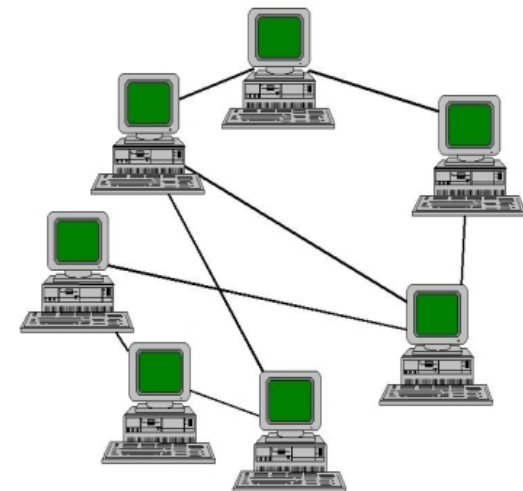


Sistemi peer to peer (P2P)

In una **rete peer to peer** (P2P) i computer condividono risorse e servizi e gli scambiano direttamente tra di loro.

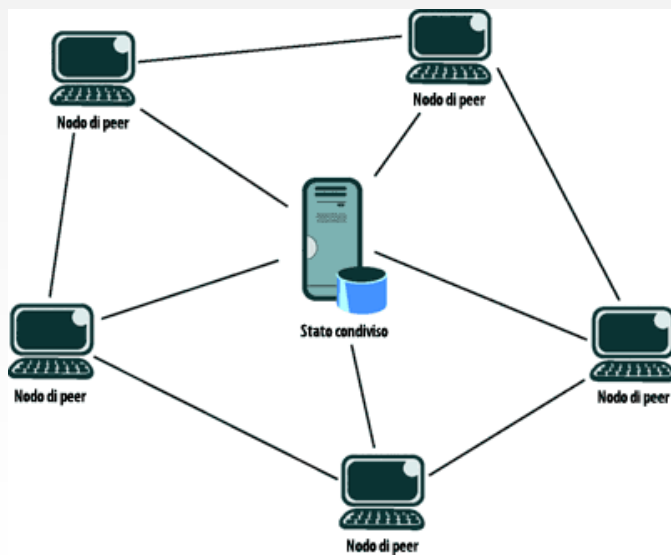
Si tratta di una rete paritaria, dato che non possiede nodi gerarchizzati come client o server, ma un numero di nodi equivalenti (in inglese peer) che fungono **sia da client che da server** verso altri nodi della rete.

Mediante questa configurazione qualsiasi nodo è in grado di avviare o completare una transazione. I nodi equivalenti possono differire nella configurazione locale, nella velocità di elaborazione, nella ampiezza di banda e nella quantità di dati memorizzati. L'esempio classico di P2P è la rete per la condivisione di file (File sharing).



Sistemi peer to peer (P2P)

Un'applicazione **client-server** è invece un'applicazione nel quale un computer client istanzia la richiesta di un servizio connettendosi ad una server application o ad un sistema di database.



Più semplicemente, i sistemi client/server sono un'evoluzione dei sistemi basati sulla condivisione semplice delle risorse.

La presenza di un server permette ad un certo numero di client di dividerne le risorse, lasciando che sia il server a gestire gli accessi alle risorse per evitare conflitti tipici dei primi sistemi informatici.

Un sistema client/server non è altro che la trasposizione informatica dei sistemi convenzionali.



P2P – Fasi principali

Le applicazioni P2P sono costituite da tre fasi principali:

- **Boot:** permette a un peer di trovare la rete e di connettersi ad essa; (nessuno o quasi fa boot P2P)
- **Lookup:** permette ad un peer di trovare il gestore responsabile di una determinata informazione; (pochi sono P2P, alcuni usano SuperPeer)
- **Scambio di file;** (sono tutti P2P)



P2P – Tipologia di applicazioni

Parleremo di applicazioni:

- P2P puro se:
 - boot, lookup e scambio di file sono P2P;
- P2P se:
 - le fasi di lookup e scambio di file sono P2P;
 - la fase di boot utilizza qualche SERVER;



