

# Sistemi Distribuiti

## Corso di Laurea in Ingegneria

*Prof. Paolo Nesi*

*2013 Parte 5: Sistemi P2P*

Department of Information Engineering

University of Florence

Via S. Marta 3, 50139, Firenze, Italy

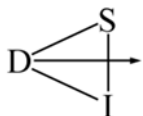
tel: +39-055-4796523, fax: +39-055-4796363

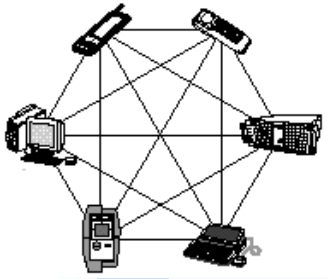
**Lab: DISIT, Sistemi Distribuiti e Tecnologie Internet**

<http://www.disit.dinfo.unifi.it/>

[paolo.nesi@unifi.it](mailto:paolo.nesi@unifi.it)

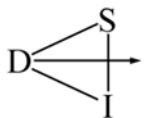
<http://www.dsi.unifi.it/~nesi>

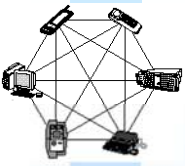




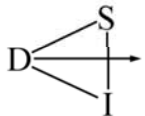
# Sistemi P2P

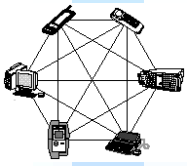
- Aspetti Generali, Applicazioni
- Evoluzione Storica
- Motivazioni per il P2P
- Requirements
- Architecture P2P e caratteristiche
- Ricerche e download multisorgente, BTorrent
- Reti P2P in Overlay
- Controllo e supervisione reti P2P
- Esempi: Skype, P2P per il B2B, basata su BTorrent
- Esempi: P2PTV, P2P webTV, progressive Download of audio/visual content
- Esempio: monitoraggio reti P2P, P2P distributed trust



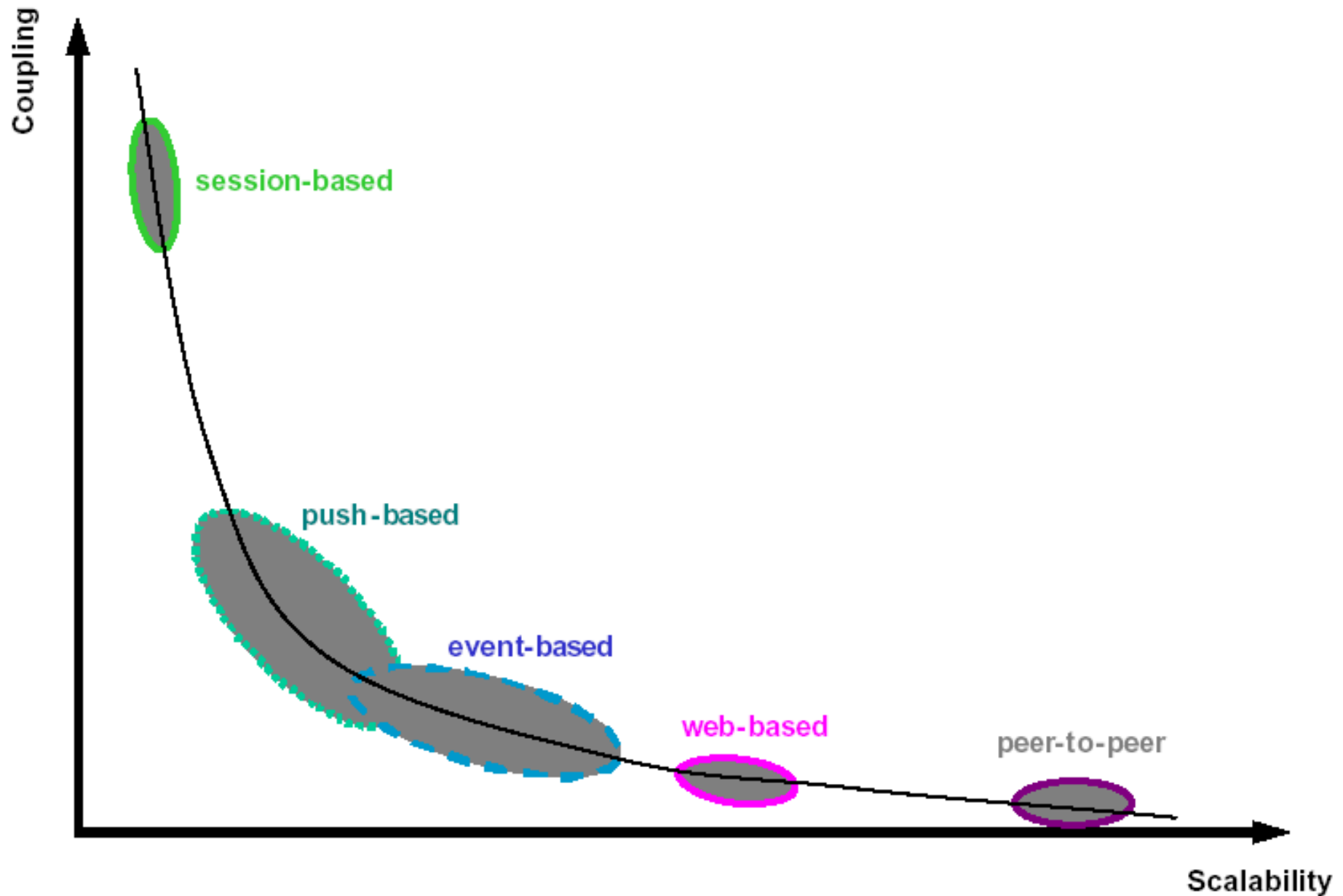


# Aspetti generali

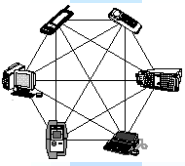




# Coupling and Scalability







# The P2P World: Current Status

## ○ first wave

- ♣ spreadsheet and word processors

## ○ second wave

- ♣ Internet and Mosaic ..... Netscape, ...IE

## ○ third wave

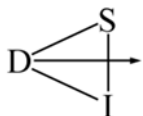
- ♣ Napster: over 60 million users
- ♣ AOL Instant Messaging: 100 million subscribers
- ♣ In search for “Killer Apps” for Networked Objects

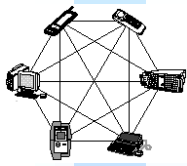
## ○ forth wave

- ♣ P2P on real time applications: streaming, WEB TV, 3D streaming,
- ♣ P2P on mobiles

## ○ Fifth wave

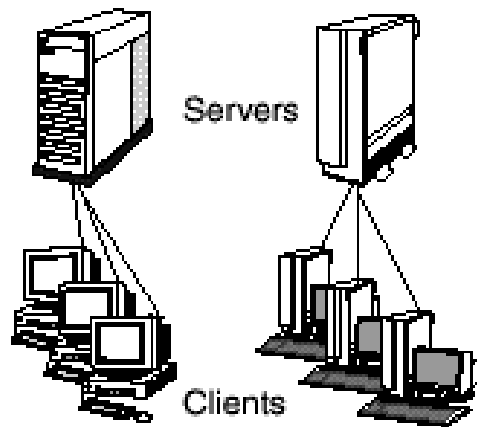
- ♣ P2P on applications: games, chat, netmeeting, etc.
- ♣ P2P on security: DRM,



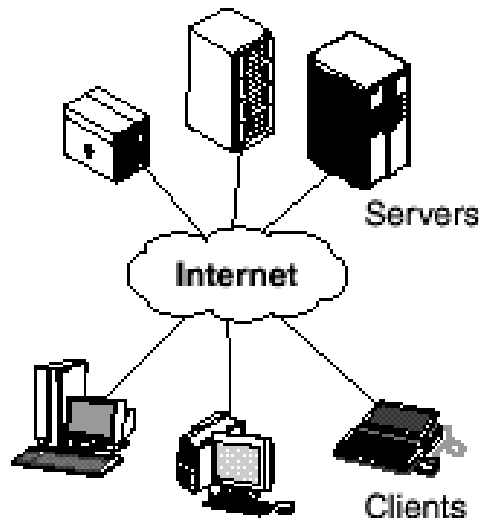


# Peer-To-Peer Computing

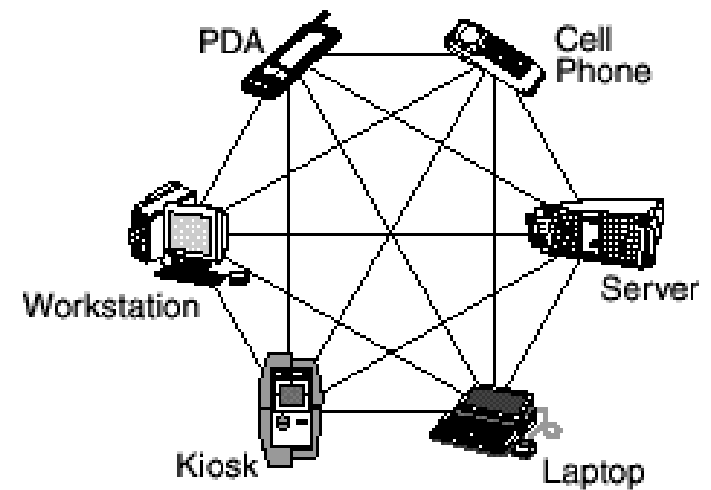
A network-based computing model for applications where computers *share resources via direct exchanges* between the participating computers



a. Client-server stacks with limited interoperability and homogeneous client and server systems



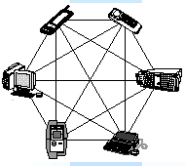
b. Web-based computing supports heterogeneous clients and servers



c. Peer-to-peer computing enables direct communication between peers and new interaction styles

Source: Sun Microsystems, Project JXTA, 2001





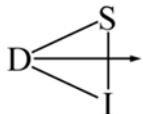
# Application areas

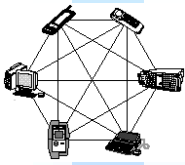
## Content and Resource sharing

- ♣ Network-wide file/document sharing (napster, eDonkey, Gnutella, Freenet, piratebay, emule, etc.)
- ♣ VOIP: Voice Over IP
- ♣ P2P CDN: Content delivering network
- ♣ P2P VOD (Video on demand), P2PTV, WebTV also in STB

## Distributed computation more GRID than P2P

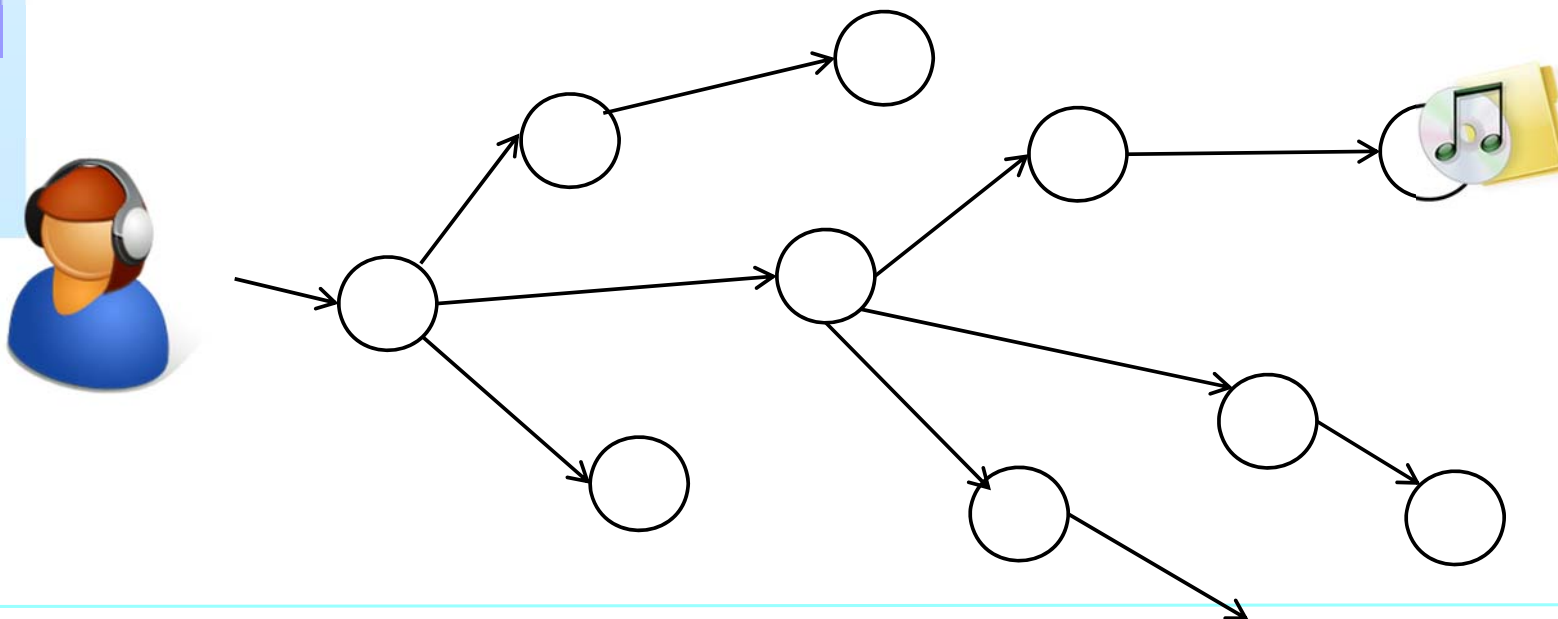
- ♣ Internet-based (e.g. United devices, entropia)
- ♣ Intranet-based ([www.datasynapse.com](http://www.datasynapse.com), NetBatch of Intel)
- ♣ Web testing (e.g., United devices)
- ♣ Esempio: gridella, etc....
- ♣ Resource sharing: seti@home, aids@home, folding@home

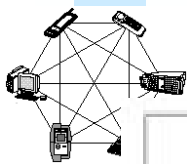




# P2P Applications for file sharing

- Napster, Gnutella, Freenet, Kazaa
- Emule, Emule Plus: both based on kademlia
- Mojo Nation
- BitTorrent (Azureus client)
  - ♣ BT based: PirateBay, Suprnova, isoHunt, TorrentSpy
- **Shareaza** supports protocols like: Gnutella, Gnutella2, eDonkey Network, BitTorrent, FTP and HTTP





## Normalized Aggregate Traffic Profile (Peak Hours, Fixed Access)

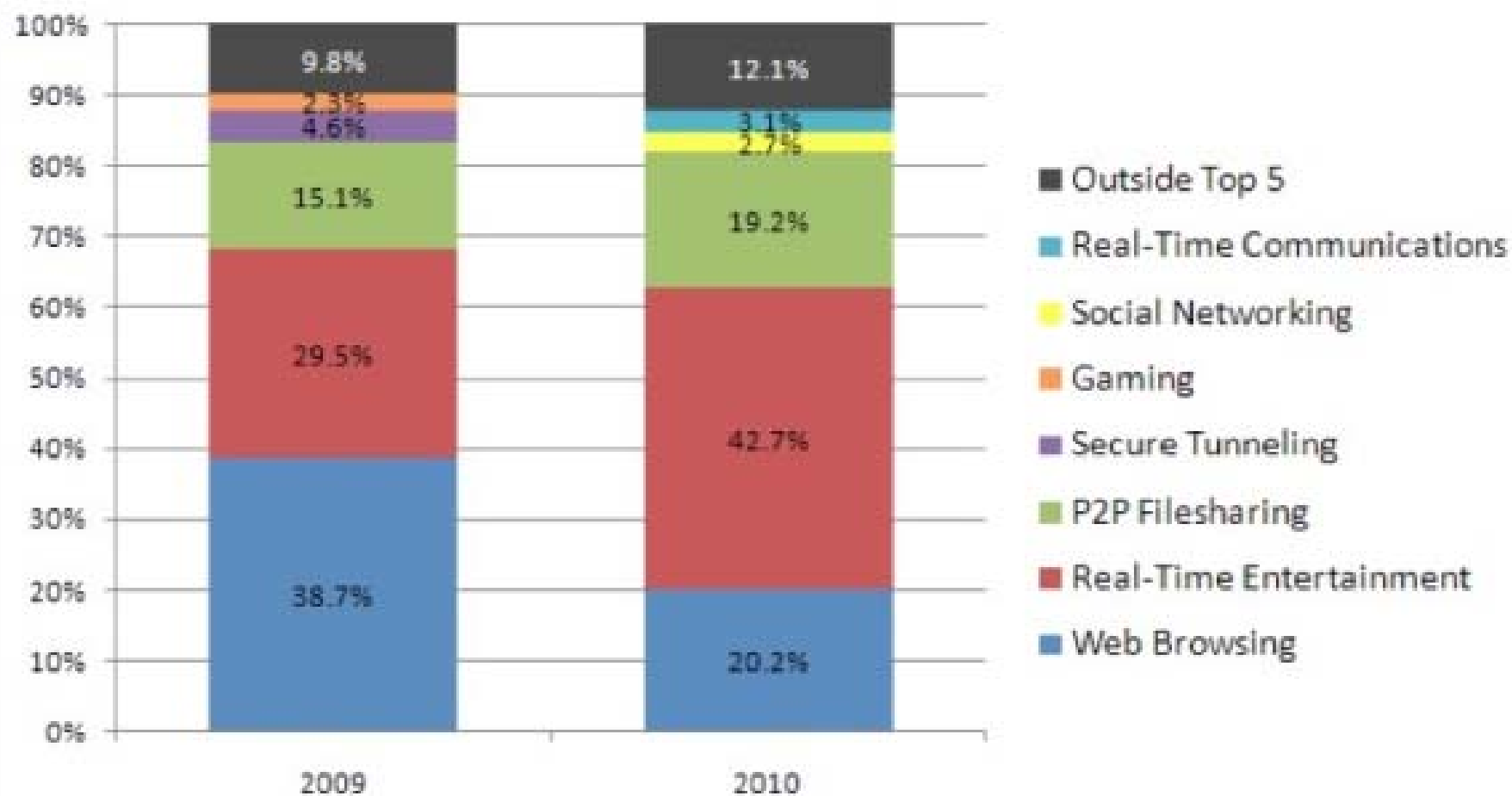
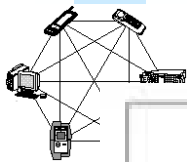