P2P – Tipologia di applicazioni

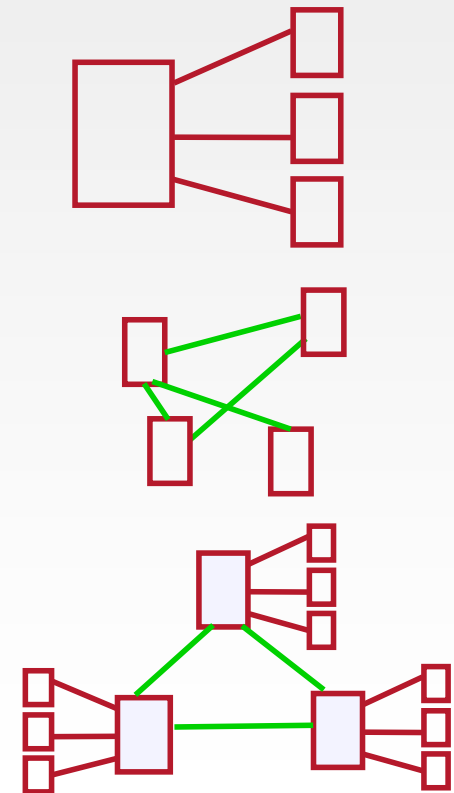
- P2P Ibride se:
 - la fase di scambio dei file è P2P;
 - la fase di boot utilizza qualche SERVER;
 - nella fase di lookup vengono usati Peer particolari:
 - SuperPeer , Ultra Peer(Gnutella2)
 - Supernodo (KaZaA) NodoRendezVous (JXTA)
 - MainPeer (EDonkey) Server (WinMX)



Reti P2P - Classificazioni

Considerando l'operazione lookup:

- Lookup Centralizzato: indice centralizzato;
- Lookup Decentralizzato: indice completamente distribuito;
- Lookup Ibrido: più sistemi centralizzati collegati in un sistema decentralizzato;





Modelli P2P

Flooded Requests Model (FRM)

Quando un peer genera una query, la propaga a tutti i suoi vicini, indistintamente, i quali fanno lo stesso. Il numero massimo di salti che una query può fare è un parametro fissato (**TTL**, *time to live* oppure *horizon*).

$$TTL(0) = TTL(i) + Hops(i)$$

(Hops: quanti server hanno già inoltrato il messaggio)

Vantaggi: efficiente in comunità di dimensioni limitate

Limiti: richiede molte risorse in termini di banda



Modelli P2P

Centralized Directory Model (CDM)

- I peer si connettono a un direttorio centrale (virtuale) per pubblicare informazioni sui contenuti che offrono in condivisione.
- Quando un peer invia una query al direttorio, quest'ultimo risponde fornendo un elenco dei peer che condividono l'oggetto richiesto.
- Lo scambio di dati avviene poi da peer a peer.

Vantaggi: semplice, dà controllo sui contenuti condivisi

Limiti: poco scalabile, e il direttorio richiede una implementazione robusta

Es.: Napster, BitTorrent



Modelli P2P

Document Routing Model (DRM)

La rete P2P di nuova generazione realizza una **Distributed Hash Table (DHT)**, cioè un sistema distribuito che suddivide equamente tra i nodi la responsabilità delle chiavi (ciascuna delle quali è associata a un valore, che nella pratica corrisponde a un file, a un blocco di dati, ecc.).

Un messaggio di ricerca contenente una chiave viene propagato all'unico responsabile di quella chiave, che fornisce in risposta il valore cercato.

La chiave viene solitamente generata applicando una funzione (detta funzione di hash) al dato stesso che deve essere inserito/cercato nella DHT.

Sono spesso utilizzate le funzioni della famiglia SHA (*Secure Hash Algorithm*)



Modelli P2P - DHT

- ▶ A ogni file e ad ogni nodo è associata una chiave: la chiave viene creata facendo l'hash del nome del file o dell'IP del nodo;
- ▶ Ogni nodo del sistema è responsabile di un insieme di file/chiavi e tutti realizzano una DHT;
- ▶ L'unica operazione che un sistema DHT deve fornire è lookup(key), la quale restituisce l'identità del responsabile di una determinata chiave.





NAPSTER

Architettura: centralizzata, non strutturata

Modello: CDM

- Vi sono tre entità fondamentali in Napster.
 - Client: utente che ricerca e richiede i file.
 - Server: sistema host finale che condivide e invia i file.
 - Metaserver/Redirector: repository dell'indice dei file.



P2P - Gnutella

Architettura: pura, decentralizzata

*Modello: **FRM** (Flooded Requests Model)*

L'instradamento dei messaggi viene eseguito tramite *flooding*.

- I descrittori ping/query sono inviati a tutti i client direttamente connessi.
- I descrittori pong/queryhit sono rimandati indietro lungo lo stesso percorso.
- Si usa un meccanismo TTL per limitare la distanza (horizon).



Gnutella – Il protocollo

BOOTSTRAP

Inserimento del peer nella rete mediante individuazione di peers attivi ed apertura di connessioni.

I meccanismi di bootstrap possono essere basati su:

- **beacon (bootstrap) servers**: memorizzano una lista di peers spesso attivi sulla rete
- **peer cache**: memorizzano liste di hosts contattati in sessioni precedenti



Gnutella – Il protocollo

ESPLORAZIONE

La rete viene esplorata mediante l'invio di messaggi di ping/pong (ping message, pong message). Questo consente di scoprire ulteriori peer sulla rete e quindi creare eventuali connessioni con altri peer.

- **Ping**: utilizzato per scoprire altri host sulla rete. Un Servent che riceve un Ping deve rispondere con un Pong se è disponibile ad accettare connessioni.
- **Pong**: utilizzato come risposta al Ping; contiene l'indirizzo del Servent (indirizzo IP e porta) ed informazioni circa i dati condivisi (numero di file, dimensione totale).



Gnutella – Il protocollo

SOTTOMISSIONE DI QUERIES

- Invio delle queries ai peer vicini (query message). I vicini a loro volta inoltrano la query mediante un meccanismo di **enhanced flooding**, limitato da **TTL** (Time To Life).
- Ricezione di risposte dai vicini della risposta “migliore” (query hit message)



Gnutella – Il protocollo

- **QueryHit:** la risposta ad una query. Un Servent che riceve una Query deve rispondere con un QueryHit se ha trovato corrispondenza con i dati richiesti.

Il messaggio contiene le informazioni per acquisire i dati richiesti tramite HTTP.

- **Push:** se il proprietario del file cercato (es. l'host B) è dietro un firewall che blocca la porta HTTP, il richiedente (l'host A) non può effettuare il download diretto. In questo caso A invia ad B un messaggio di Push e chiede a B di connettersi ad A ed effettuare l'upload del file.



Gnutella: il Routing

Quando un peer riceve un **ping message** oppure un **query message** decrementa il TTL del messaggio

- Se $TTL=0$, il messaggio viene scartato
- Se il peer ha già inoltrato lo stesso messaggio (utilizzo di identificatori unici di messaggio), il messaggio viene scartato. Questo consente di evitare cicli.
- Altrimenti il messaggio viene spedito a tutti i vicini (escluso il peer da cui si è ricevuto il messaggio)



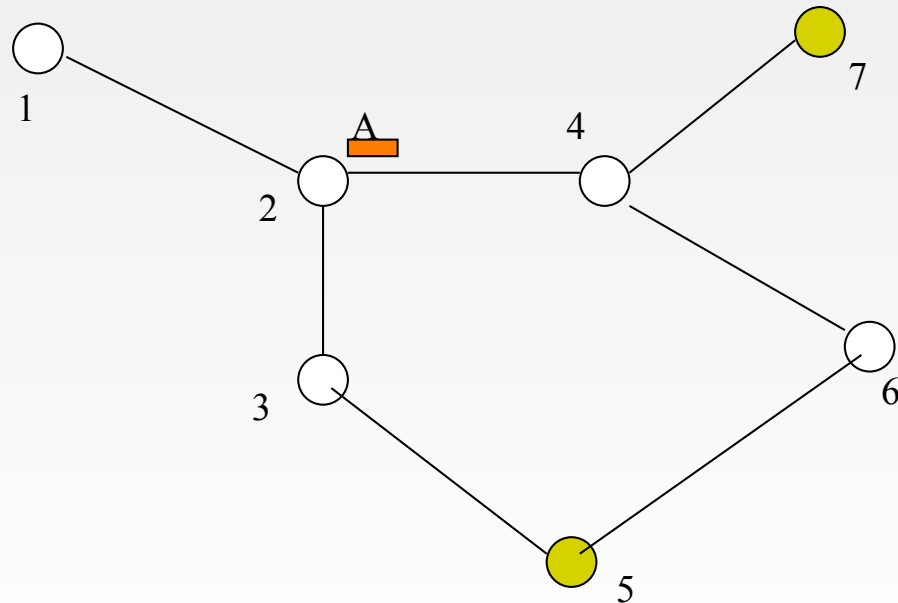
Gnutella: il Routing

Quando un peer riceve un **pong** oppure un **query hit**

- Il messaggio viene inoltrato al peer che aveva spedito il corrispondente messaggio di ping o il corrispondente messaggio di query (il peer deve memorizzare le relazioni tra messaggi ricevuti e messaggi inoltrati)