Figure 7 - North America - Normalized Aggregate Traffic Profile Comparison (Peak Hours, Fixed Access)



## Normalized Aggregate Traffic Profile (Peak Hours, Fixed Access)

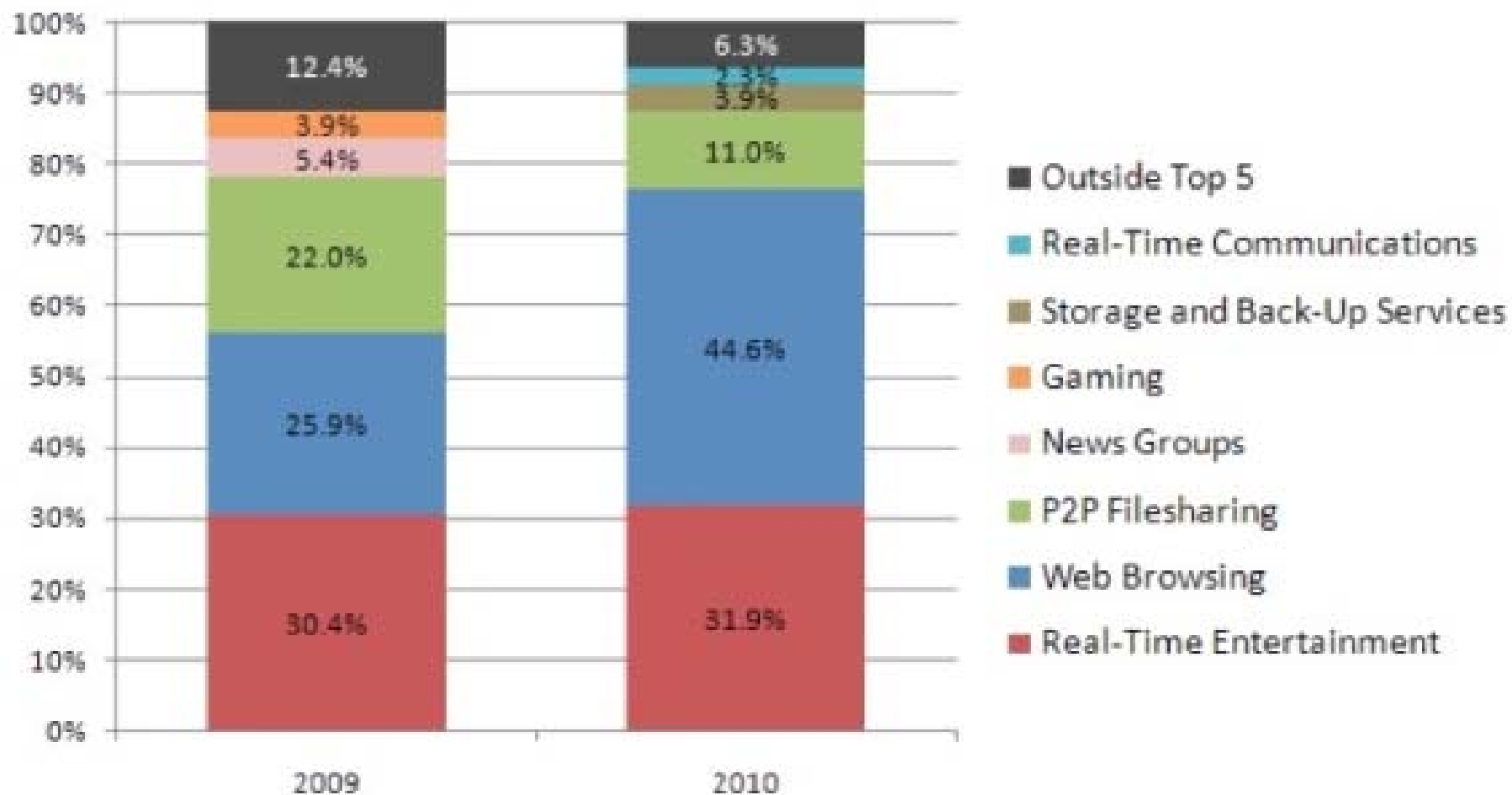
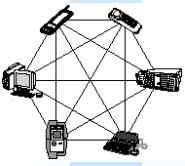
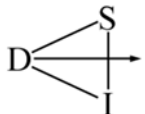


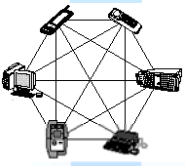
Figure 29 - Europe - Normalized Aggregate Traffic Profile Comparison (Peak Hours, Fixed Access)



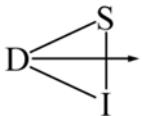
# Definition of Peer

- The peer is a node client of the P2P network
  - ♣ Each client peer has many files
  - ♣ Some in download and/or uploads
- The ***peer is a single thread***, process of download and/or upload, such as in BitTorrent Terminology
  - ♣ Each client node has many peers, typically no more than 5/10 at the same time.
- We can have a network that at a given time instant may have
  - ♣ 4Mpeers, 1.2Mfiles and 890Kusers
  - ♣ Some are seeders the other are passively reading only !

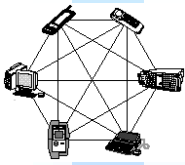




# Requisiti sistemi P2P







# P2P Main requirements

## ○ Creation of the P2P community of Peers/clients

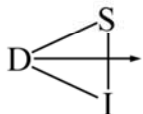
- ♣ Discovering of the Peers/clients to reach the resources

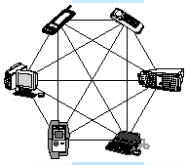
## ○ THUS: Discovering resources/services

- ♣ Resources may be objects, files or disk space, or computational power, users, etc.
- ♣ Allocation of resources/objects into the P2P network
  - ➔ Global Unique ID, GUID for the objects
- ♣ Indexing of the resources into the P2P network
  - ➔ Customization of query for getting information

## ○ Managing updates in the information shared

- ♣ In the information requested only
- ♣ Removing obsolete files and/or references
  - ➔ it is not always possible in P2P solutions in which it is tracked who has downloaded the file, or has the reference
- ♣ Notification of changes in the downloaded files, in the accessed resources, etc.
  - ➔ Versioning, replacement
  - ➔ Notification to all peers that have downloaded, please stop providing the last version.
  - ➔ Again: also in this case the system has to keep trace of who has the file or the reference

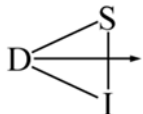


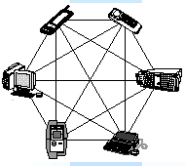


# P2P Main requirements

- **Interoperability** between the applications (peer client) built by using different P2P infrastructures
  - E.g.: JXTA and Microsoft P2P
  - Different protocols: from low level to high level
  - Different information management
  - Different Routing models
  - Different metadata
  - Different certification and authentication of content
- ♣ Esistono anche client che fanno query su piu' protocolli P2P
- ♣ La vera interoperabilita' sarebbe poter postare un file in una rete P2P e vederlo nell'altra rete P2P indicizzato direttamente
  - Client che utilizzano piu' protocolli fanno solo download da piu' canali
  - Partially true on BTorrent

~~R P P<sub>1</sub> R P P<sub>2</sub>~~

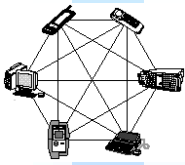




# P2P Main requirements

- **Scalability of the P2P solution**
- From 1 peer/resource/user to Millions and Millions
  - ♣ How is the capability in penetrating the network, Intranet to internet,
    - Discovery, query, versioning, maintenance, etc.
    - In intranet UTP can be used
    - In Internet UTP CANNOT be used
    - Peers need to perform the boot of the P2P network in Internet
  - ♣ Intrinsic limits of models,
    - for example a limit on the code for unique ID for files, users, etc..
    - How it may grow ?
  - ♣ how is the costs of the model to grow in terms numbers of Peers??
    - How many servers are needed ?
    - Which networks capability, bandwidth, they need ?
    - Which is the velocity that the grow may sustain ??

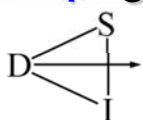


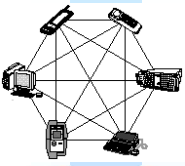


# P2P Main requirements

## Performance

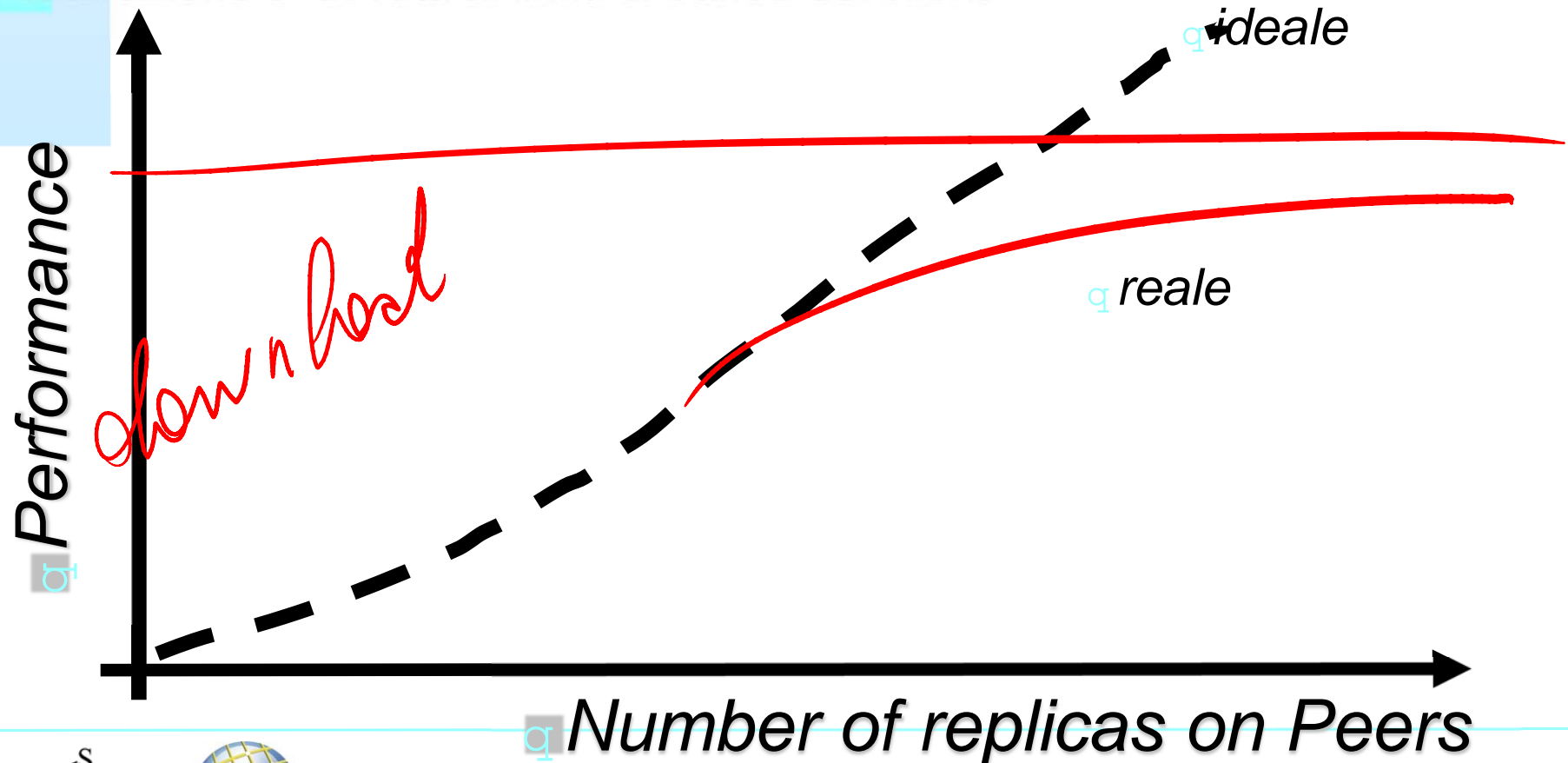
- ♣ Analysis and control of the network connections
  - ➔ Banda per esempio in termini di Mbps/Kbps
  - ➔ massimo numero di connessioni apribili/attive in ingresso (download) ed uscita (upload)
  - ➔ From the peer to a set of reference peers/servers
- ♣ Measuring CPU features:
  - ➔ fixing % of free CPU reserved
- ♣ Space on the HD disk:
  - ➔ space reserved, (maximum) space accessible, effectively free space used in the shared files, etc..
- ♣ Max Number of shared files, opened connections:
  - ➔ reserved and maintained visible
- ♣ Time to download, time to start the download
  - ➔ Time to perform the download, start-end time
- ♣ etc.



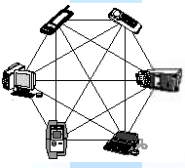


# Performance of P2P solutions

- Grafo dell'andamento delle prestazioni (download rate) in funzione del numero dei peer che hanno un certo file, misurato in un certo punto della rete.
- L'obiettivo della distribuzione e' arrivare a superare una certa soglia nel minor tempo possibile. Il superamento della soglia di seeding mi garantisce delle prestazioni ragionevoli in termini di capacita' di prestazioni per/gli l'utente/i e penetrazione della rete.
- La saturazione e' dovuta ai limiti di banda del client

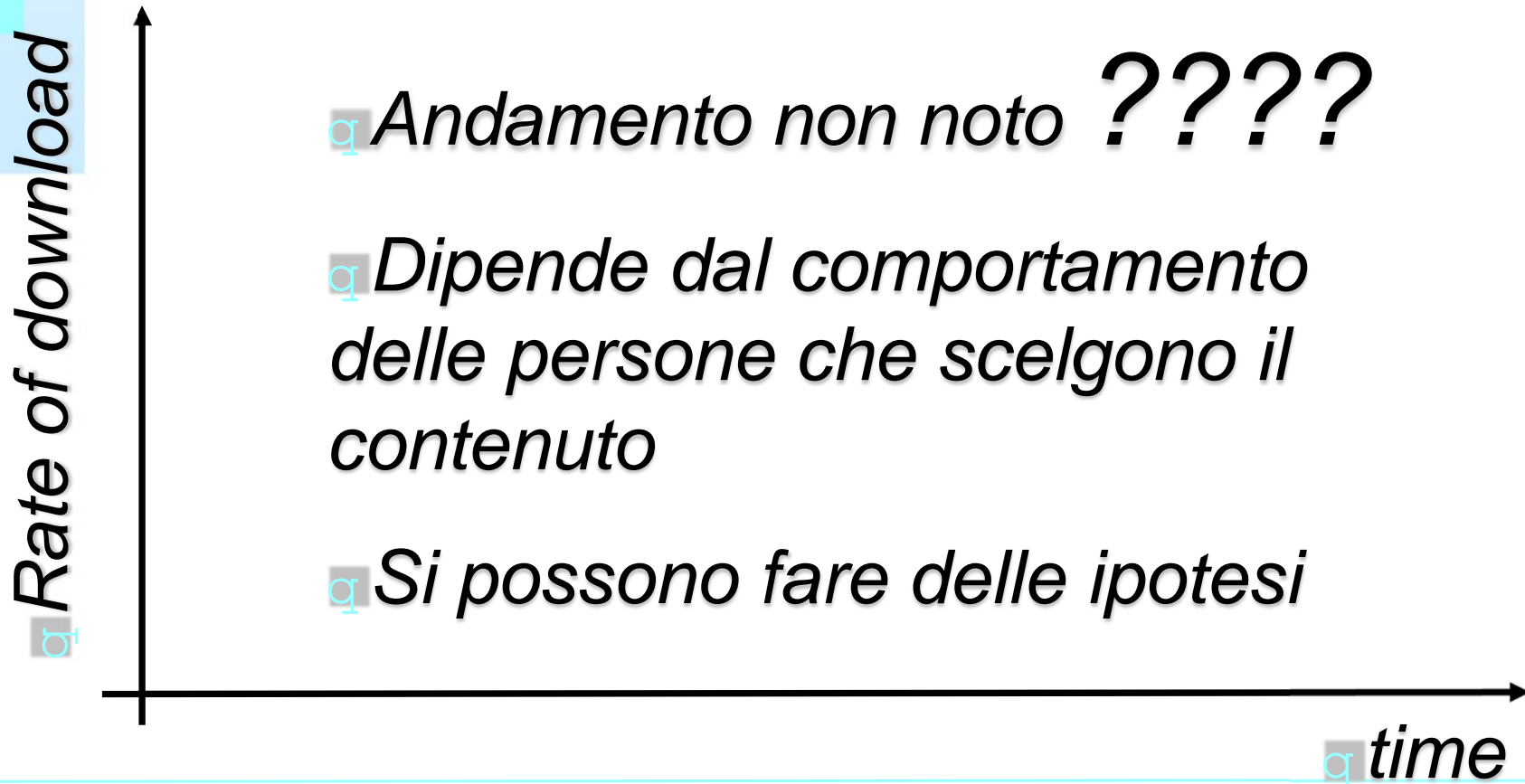


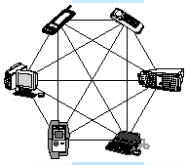




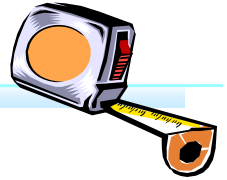
# Velocity of Penetration/diffusion/seeding

- Velocity by means of which a file becomes accessible (is seeded) in enough peers to guarantee a certain download rate in a certain area with a certain level of certainty





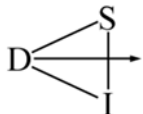
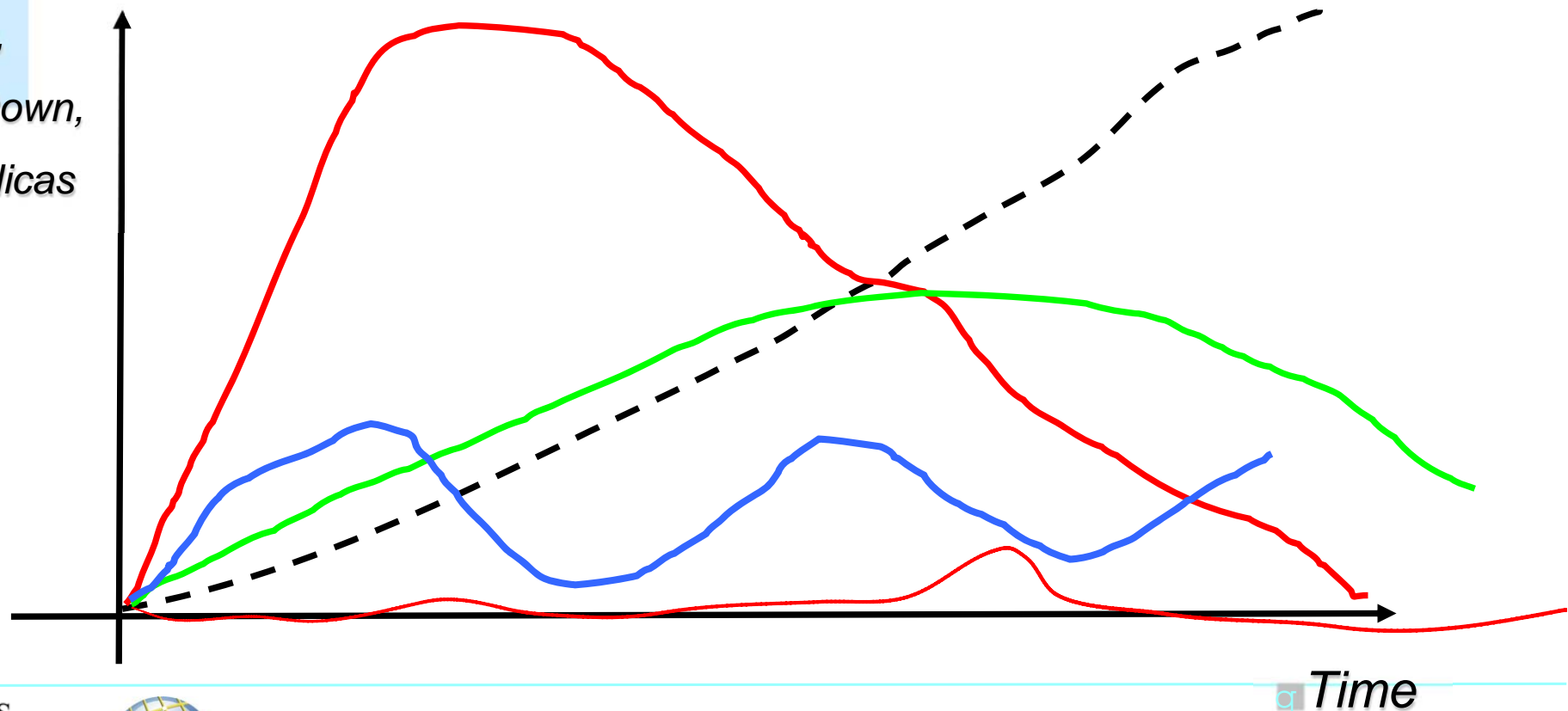
# Trend of Penetration in a given area

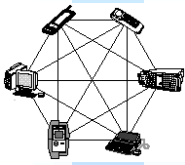


○ In general non predicatable, depends on:

- ♣ the users' interest and activities
- ♣ The content type and size
- ♣ The time, etc...

q # Down,  
q Vel of Down,  
q # of replicas





# P2P Main requirements

## ○ Assessment of peers, Assessment of user's behaviour (single or cluster)

♣ For:

- organizing requests in a queue on the basis of past assessment and scoring of peers
- Providing better services to more virtuous peers
- Recognising bonus to more virtuous peers, if the P2P network is created for e-commerce

♣ Assessment for reputation and scoring of peer behaviour

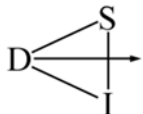
- Number of provided files, bandwidth, behaviour (leaving files)

♣ Assessment for repudiation of peers

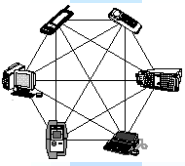
- Performance evaluation of peers, etc.
- Not enough: CPU power %, disk space, files to be downloaded, sockets open,

♣ Assessment of peers to exploit them as super-peers

♣ etc.







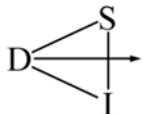
# P2P Main requirements

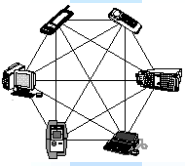
## ○ Security and Trust of users

- ♣ Authentication of users
  - ➔ Identification of the user in terms of name/surname, etc., or in terms of a simple UID, watermark
    - Knowledge of who has posted the file
- ♣ Privacy is typically preferred by P2P users
  - ➔ Privacy vs Authentication of users and Peers

## ○ Security and Trust of Content

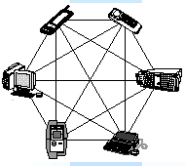
- ♣ Data/file/object certification: consistency
  - ➔ Authentication lead to higher level of trustiness
  - ➔ Trust of metadata and data, certification of content:
    - ➔ per verificare/riconoscere la firma del content,
    - ➔ garantire la consistenza fra metadati e dati
- ♣ Authorization to delivering and use content
  - ➔ Controllo dei diritti, Digital rights management
  - ➔ gestione dei diritti/rights, licenze, etc.





# P2P Main requirements

- **Security and Trust of Peer applications**
- To be sure that the Peer Client is working according to the rules of the community
  - ♣ In terms of respecting scoring, not presenting UserID of others, etc.
  - ♣ See later the routing overlay
- authentication and certification of peers devices
  - ♣ Tool certification/fingerprint to know who is the tool owner and avoid to assign to him a score of another user.
  - ♣ Certification that the Peer has not been manipulated to provide a behavior non conformant to the rules of the community/protocol



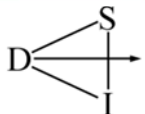
# P2P Main requirements

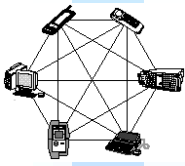
## ○ Robustness, Fault tolerance

- ♣ Robustness with respect to eventual fault of
  - Single peers
    - Interruption of downloads
    - Interruption of query/interrogation service
    - Interruption of instradation service (see Routing Overlay)
  - SuperNodes that permit the indexing and/or the boot of the P2P community
  - Network problems, turn off/on of the peers

## ○ Definition of solutions for recovery from failure

- ♣ Recovering the interrupted download of file when it is
  - monolithic: total restart
  - segmented: restart of the segment
  - Choosing a different Peer from which the download can be performed
- ♣ Duplication of resources (usage of strategies for duplication)

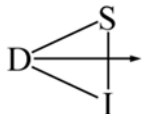


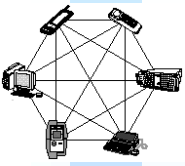


# P2P Technological Challenges

## ○ Architecture

- ♣ Usable on different platforms
  - ➔ Java has been the most selected
- ♣ Interoperability between the applications built on different P2P infrastructures (diff. protocols, languages, etc.)
- ♣ Controllability of Peers
  - ➔ Monitoring of user/peer behavior
- ♣ Performance, Scalability
- ♣ Fault tolerance
- ♣ Security: Privacy, Trust of content and of Peers

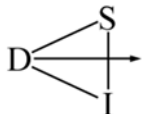


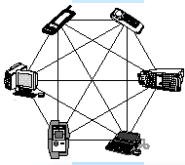


# P2P Technological Challenges

## ○ Architecture

- ♣ Efficient in joining the network, discovery if needed
- ♣ Efficient setup of the P2P community
- ♣ Query/search of resources
  - ➔ Complex queries: such as those based on SQL or RDF, based on semantics: title xx, author YY, ....
  - ➔ Connection with local databases: ODBC, JDBC, etc.
- ♣ Deleting of files
  - ➔ Removing from the network
- ♣ Changing of files
  - ➔ Notification of changes in the files posted/changed on the network
  - ➔ versioning
- ♣ QOS: Quality of Service
  - ➔ performance in *querying* and in the *download* for B2B and for B2C





# P2P Technological Challenges

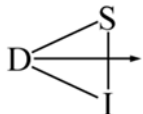
## ○ Development capabilities

### ♣ Several Groups:

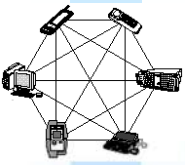
- DHT libraries
- JXTA libraries
- BTorrent libraries
- java (JXTA), open source, ...
- Microsoft P2P Tool Kit, .....
- Others are C++, etc.
- Many of them are open source projects

### ♣ at DISIT Lab, DSI, University of Florence:

- DIMOB P2P
- AXMEDIS P2P, derived from BT
- Bamboo based DHT P2P

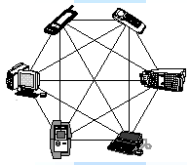






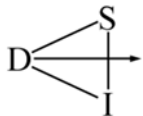
# Architetture P2P



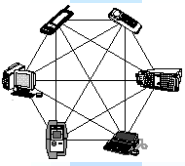


# Architetture P2P

- **Concentrate, centralized**
- **Distribuite, Distributed, decentralized**
- **Hierarchical or hybrid**







# Centralized P2P Architectures

- **Concentrated, centralized**

- ♣ One server and N peers; in some cases, more servers
- ♣ Example: Napster (central index)

- **Also called “Server-based” which may support:**

- ♣ Login, registration of peers to the central server
- ♣ Boot: performed asking to the server to get list of nodes
- ♣ Search: performed asking to the server
- ♣ Collection of data, index, query, etc.
  - ➔ Table to know where the files (their replicas) and their segments are: obj45: n3, n4, n56, n78

- **Server Problems: fault, size, performance, cost...**

- **Gli scambi dei file/risorse possono essere:**

- ♣ Centralised or P2P, multisource



<http://www.napster.com>

direct swapping MP3 music files  
over 50 million users by mid-2001

# Napster



**BearShare**

Hosts | Search | Downloads | Uploads | Monitor | Statistics

Host	Origin	Status	Time	Shared	% In Out Dup	Bandwidth	Version
enigma...	Australia	Auto	1m		65 34 6	57 Kbps	
adsl-20...	BellSouth	Auto	1m	242/983 MB	54 45 4	20 Kbps	BearSh..
u97n21...	Canada	Auto	1m	2/20 MB	78 21 3	31 Kbps	
006man...	Charter Communications	Auto	1m		27 72 9	8.8 Kbps	
bhc2.res...	Cornell University	Auto	2m	54/208 MB	73 26 2	144 Kbps	
ceres.as...	Croatia/Hrvatska	Auto	2m		47 52 5	24 Kbps	
us-6132...	Florida State University	Auto	1m	93/346 MB	49 50 5	66 Kbps	
nas1-83...	France	Auto	1m	23/1.0 MB	47 52 5	4.8 Kbps	
pc09.m...	Germany	Auto	1m		77 22 2	104 Kbps	
hive.telia...	Latvia	Auto	1m		75 24 4	77 Kbps	
fcalabro...	Princeton University	Auto	2m		59 40 3	30 Kbps	
cs16292...	RoadRunner	Auto	1m		71 28 2	33 Kbps	
h24-64-8...	Shaw Cable	Auto	2m		52 47 3	29 Kbps	
36-BAR...	Spain	Auto	2m		22 77 1	5.5 Kbps	
camp05...	Sweden	Auto	1m		60 39 7	47 Kbps	
d195-92...	University of Oregon	Auto	1m		75 24 2	36 Kbps	

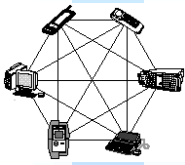
To connect manually, enter the IP address or host name here Connect

☐ Auto connect to  hosts Hosts 22

☒ Accept incoming hosts Bandwidth 845 Kbps

About... Community... Setup... Exit Help

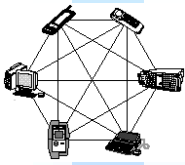




# Napster Pros and Problems

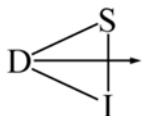
- **Veloce per l'entrata nella comunità** tramite server centrale o server locali ma sempre tramite server
- Anonimità degli utenti
- **Nessun controllo** su dati coperti da IPR (intellectually property right) e questi venivano centralizzati (come indice) in modo non autorizzato
  - ♣ Questo problema e' stato risolto nella versione attuale non molto diffusa ed apprezzata, dalla massa
- **Sicurezza:**
  - ♣ Contenuti non certificati
  - ♣ Utenti non autenticati
  - ♣ Non accesso a database, query molto semplici
- **Protocollo proprietario**, filtrato da firewall
- **Scarsa scalabilità**
  - ♣ Costi elevati di gestione