Gnutella: Meccanismo di ricerca

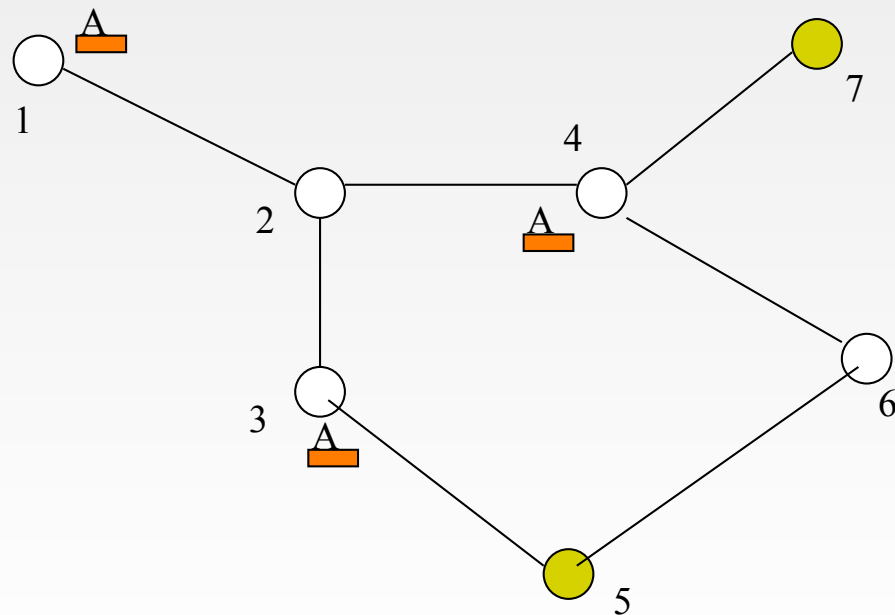


Passi:

- Il nodo 2 inizia la ricerca per il file A



Gnutella: Meccanismo di ricerca

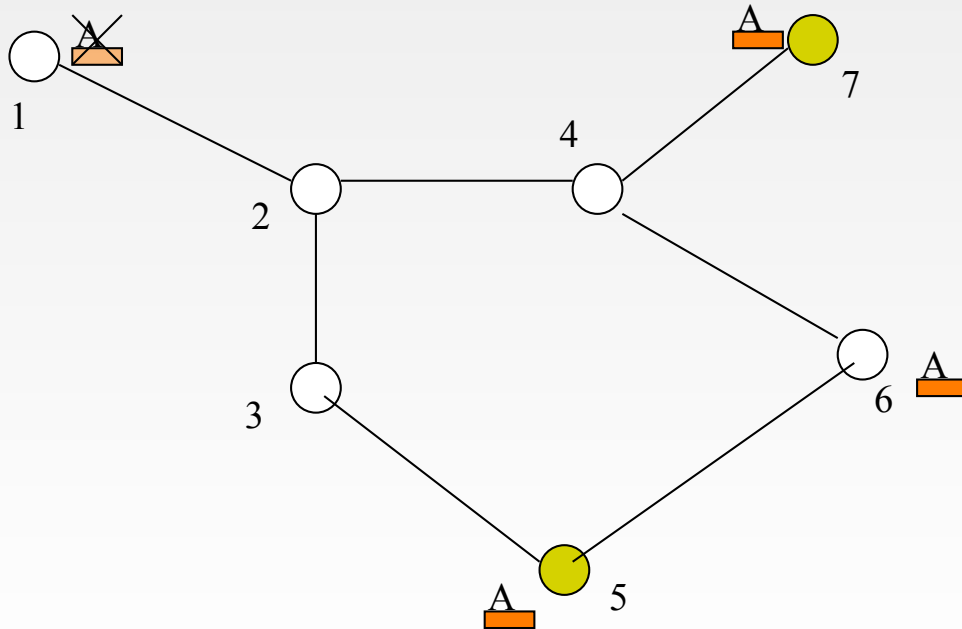


Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini



Gnutella: Meccanismo di ricerca

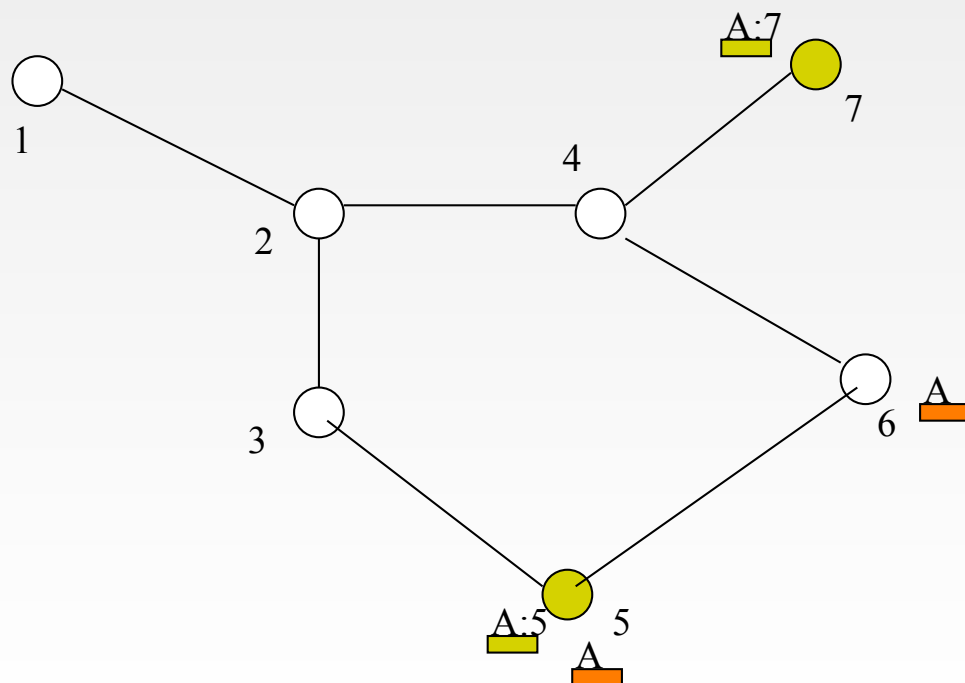


Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini
- I vicini inoltrano la richiesta



Gnutella: Meccanismo di ricerca

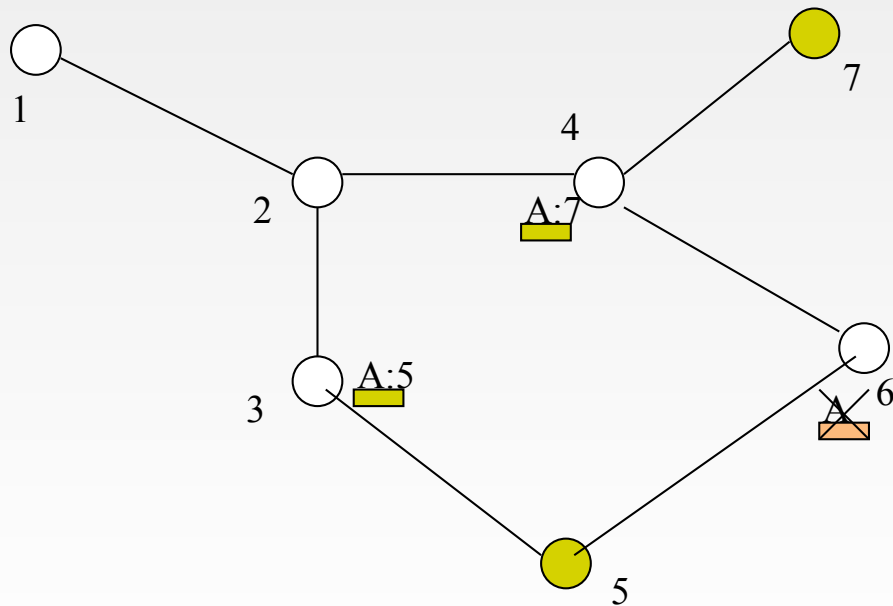


Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini
- I vicini inoltrano la richiesta
- I nodi che hanno il file A inviano una risposta