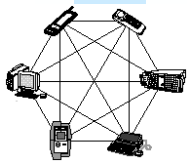




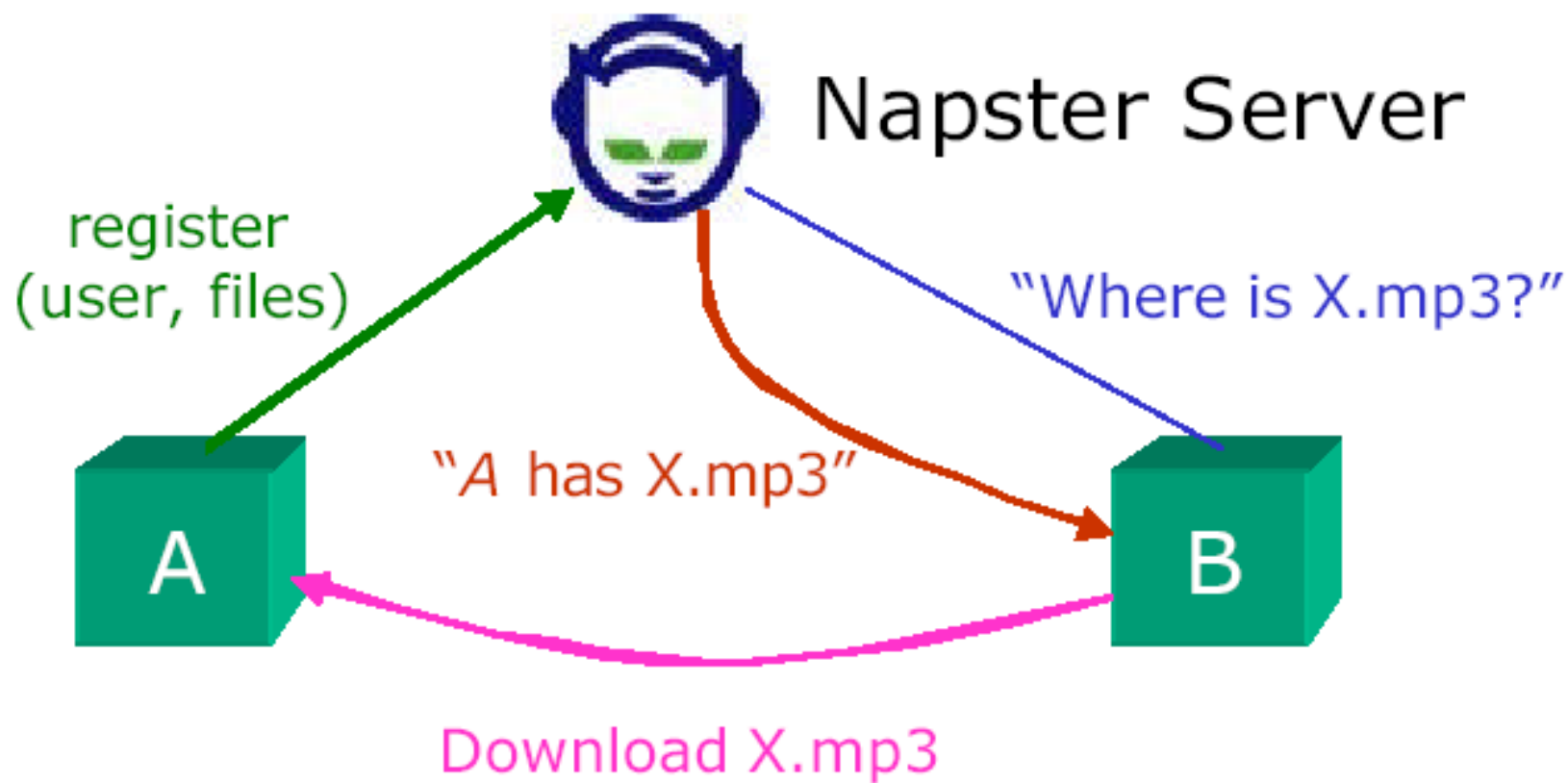
# Napster

- Architettura centralizzata per la condivisione di file
- Query centralizzata
- Copie dei contenuti centralizzate, in parte
- Al crescere del numero degli utenti e' necessario aumentare le prestazioni e lo spazio disco del server centrale che dai il servizio
- Una o piu' soluzioni:
  - ♣ Fare un cluster di server
  - ♣ Duplicare le risorse su piu' server
- Aumento dei costi

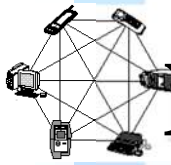




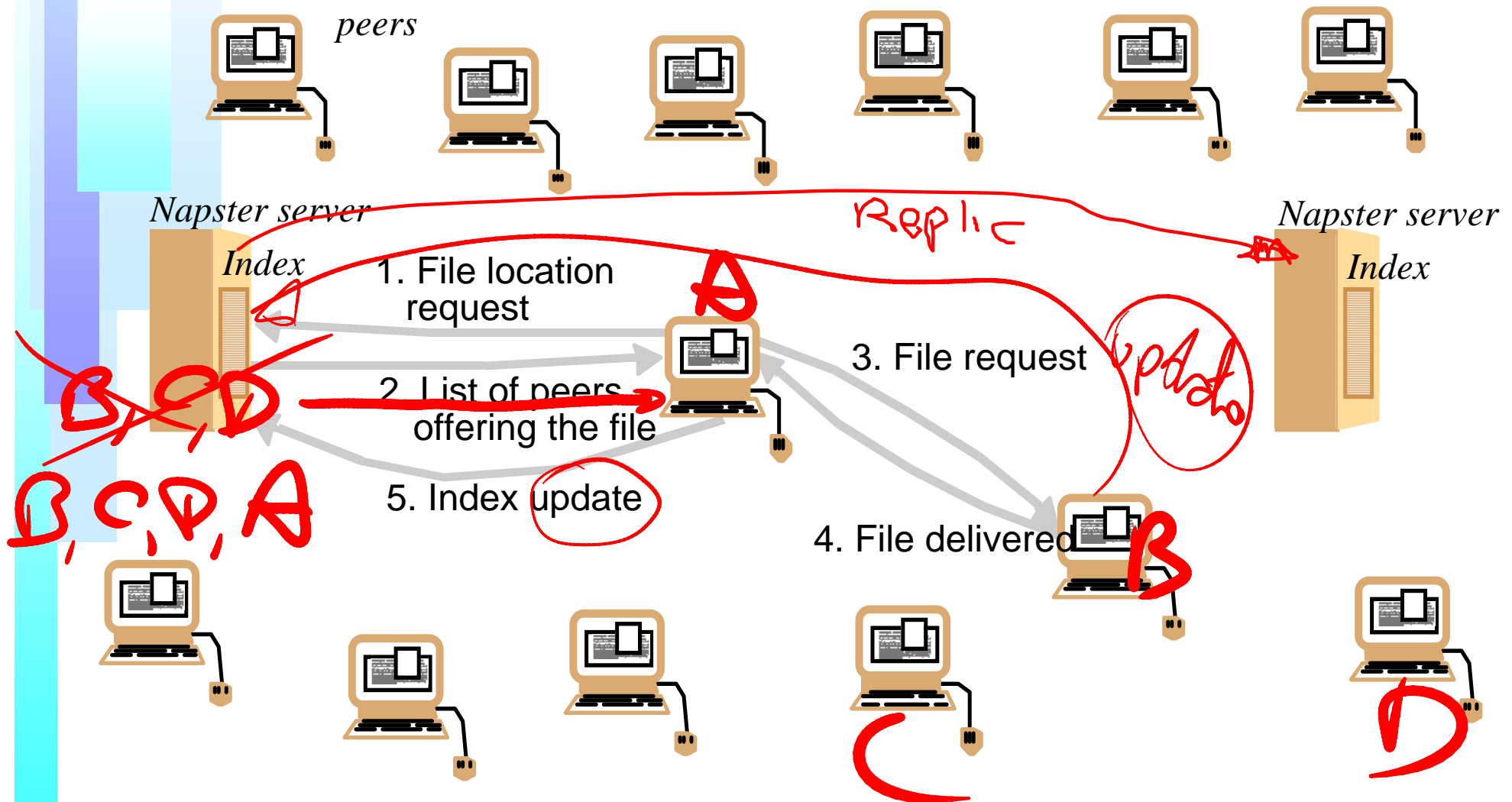
# Napster: search files

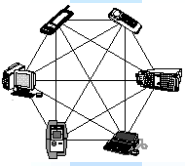






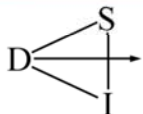
# Napster: peer-to-peer file sharing with a centralized, replicated index

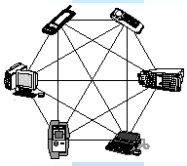




# Distributed P2P Architecture

- **Distributed, decentralized**
- **Also called Pure P2P networks**
  - ♣ N peers, all identical
  - ♣ Example: Gnutella (gnutella hosts), freenet
  - ♣ Boot: massive discovery, highly complex/net-costs
  - ♣ Search: fully distributed!, high complexity
  - ♣ No problems of fault
    - ➔ redundancy of information and services
  - ♣ The most common problems:
    - ➔ Low performance on search and discovery (distributed), etc.
    - ➔ No administration, no certification
    - ➔ No control on the network





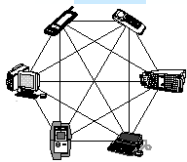
# Gnutella

- March 2000, Molto semplice
- Meccanismi di distribuzione e monitoraggio dei file in HTTP
- Non vi sono meccanismi di sicurezza
  - ♣ Non e' possibile autenticare gli utenti
  - ♣ Gli oggetti non sono certificati
  - ♣ *metadati e contenuti possono non essere consistenti*
- <http://www.gnutelliums.com>
- [www.gnutella.com](http://www.gnutella.com)
- Several implementations of Gnutella clients
  - ♣ For example: limewire,

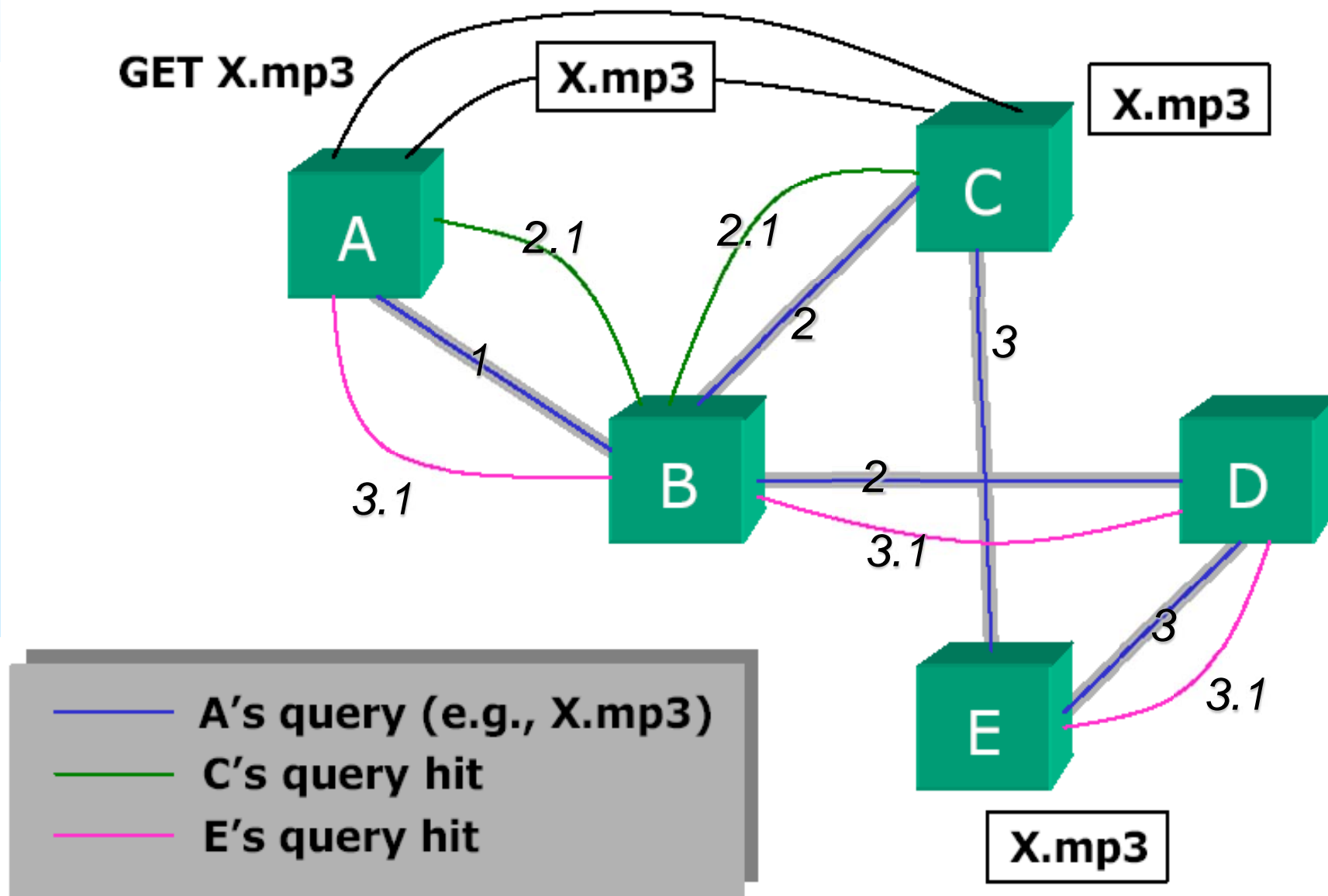


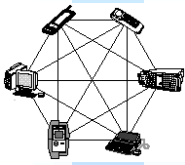






# Gnutella: searching (via routing)





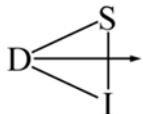
# Hybrid P2P Architettura

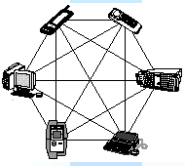
## ○ Hierarchical, hybrid

- ♣ Mix of centralized and decentralized
- ♣ N peers *not all identical* (at least in the role)
  - some with the role of local **concentrator** that can be activated when needed, the so called “super peers”
- ♣ Example:
  - Fast Track
  - Emule: with the servers for boot

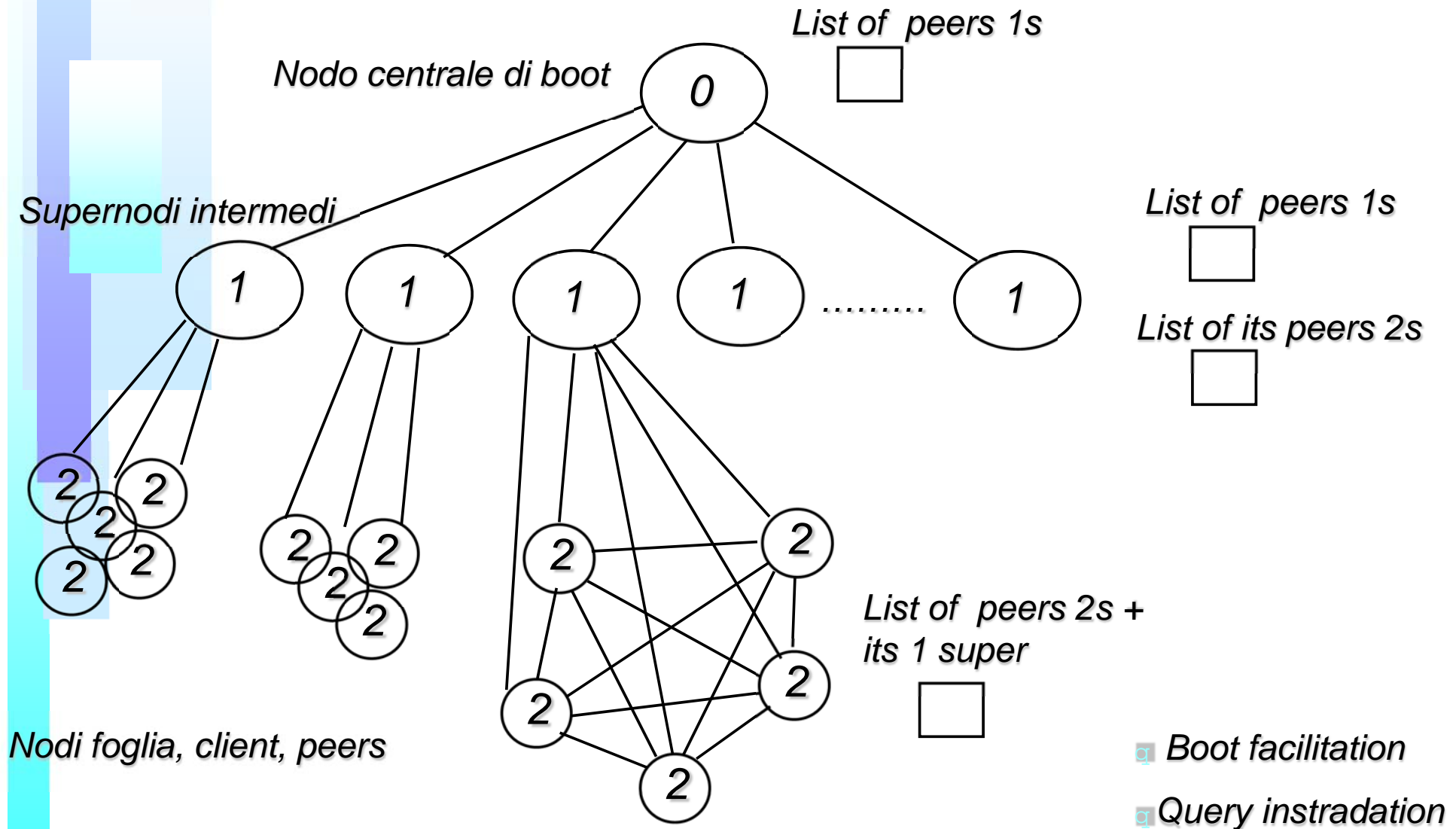
## ○ Super peers

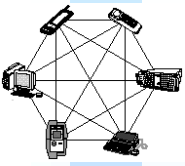
- ♣ facilitating the starting/booting of the peer network,
- ♣ recovering the list of closer peers
- ♣ May create a restricted community around which the content is shared
  - marginally connected with others communities





# Hierarchical/Hybrid P2P Network





# Main Functionalities of Nodes

## ○ **Nodo Centrale di Boot:**

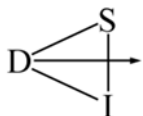
- ♣ NodeList GetList(): to provide list of supernodes level 1
- ♣ AddNode(node): to add a supernode of level 1 to the list
- ♣ DelNode(node): to remove a supernode of level 1 to the list, performed by missing a ping for a while
- ♣ Bool Alive(): to verify if the node is alive

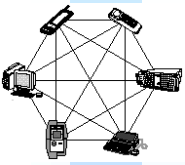
## ○ **Level 1 Node :**

- ♣ NodeList GetList(): to provide list of supernodes level 2
- ♣ Alive(), AddNode(node), DelNode(node), as above
- ♣ Result PassQuery(query): to pass a received query to lower level nodes in its Node2List
- ♣ Result RedirectQuery(query): to pass a received query to nodes of its level: Node1List

## ○ **Level 2 Node: are almost all notifications**

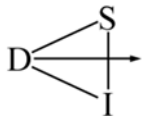
- ♣ It does not receive any command from other nodes on the list
- ♣ Result Query(query): another node is making a query
- ♣ Data GetFileSegment(GUID): to get a file segment
- ♣ Alive(), etc.