Gnutella: Meccanismo di ricerca

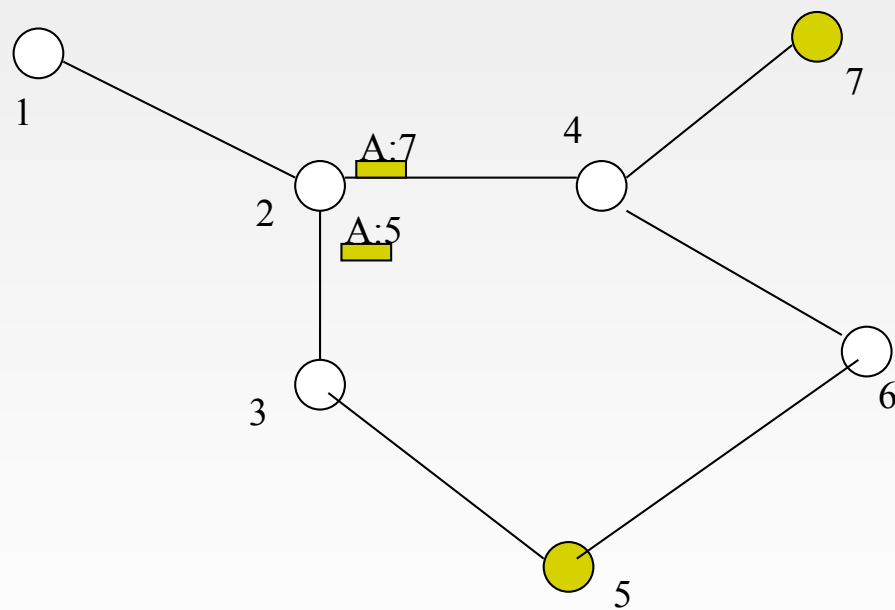


Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini
- I vicini inoltrano la richiesta
- I nodi che hanno il file A inviano una risposta
- La risposta viene propagata indietro fino al nodo 2



Gnutella: Meccanismo di ricerca

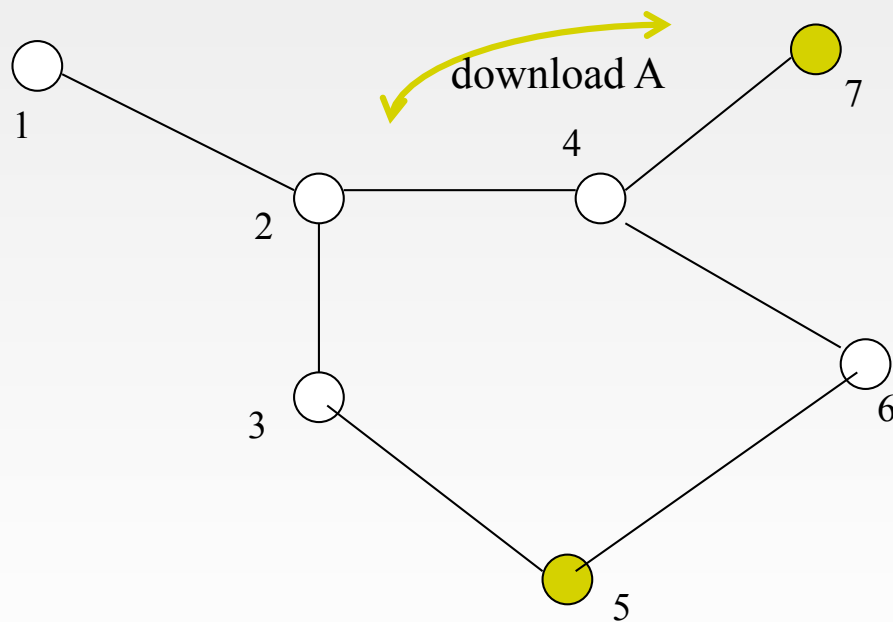


Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini
- I vicini inoltrano la richiesta
- I nodi che hanno il file A inviano una risposta
- La risposta viene propagata indietro fino al nodo 2



Gnutella: Meccanismo di ricerca



Passi:

- Il nodo 2 inizia la ricerca per il file A
- Invia il messaggio a tutti i peer vicini
- I vicini inoltrano la richiesta
- I nodi che hanno il file A inviano una risposta
- La risposta viene propagata indietro fino al nodo 2
- File download