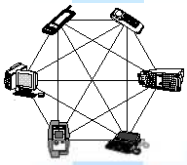


# KaZaA example

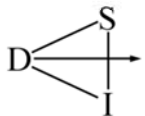
- It is a **semicentralized P2P solution**
- **Super-peers maintain** info/DB with:
  - ♣ file identifiers, their children are sharing
  - ♣ metadata (file name, size, contentHash, descriptors)
  - ♣ IP addresses of children
- **peers frequently exchange list of super-peers**
  - ♣ Peer clients maintains list of 200 super-peers
  - ♣ Super-Peers maintain a list of thousands of SPs
- All of the signaling traffic between peers is encrypted
  - ♣ Lists, Metadata upload, Queries and replies
- File transfer among nodes is not encrypted
- TCP is used for both file transfer and other communications



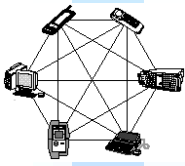




# Ricerche e Download multisorgente

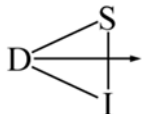


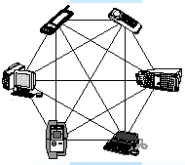




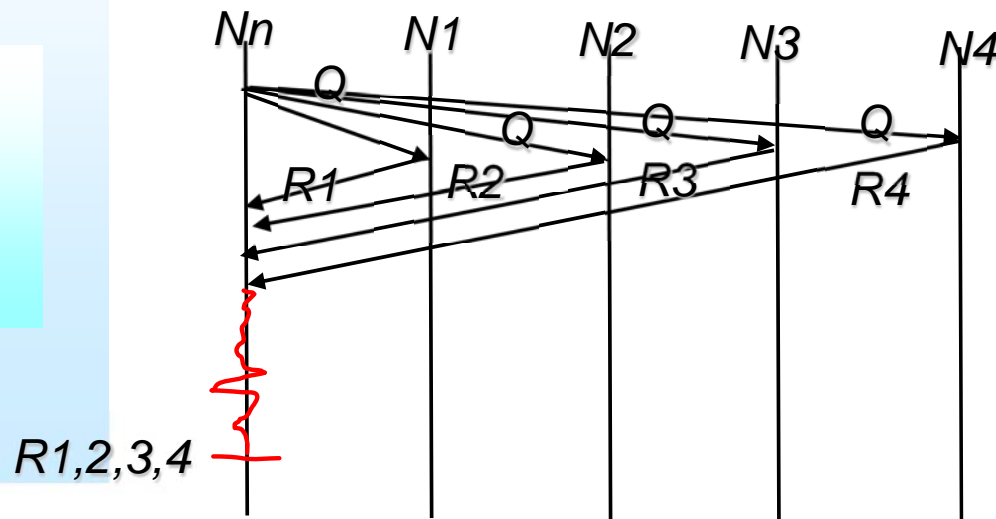
# Risposte delle Query

- **A fronte di una query**
- **Risposta con file singoli monolitici:**
  - ♣ FileABC: N1, N34, N56, N58, N67
  - ♣ FileFGH: N5, N4, N75, N6, N88, N92, N60
- **Risposta con file a segmenti:**
  - ♣ FileABC: N1(1,3,4,56), N34(345,3,2,1), N56(4,56)
  - ♣ FileFGH: N5(all), N4(3,5,7), N60(56,78,125)
  - ♣ All means 100%
- **Where for Nx we intend:**
  - ♣ IP address, Port, position in the list, estimated time to wait, i consider you a friend, location, bandwidth, etc.

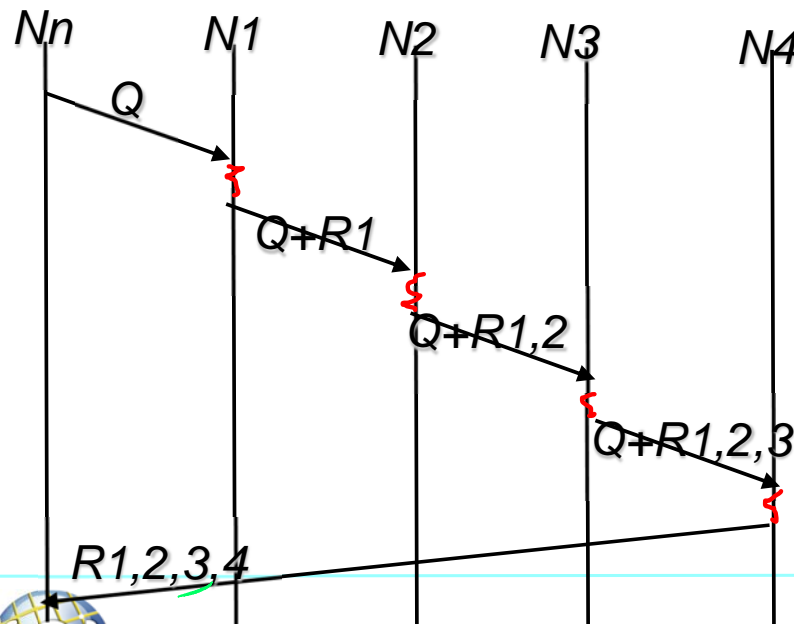




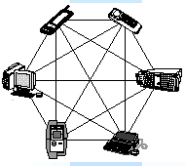
# Recombination of Query Results



- $N_n$  invia la query
- $N_n$  deve ricombinare  $R_1, 2, 3, 4$
- i duplicati possono essere molti, etc.

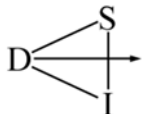


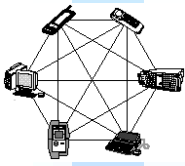
- $N_n$  invia la query
- $N_i$  riceve la query con i risultati del nodo precedente e ne fa l'unione con i propri, li passa al nodo successivo
- I duplicati vengono integrati via via
- $N_n$  non ricombina risultati
- Carico distribuito
- Soluzioni anche ibride



# Download Multisorgente

- File diviso in Parti di dimensioni ragionevoli per la rete, qualche Kbyte o decina di Kbyte:
  - ♣ F1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- Nodi hanno delle parti nella loro memoria cache:
  - ♣ N1: 1, 3, 5, 7, 8
  - ♣ N2: 2, 4, 5, 7, 8, 10
  - ♣ N3: 5, 6, 2, 9
  - ♣ Etc..
- Alcuni nodi possono anche averle tutte, cioè' il file completo
- Un nodo puo' scaricare parti diverse da nodi diversi anche allo stesso tempo sfruttando in questo modo un parallelismo
  - ♣ P.es.: N3 puo scaricare 4 e 10 da N2, ed 3, 1, 8 da N1

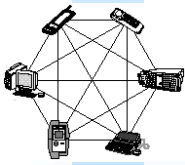




# Download Multisorgente

## ○ Politiche per scaricare le parti/file dai nodi:

- ♣ Il Nodo permette lo scarico in base ad una coda di richieste
  - ➔ Il nodo che chiede viene messo in coda, quando quelli prima hanno avuto almeno una parte vengono messi in fondo alla coda
  - ➔ Il nodo può salire nella coda se ha da dare delle parti anche lui all'altro nodo, per esempio
- ♣ Il Nodo ha una limitazione
  - ➔ sul numero di scaricamenti contemporanei
  - ➔ sulla banda sfruttata in uscita e/o ingresso
- ♣ Un Nodo può acquisire un credito (uno score/voto) in base al suo comportamento nel lasciare scaricare file o nel permettere in uscita una banda larga.
  - ➔ In base a questo credito potrebbe/dovrebbe avere delle facilitazioni/score in caso di richieste, per scalare delle posizioni nelle code, etc.



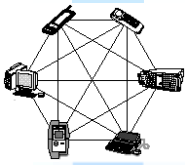
# bitTorrent

## ○ Programma di file sharing

- ♣ Principalmente P2P, ma con seme iniziale su semplici pagine WEB, .torrent file
- ♣ Open Source
- ♣ Soluzioni in vari linguaggi, C++, Java, etc.

## ○ prestazioni migliori per file di grosse dimensioni

- ## ○ L'ipotesi e', come per la maggior parte dei sistemi i P2P:
- ♣ che quando uno ha un file anche intero/completo lo continui a condividere con gli altri e non lo tolga dalla directory che contiene i file visibili per gli altri



# bitTorrent

## ○ Idea:

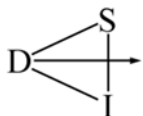
- ♣ Un file *.torrent* contiene informazioni su come prendere le parti del file DATI, dove prenderle
  - ➔ chi sono i nodi che hanno porzioni di quel file
- ♣ Il file DATI viene diviso in parti e queste in segmenti
- ♣ Il primo pezzo che viene scaricato e' casuale, i successivi vengono scelti in modo da dare precedenza al più raro, in modo che la sua rarità si attenui visto che viene copiato su di un altro nodo.

A ha 1,4,5,7,8, e metà di 3

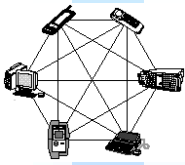
B ha 2,3,4,5, e metà di 6

C ha 2,5,6,7 e metà di 8

- Quando A finisce con la parte 3, richiede una nuova parte.
- Il programma analizza lo stato generale e trova i pezzi 1 e 6 che sono i più rari.
- Fra questi, A ha bisogno solo della parte 6, così A inizia a scaricare 6
- poi passa agli altri segmenti

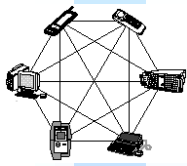






# Il tracker in bitTorrent

- Quando si effettua una query, la risposta e' una lista di file e per ognuno di questi un file *.torrent*
- Il file *.torrent* contiene informazioni su chi ha i segmenti del file, eventuali duplicazioni, etc.
- Il nodo contatta gli altri peer e parte con lo scarico in base alla strategia vista
- Archivi/tracker diversi possono avere file *.torrent* diversi e questi possono o meno essere mantenuti aggiornati con le informazioni su chi ha il file in questione
- Un client può essere connesso a uno o più *tracker* per la ricerca dei file *.torrent*



# Azureus: Monitoraggio dello stato

**Azureus** File View Language Help

My Torrents 10.2% :

General Details Pieces Files

Downloaded 10.2 %

Availability 27.995

Transfer

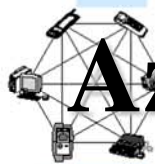
Time Elapsed :	1h 21m	Remaining:	9h 28m 1.52 GB	Share Ratio :	0.217
Downloaded :	202.4 MB ( 2.4 MB discar	Download Speed:	53.3 kB/s	Hash Fails :	10 ( ~ 10.0 MB )
Uploaded :	44.1 MB	Upload Speed :	12.7 kB/s	Max Uploads :	<input type="text" value="5"/> Max Down kB/s: <input type="text" value="0"/>
Seeds :	21 connected (26 in swai	Peers :	30 connected (32 in sv	Total Speed :	327.7 kB/s

Info

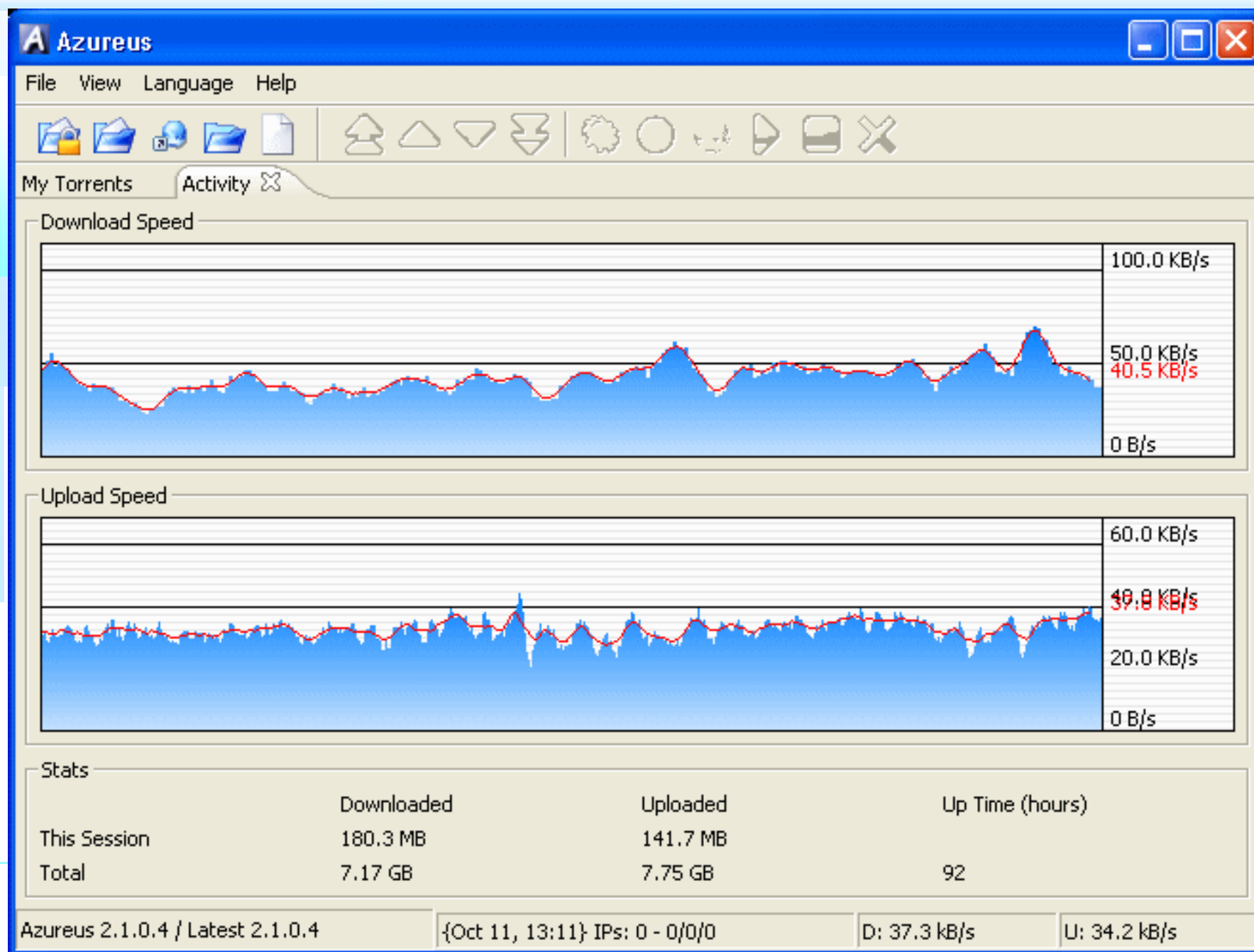
File Name :		Total Size :	1.70 GB
Save In :	D:\Downloads\	Hash :	
# of Pieces :	1739	Size :	1.0 MB
Tracker URL :		Created On :	08-Oct-2004 18:30:58
Tracker Status:	ok		
Update in :	8:39	<input type="button" value="Update Tracker"/>	
Comment :			

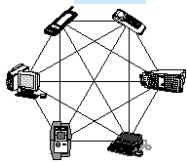
Azureus 2.1.0.4 / Latest 2.1.0.4 {Oct 11, 13:11} IPs: 0 - 0/0/0 D: 53.3 kB/s U: 34.9 kB/s





# Azureus: Andamento del download/upload





# Azureus: Mappa delle parti

**Azureus**

File Trasferimenti Visualizza Strumenti Plugins Aiuto

Miei Torrents Mio tracker 13,6% : x1... ☒

Generale Seeds/Fonti Swarm Parti Files Impostazioni Console

Ip	Client	T	Parti Disponibili	%	Velocità...	Velocità...	Stato
151.32.188.6	BitComet 0.70	R		47,1%	7,1 kB/s	17,2 kB/s	Piename...
80.116.146.218	BitComet 0.70	L		13,3%	2,0 kB/s	4,1 kB/s	Piename...
200.141.184.74	µTorrent 1.6.0	R		60,0%	1,1 kB/s	0 B/s	Piename...