Gnutella: Descriptors

GNUTELLA Descriptors

Descriptor
Header (22 byte)

Descriptor
Payload (variabile)

Descriptor types:

- Ping: scoperta di host e struttura rete.
- Pong: risposta al Ping (include l'indirizzo GUID del servente, oltre ad informazioni relative alla disponibilità della risorsa nella rete)
- Query: inoltro di una ricerca
- QueryHit: risposta a Query (contiene il GUID e info sul file)
- Push: meccanismo per gestire servents sotto firewall

Gnutella: Header

Struttura Generale dell'Header:

HEADER DEL MESSAGGIO: 23Byte				
GUID 16 Bytes	Function 1 Byte	TTL 1 Byte	Hops 1 Byte	Payload Length 3 Bytes

Descrive il tipo del messaggio (es: query, pong)

Lunghezza del messaggio che segue l'header

- **GUID:** stringa di 128 bits che identifica univocamente il messaggio
- **TTL (Time-To-Live):** quante volte il messaggio può ancora essere inoltrato dai serventi prima di essere eliminato dalla rete
- **Hops:** quanti serventi hanno già inoltrato il messaggio

$$TTL(0) = TTL(i) + Hops(i) \leq K \text{ (K in genere = 7)}$$



Gnutella: Tipi di messaggio

Ping : non contiene nessun payload

Pong:

Port 2 Bytes	IP Address 4 Bytes	Nb. of shared Files 4 Bytes	Nb. of Kbytes shared 4 Bytes
-----------------	-----------------------	--------------------------------	---------------------------------

Payload:

- **Porta** alla quale il peer che risponde accetterà una connessione
- **indirizzo IP** identifica il peer che invia il pong.
- **numero di file** che il peer condivide
- **numero di Kb** condivisi



Gnutella: Tipi di messaggio

Query:

Minimum Speed
2 Bytes

Search Criteria
n Bytes



Gnutella: Tipi di messaggio

- **Minimum Speed:**

Velocità minima (kb/sec) al quale l'host risponderà. Nelle versioni più recenti di Gnutella il campo è stato sostituito da un insieme di flags, tra cui

-> *Firewalled Indicator*: indica che l'host che ha inviato la query non può accettare connessioni perché si trova a monte di un firewall. In questo modo un peer che può soddisfare la query evita di inviare un query hit se si trova a sua volta a monte di un firewall.

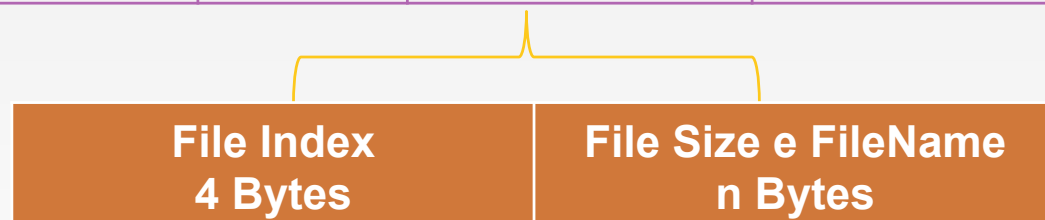
- **Search Criteria:**

Contiene una stringa costituita mediante la concatenazione di un insieme di keyword, separate da spazi bianchi. (es. *MiaCanzone.mp3*)

Gnutella: Tipi di messaggio

Query Hit:

Nb. of Hits 1 Byte	Port 2 Bytes	IP Address 4 Bytes	Speed 1 Byte	Result Set n Bytes	GnodeID 16 Bytes
------------------------------	------------------------	------------------------------	------------------------	------------------------------	----------------------------



- Numero di query hits presenti nel result set
- Port + IP Address (Individuano l'host che ha generato l'hit)
- Speed dell'host che genera il query hit
- Speed = Velocità (kb al secondo) del peer



Gnutella: Tipi di messaggio

Push:

Server Identifier 16 Byte	File Index 4 Bytes	IP Address 4 Bytes	Port 1 Byte
-------------------------------------	------------------------------	------------------------------	-----------------------

- **Servent Identifier:** ID del target servent (host che fa la richiesta) (16-byte string) - necessario per eseguire il push del file
- **File Index:** ID del file da inviare al target servent
- **IP Address:** indirizzo IP del target host per cui il file va uplodato
- **Port:** porta del target host