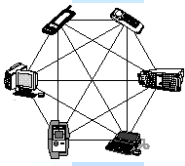
Mappa Delle Parti Console

µTorrent 1.6.0; 200.141.184.74; 60,0%

● Entrambi Lo Possedete ● Peer Possiede; Tu No ● Lo Possiedi; Il Peer No ● Nessuno Lo Possiede ● In Traferimento ● Prossima Richiesta ● Disponibilità

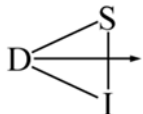
Azureus 2.4.0.2

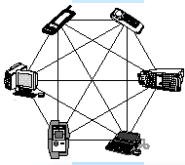
Ratio NAT OK 740.092 Utenti {lug 26, 14:21} IPs: 0 - 0/0/0 24,1 kB/s [40K] 39,3 kB/s



# BitTorrent Terminology

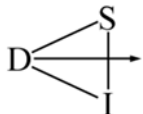
- **Choked**: a peer to whom the client refuses to send file pieces.
- **interested** a downloader who wishes to obtain pieces of a file the client has.
- **leech** a peer who has a negative effect on the swarm by having a very poor share ratio - in other words, downloading much more than they upload.
- **peer** one instance of a BitTorrent client running on a computer on the Internet to which other clients connect and transfer data.
- **seeder** a peer that has a complete copy of the torrent and still offers it for upload.
- Nodes of the BT are the peers and seeders



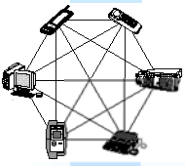


# BitTorrent Terminology

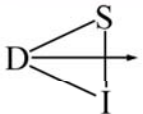
- **superseed** When a file is new, much time can be wasted because the seeding client might send the same file piece to many different peers, other pieces have not yet been downloaded at all.
  - ♣ Some clients, like ABC, Azureus, BitTornado, TorrentStorm, and  $\mu$ Torrent have a "superseed" mode,
  - ♣ they try to only send out pieces that have never been sent out before, making the initial propagation of the file much faster.
- **swarm** all peers (including seeders) sharing a torrent are called a swarm.
  - ♣ For example, six ordinary peers and two seeders make a swarm of eight.

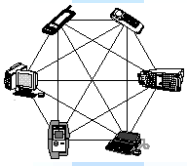






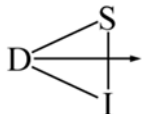
- Reti P2P
- Overlay
- DHT

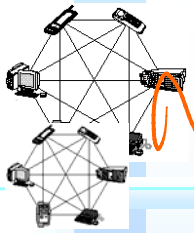




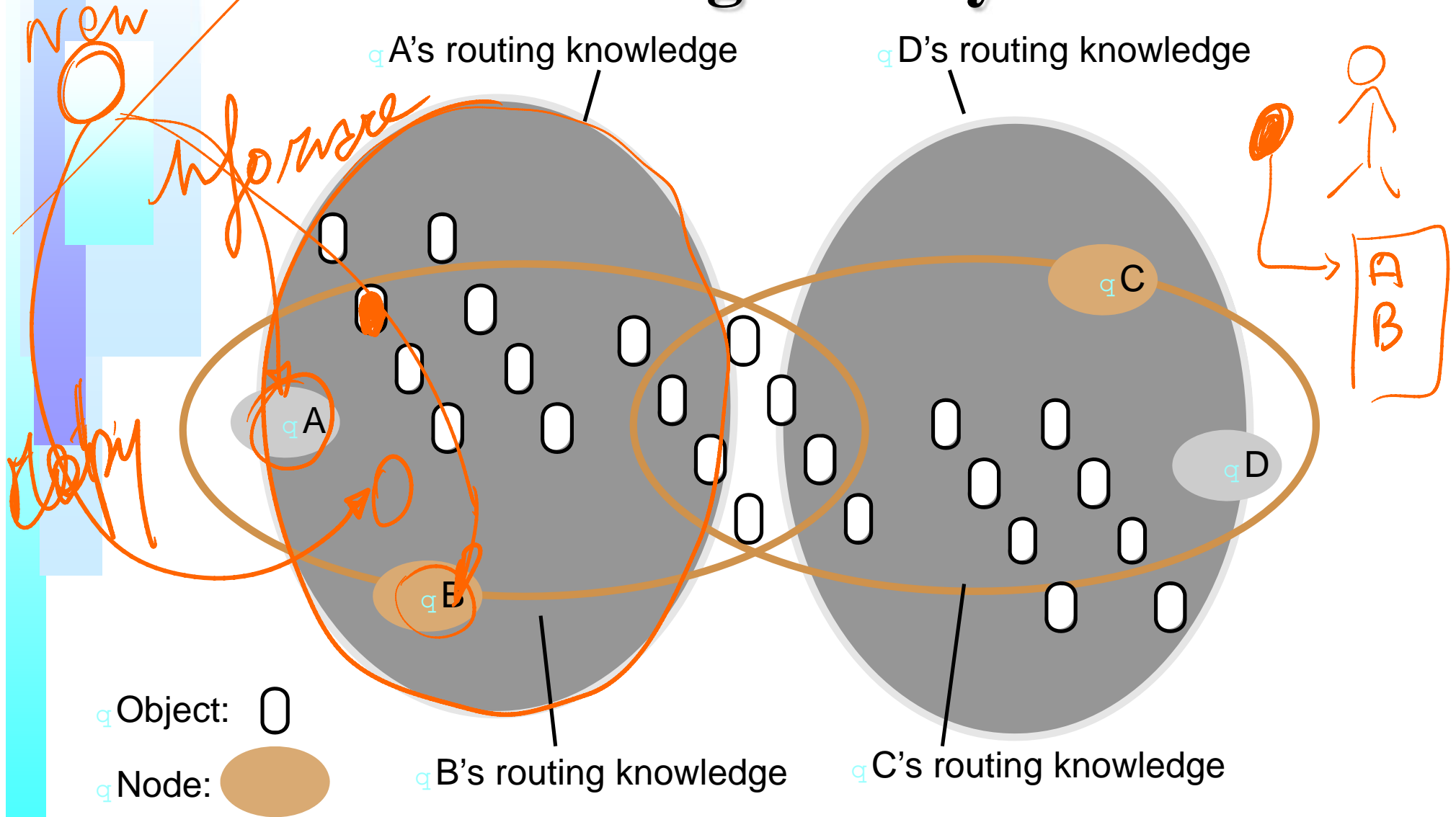
# Rounting Overlay

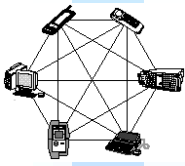
- Soddisfare tutti i requisiti precedenti e' molto complesso
- RO garantisce che ogni nodo può accedere ad ogni oggetto instradando la richiesta al fine di far raggiungere il nodo dove si trova la risorsa/info (tramite una sequenza di nodi)
  - ♣ L'oggetto può essere spostato in altri nodi senza coinvolgimento degli utenti
  - ♣ Si crea una catena di riferimenti
  - ♣ Usato in molti casi, vedasi: Skype, P2PTV, etc.
- Sistemi P2P usualmente replicano la risorsa
  - ♣ In questo caso, l'algoritmo di RO deve tenere conto di dove sono le repliche e può facilitare la consegna fornendo a fronte delle richieste/query il nodo più vicino





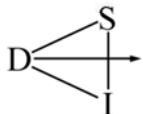
# Distribution of information in a routing overlay

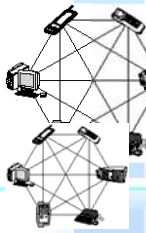




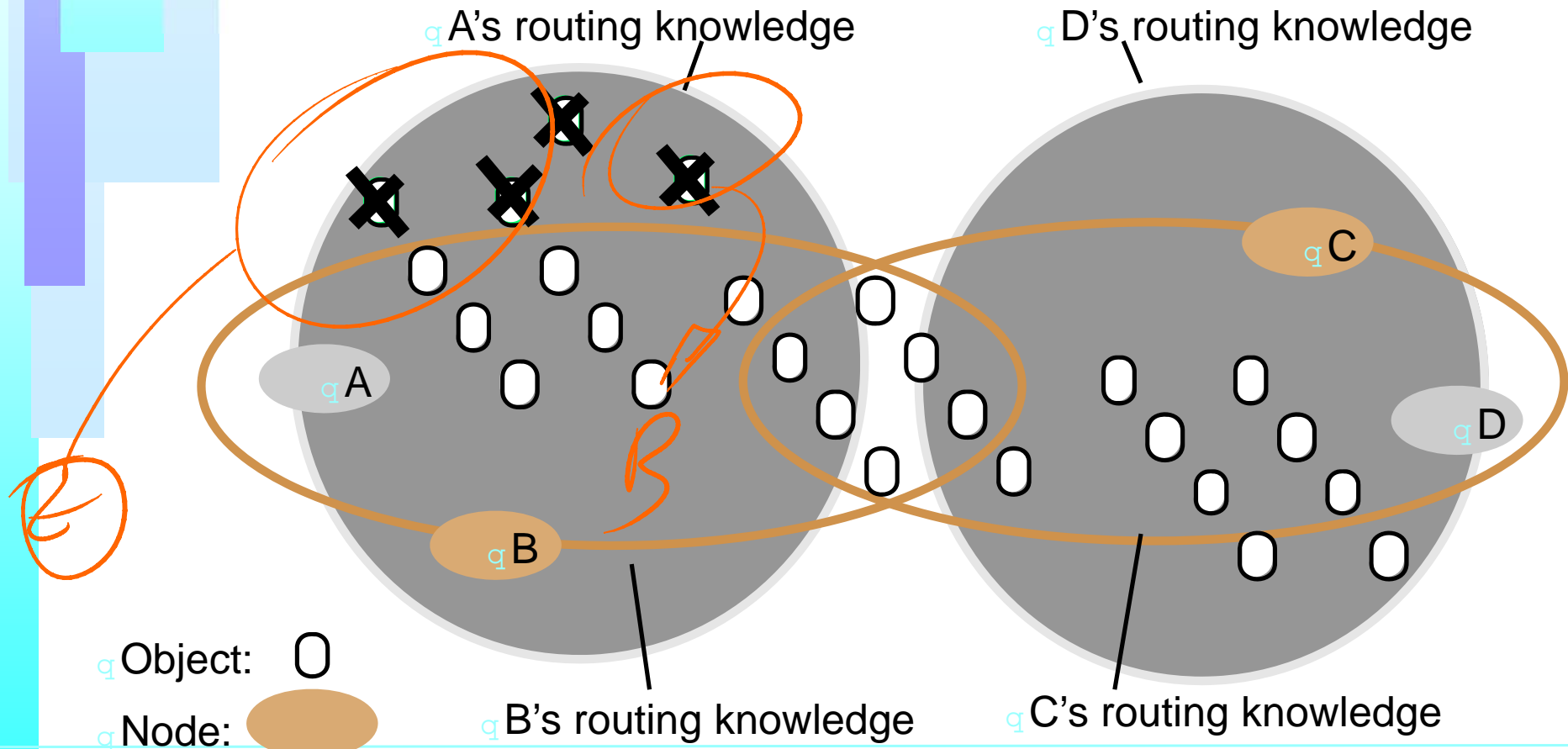
# Rounting Overlay

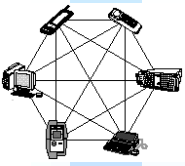
- Le richieste possono essere effettuate tramite un GUID (global unified ID)
  - ♣ La richiesta fatta al RO produce in risposta il/un nodo che ha la risorsa/info.
- L'algoritmo di RO deve anche:
  - ♣ Pubblicare/Rendere-noto a tutti i nodi le eventuali nuove pubblicazioni di GUID
  - ♣ **PRO:** Poter cancellare da tutti i nodi gli oggetti e pertanto la loro GUID che e' stata rimossa
  - ♣ **PRO:** Rendere aggiornati i nuovi nodi con la lista dei GUID dandogli alcune delle responsabilita', la gestione del segmento di conoscenza che loro rappresentano
  - ♣ **CONTRO:** Al momento in cui un nodo lascia la rete deve ridistribuire le responsabilita'/(la conoscenza) ai nodi che rimangono. In modo da non creare delle falle.





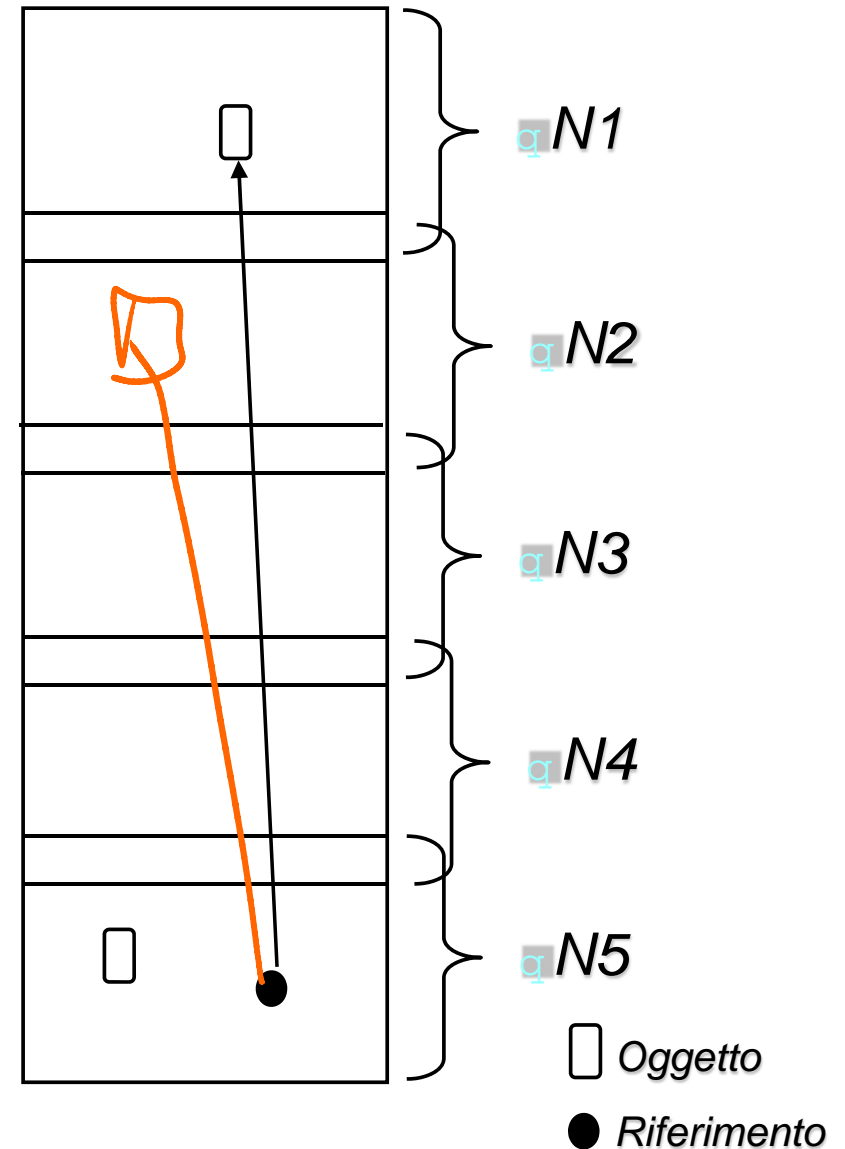
**Al momento in cui un nodo lascia deve ridistribuire le responsabilità/(la conoscenza) ai nodi che rimangono. In modo da non creare delle falle. Se A va via, gli oggetti X devono essere presi in carico da B o da altri, altrimenti vengono persi.**



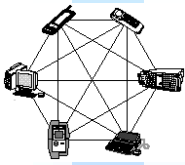


# GUID and DHT

- GUID puo' essere calcolato tramite:
  - ♣ HASH function delle info
- Pertanto il problema e' simile ad avere una tabella Hash distribuita: Distributed Hash Table, DHT.
- Si veda a destra una semplificazione

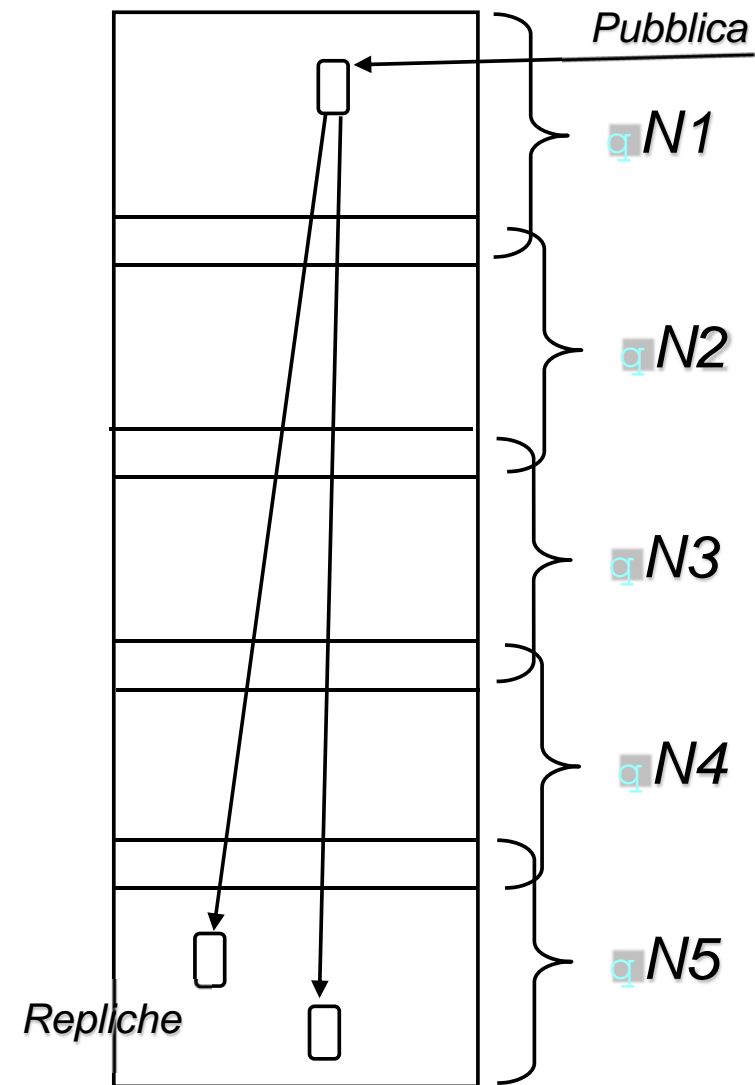


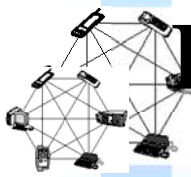




# Routing Overlay e repliche

- un algoritmo random decide dove mettere l'oggetto e le sue repliche in modo da assicurare la loro accessibilita'
- Il numero di repliche puo' essere variabile. Se l'alg. Random identifica ancora lo stesso nodo deve essere ricalcolata un nuova posizione
- La posizione dipende dal valore di GUID dell'oggetto
- Un oggetto con GUID x (e.g., 5) viene posto in nodi che hanno GUID prossimi/vicini in modo da massimizzare la probabilita' di trovarlo in fase di ricerca.





# Basic programming interface for a distributed hash table (DHT) as implemented by the PAST API over Pastry

*put(GUID, data)*

The *data* is stored in replicas at all nodes responsible for the object identified by *GUID*.

*remove(GUID)*

Deletes all references to *GUID* and the associated data.

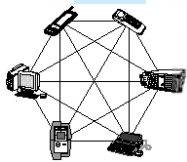
Solo accedendo a quelli che coprono tale conoscenza.

Pertanto se vi sono delle repliche prodotte da utenti, possono essere o meno cancellate se non si operano particolari accorgimenti. Comunque non sono piu' recuperabili da altre operazioni di GET pertanto la rete non le considera piu'

*value = get(GUID)*

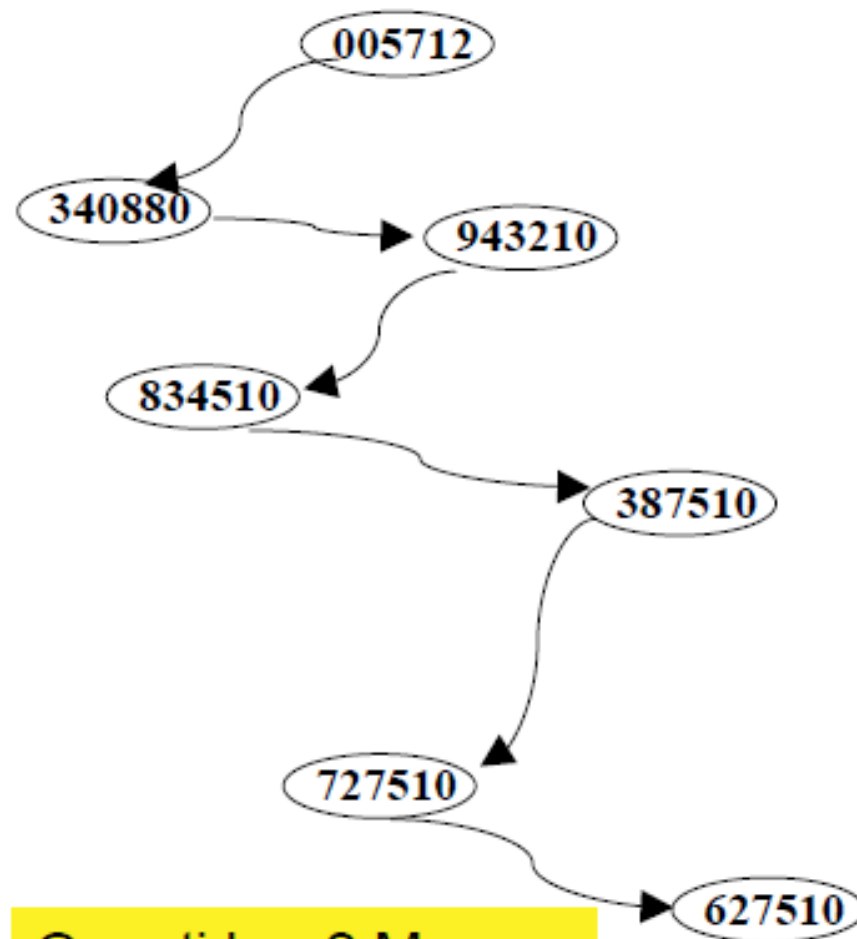
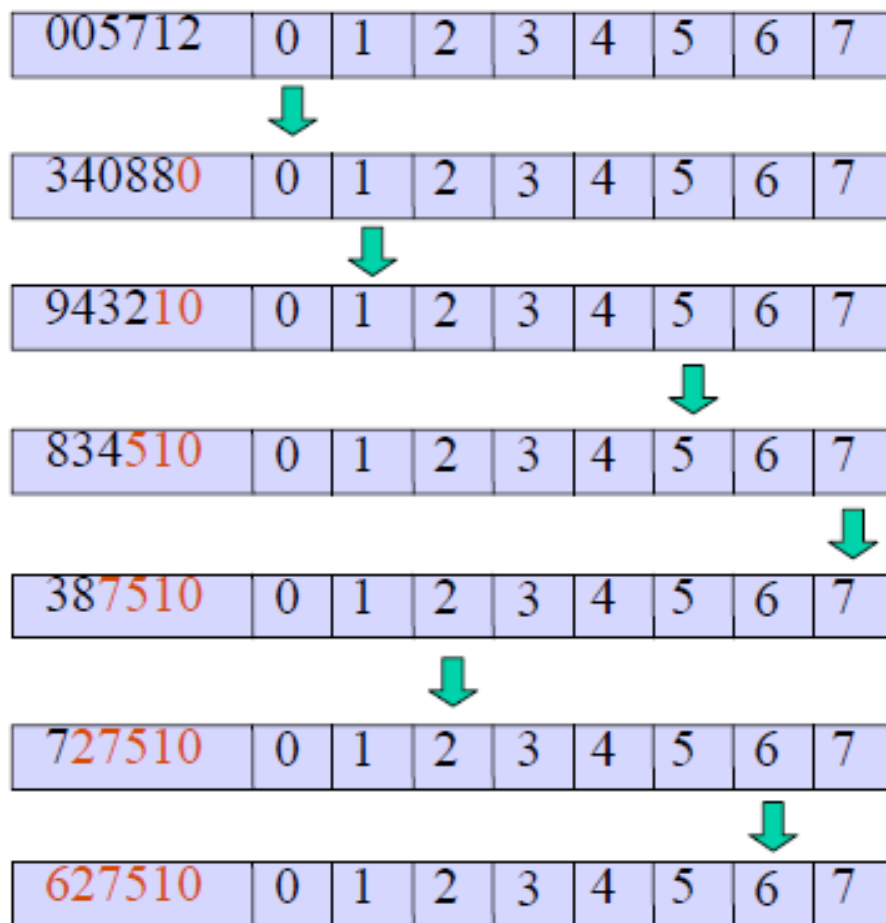
The data associated with *GUID* is retrieved from one of the nodes responsible for it. Quelli che coprono quella conoscenza



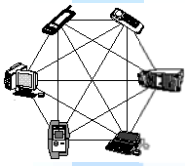


# Pastry (esempio preso da Cardellini) (3 bit per ogni cifra)

Consideriamo un namespace di  $2^{18}$ , **005712**  $\rightarrow$  **627510**



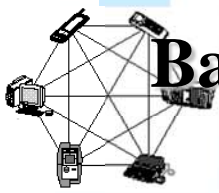
Quanti hop? Meno di  $\lceil \log_2^b N \rceil$



# Distributed Object Location and Routing

- **DOLR:** e' un modello di RO leggermente migliorato
- **Idea di base:**
  - ♣ Gli oggetti sono disposti dove si vuole, sono i riferimenti che vengono disposti sulla base del GUID
  - ♣ DOLR ha il compito di definire un mapping fra gli indirizzi dei nodi che contengono repliche e gli oggetti (con i loro GUID)
  - ♣ Gli oggetti sono memorizzati con lo stesso GUID in nodi diversi, questi sono repliche
  - ♣ RO ha la responsabilità di instradare le richieste verso il nodo più vicino al richiedente
- **Posizione degli oggetti:**
  - ♣ Le repliche sono poste senza considerare la vicinanza del valore di GUID secondo delle politiche: e.g., random
  - ♣ Ogni replica deve essere notificata al DOLR tramite Publish()



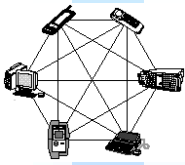


# Basic programming interface for distributed object location and routing (DOLR) as implemented by Tapestry

- ***publish*(GUID)**
  - *GUID* can be computed from the object (or some part of it, e.g. its name). This function makes the node performing a *publish* operation the host for the object corresponding to *GUID*.
- ***unpublish*(GUID)**
  - Makes the object corresponding to *GUID* inaccessible.
- ***sendToObj*(msg, GUID, [n])**
  - Following the object-oriented paradigm, an invocation message is sent to an object in order to access it. This might be a request to open a TCP connection for data transfer or to return a message containing all or part of the object's state. The final optional parameter [n], if present, requests the delivery of the same message to n replicas of the object.

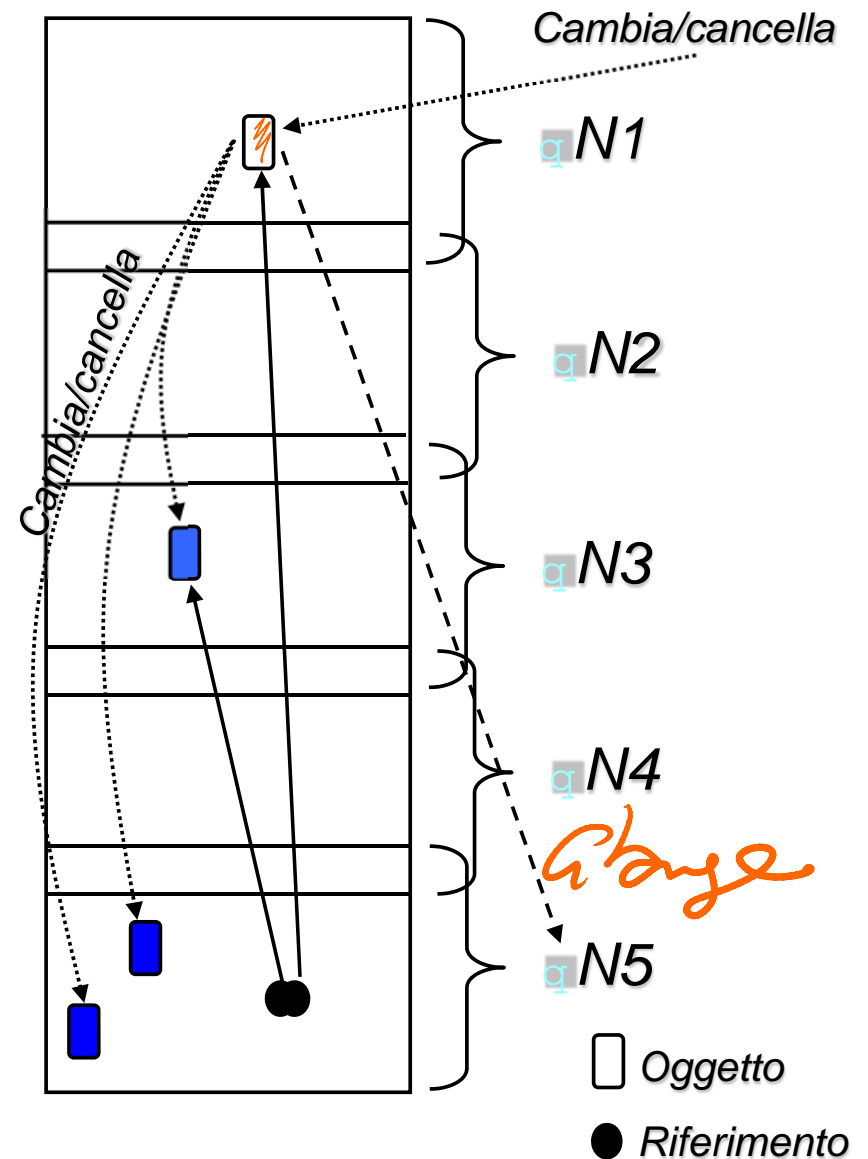




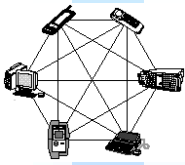


# Cambiamenti sui file

- Se un certo oggetto e' stato cambiato/cancellato
  - ♣ Notifcation to who:
    - ➔ has performed the download for that object in the past or
    - ➔ is managing replica for that object
    - ➔ has references for that object
  - ♣ The object has to be reloaded, replicated again, substituting the old one, not very nice privacy problems
- If the object is
  - ♣ not replicated the change is immediate
    - ➔ Who has downloaded has to be informed as well
  - ♣ replicated:
    - ➔ Make a query to know where the object is replicated
    - ➔ removing/deleting the old versions
    - ➔ Put/publish the new one

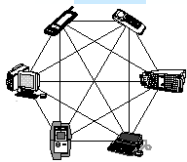






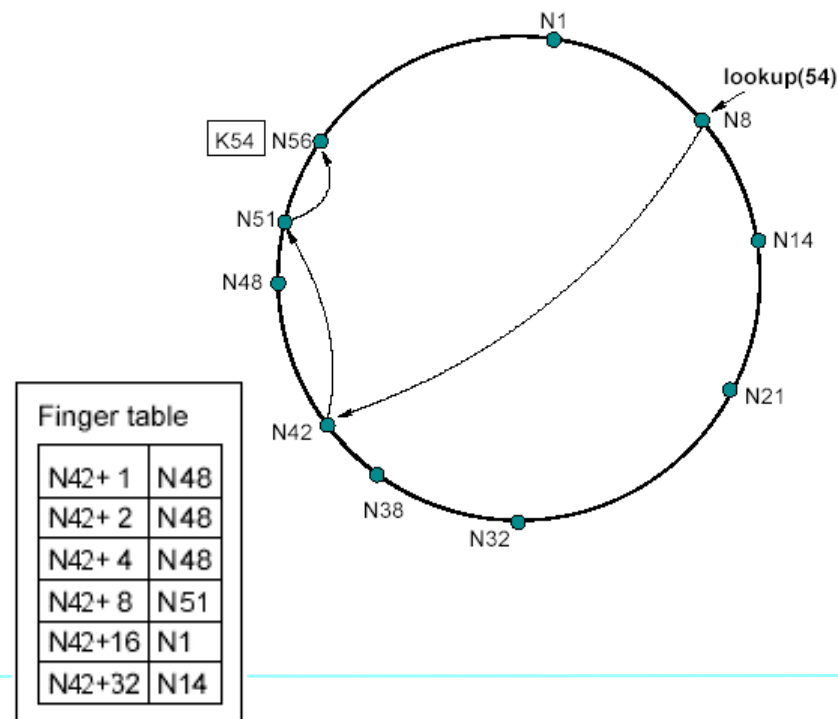
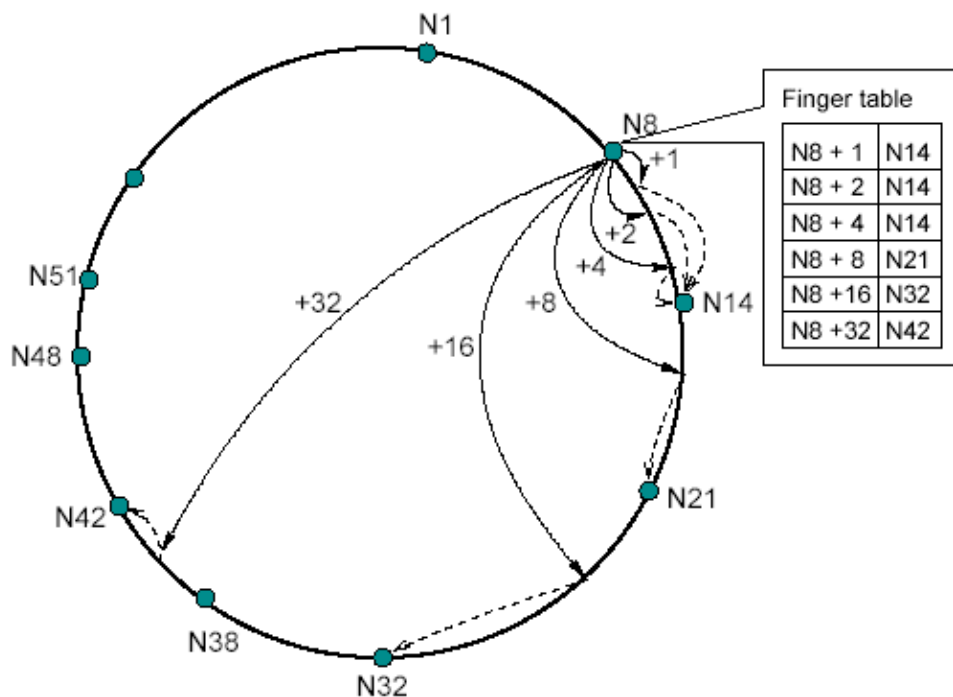
# Come instradare

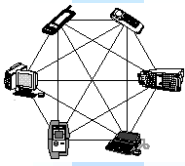
- **Pastry e Tapestry** usano il Prefix Routing per determinare il percorso per l'instradamento per la consegna dei messaggi/pacchetti/richieste indirizzate ad un certo GUID.
- Idea di base:
  - ♣ Modelli basati sulla disposizione dei nodi in base ad una gerarchia come per esempio in routing IP packets:
    - ➔ ogni byte identifica 256 possibili figli, ogni figlio 256 figli, etc.. IP4 has 4 livelli.
    - ➔ Segmento il GUID in livelli, etc.
  - ♣ Praticamente dei meccanismi che permettono di definire delle distanze massime fra nodi



# Criteri per la stima della distanza

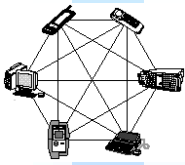
- **CHORD** come distanza usa la differenza fra il GUID del nodo presente e di quello che si cerca.
- ♣ Distanza in un modello Hash uniforme
- ♣ Nodi geograficamente distanti potrebbero trovarsi vicini nello spazio della tabella, questo non e' positivo per ottimizzare i tempi di comunicazione visto che nodi vicini si devono parlare spesso
- ♣ Si basa su un match esatto della stringa di ricerca





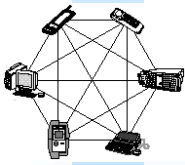
# Criteri per la stima della distanza

- **CAN:** usa una distanza  $d$ -dimensionale nell'iperspazio delle possibili posizioni dei nodi
  - ♣ Dimensioni per esempio possono essere: IP (geografico), location (nationality), language, fuso orario, codice postale, hash, etc.
- **Kademlia:** usa lo XOR sulla coppia di GUID come distanza fra i nodi
  - ♣ Anche questo puo' avere i problemi che si hanno per CHORD.



# GUID storing

- GUID non hanno senso per gli umani sono semplici codici binari/esadecimali/ottali “lunghi”, non hanno un significato diretto alla loro lettura
- GUID puo' essere calcolato sulla base dei dati che compongono l'oggetto e/o le informazioni del nodo, per esempio sulla base delle keyword che lo descrivono, il file name, etc.
- potrebbero essere anche ottenuti tramite lo standard UUID che pero' produce un valore assoluto non ricostruibile dai dati dell'oggetto e dovrebbe essere dato da un ente superparte, che oltre che generare l'ID verifica di non avere dei duplicati



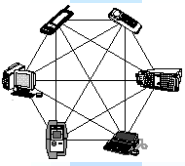
# Skype P2P architecture



- Skype is a P2P application for VOIP, Voice Over IP
- Skype has a proprietary protocol,
  - ♣ the P2P is similar to KaZaA
  - ♣ AES 256 is used to protect the channel
- There are three types of nodes in the P2P network:
  - ♣ Ordinary-peers (OP, the client), Super-peers, Central login server
  - ♣ The boot of OP is performed on a Super Peer (SP), and ask to the Central server to perform the authentication
- For user search
  - ♣ OP send the user name to SP which provide 4 IP addresses, if it is not there, with another request to SP obtain 8 peers, etc...

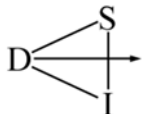




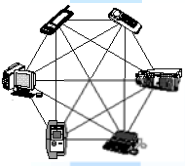


# Some info on Skype

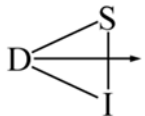
- Skype usa il P2P per implementare la directory distribuita degli utenti
  - ♣ DHT – Chord algorithm
  - ♣ Costo della ricerca:  $O(\log N)$
- For user search
  - ♣ OP send the user name to SP which provide 4 IP addresses, if it is not there, with another request to Sp obtain 8 peers, etc...

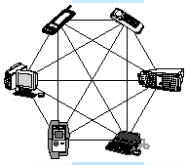






# Controllo e supervisione reti P2P

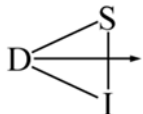


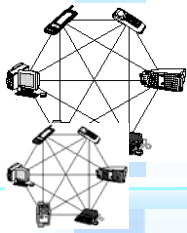


# GUID e indirizzi dei peer

## ○ Indirizzi logici e fisici dei peer

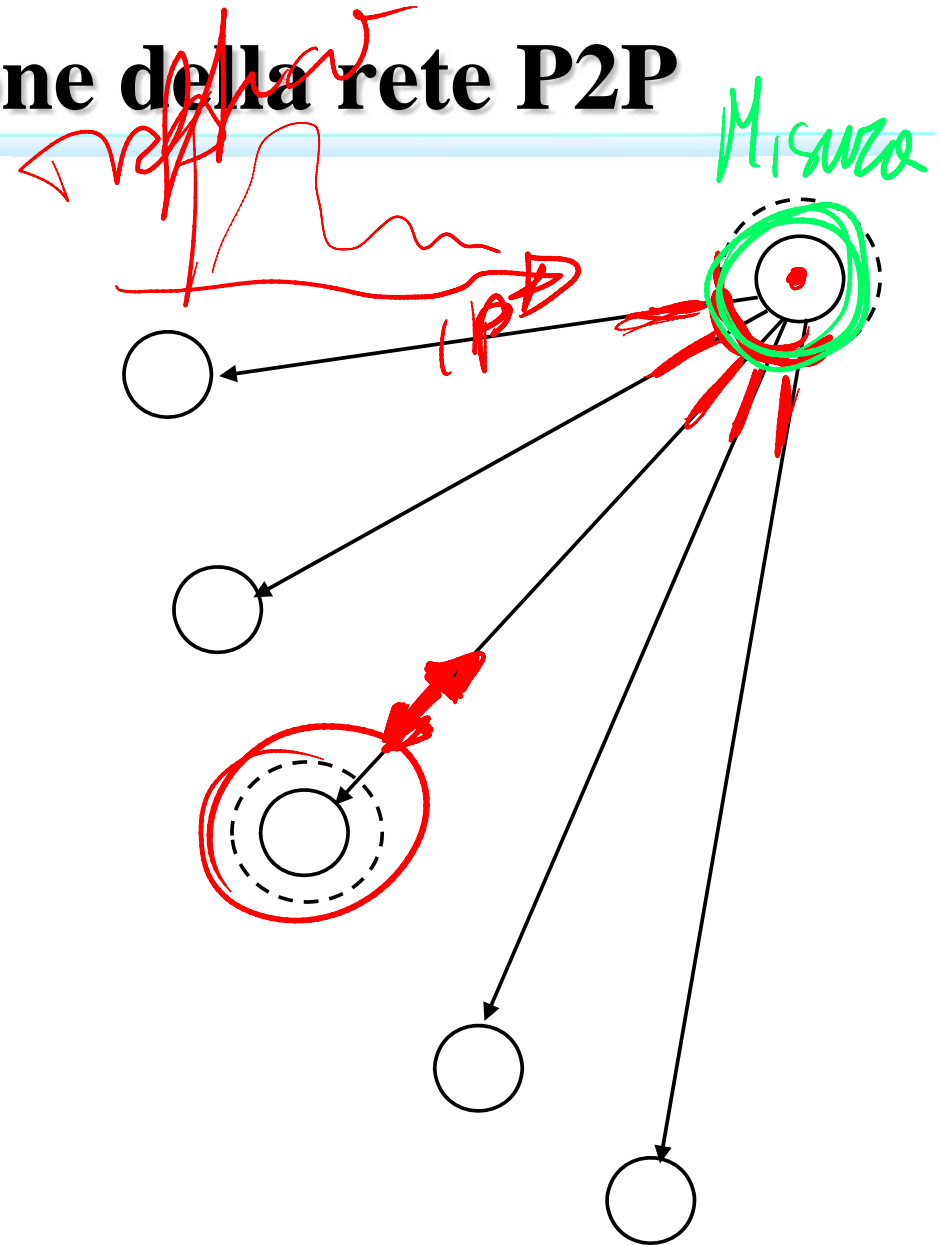
- ♣ Internet Service Provider danno in modo dinamico degli indirizzi su base DHCP, sempre in un certo range, ma diversi
  - ➔ perdita di validità dei link, dei riferimenti ai file, etc...
  - ➔ L'ID del nodo dovrebbe essere effettuato su una base diversa.
- ♣ I Firewall di struttura offrono verso l'esterno un unico indirizzo per tutti i nodi che ci stanno dietro
  - ➔ Si possono usare protocolli che espongono anche l'indirizzo reale interno del nodo nella intranet, ma devono essere tenuti tutti e due, tutte le intranet usano lo stesso range.
- ♣ Intranet e DHCP (ISP o firewall)
  - ➔ Possono dare indirizzi a rotazione periodica (alcune universita')
  - ➔ altre hanno la reservation su base del MAC address pertanto vegono sempre assegnati gli stessi con elevata prob.
- ♣ Il MAC address sarebbe un migliore identificativo ma molti calcolatori ne hanno diversi...diverse schede di rete. In alternativa, un fingerprint del calcolatore potrebbe aiutare a fare del riconoscimento

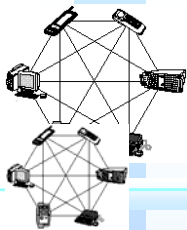




# Controllo e supervisione della rete P2P

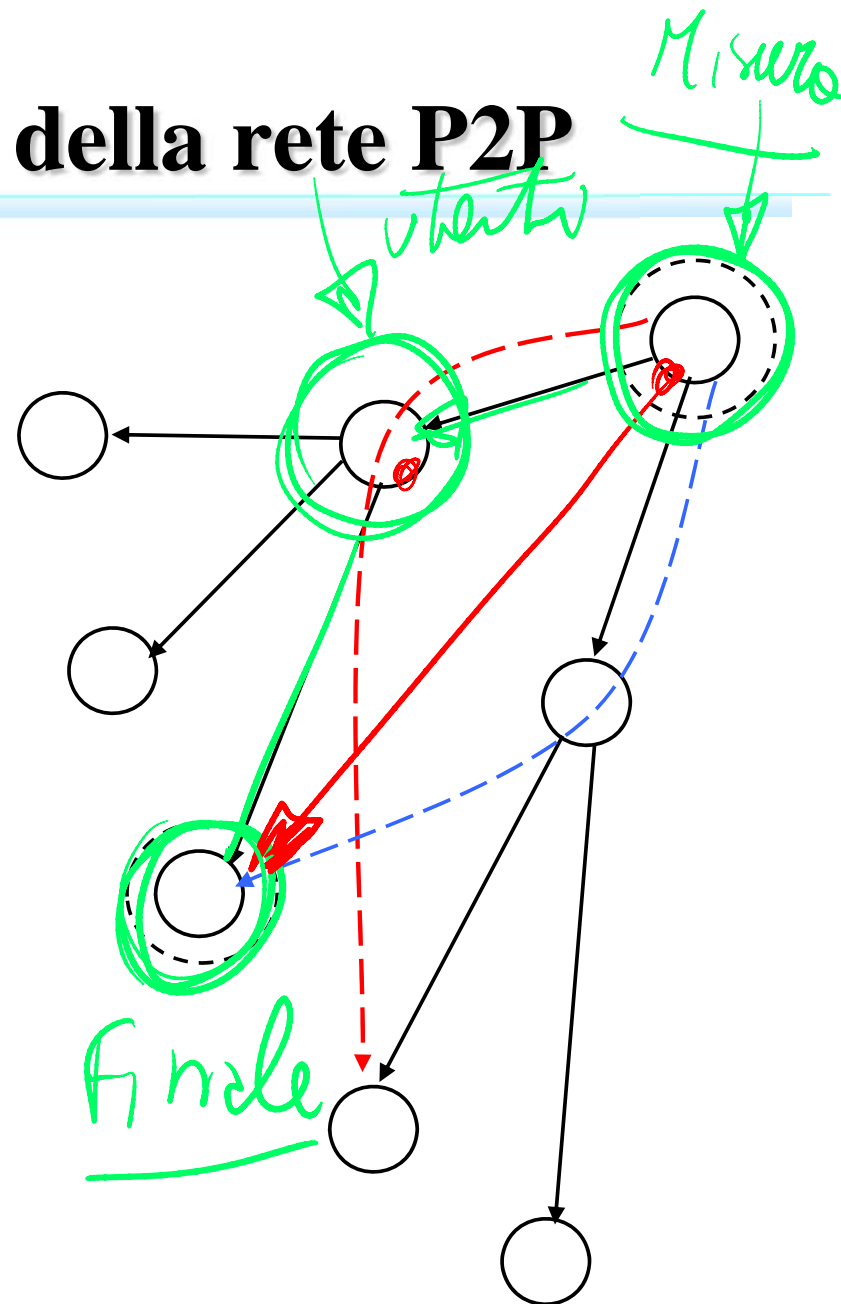
- Sulla rete passano molte informazioni
  - ♣ Alcune sono sensibili per IPR
  - ♣ Altre potrebbero esserlo per la sicurezza nazionale, per il penale
- Eventuali monitoraggi, con sniffer
  - ♣ Controllo intorno al nodo che riceve
  - ♣ Controllo intorno al nodo che trasmette
  - ♣ Controllo sul provider e leve legali

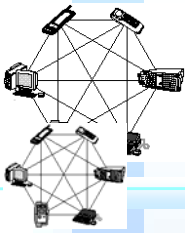




# Controllo e supervisione della rete P2P

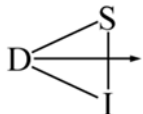
- Per evitare di essere troppo visibili come nodo provider
- Soluzione di instradamento:
  - ♣ Instradamento del file tramite nodi intermedi terzi
  - ♣ Se i nodi intermedi non fanno caching, il controllo intorno al nodo che trasmette e' ancora possibile visto che il traffico esce comunque da quello
  - ♣ Instradamento di uno stesso file tramite nodi diversi

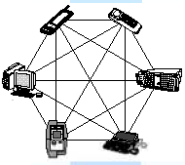




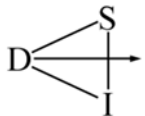
# Controllo e supervisione della rete P2P

- Per evitare di poter essere controllati sulla rete
  - ♣ Soluzioni che utilizzano canali protetti, proteggono solo il traffico ma non il monitoraggio dei volumi SSL
  - ♣ Soluzioni che utilizzano la rete P2P come un database virtuale
    - ➔ Divisione del file in segmenti spread sui nodi, anche in forma criptata....
    - ➔ Segmenti dello stesso file finiscono in nodi diversi anche lontani
    - ➔ Perdita di controllo della porzione dell'HD, l'utente non conosce cosa contiene, problemi di sicurezza visto che potrebbe essere sfruttato per azioni non legali
    - ➔ Controlli non facilmente realizzabili

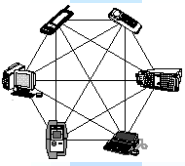




# ● Esempi di soluzioni

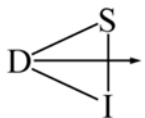


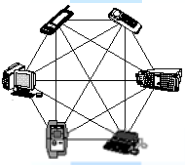




# Vediamo alcuni esempi

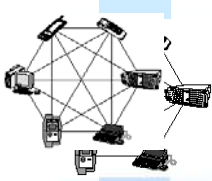
- **Bit Torrent**
- **P2P AXMEDIS del DISIT**, basato su BitTorrent, palestra per le valutazioni e nuovi modelli.
- **Monitoraggio di sistemi P2P**
- **P2P per video streaming**
- **P2P per distributed trust**





# BitTorrent

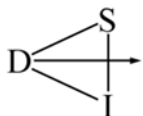
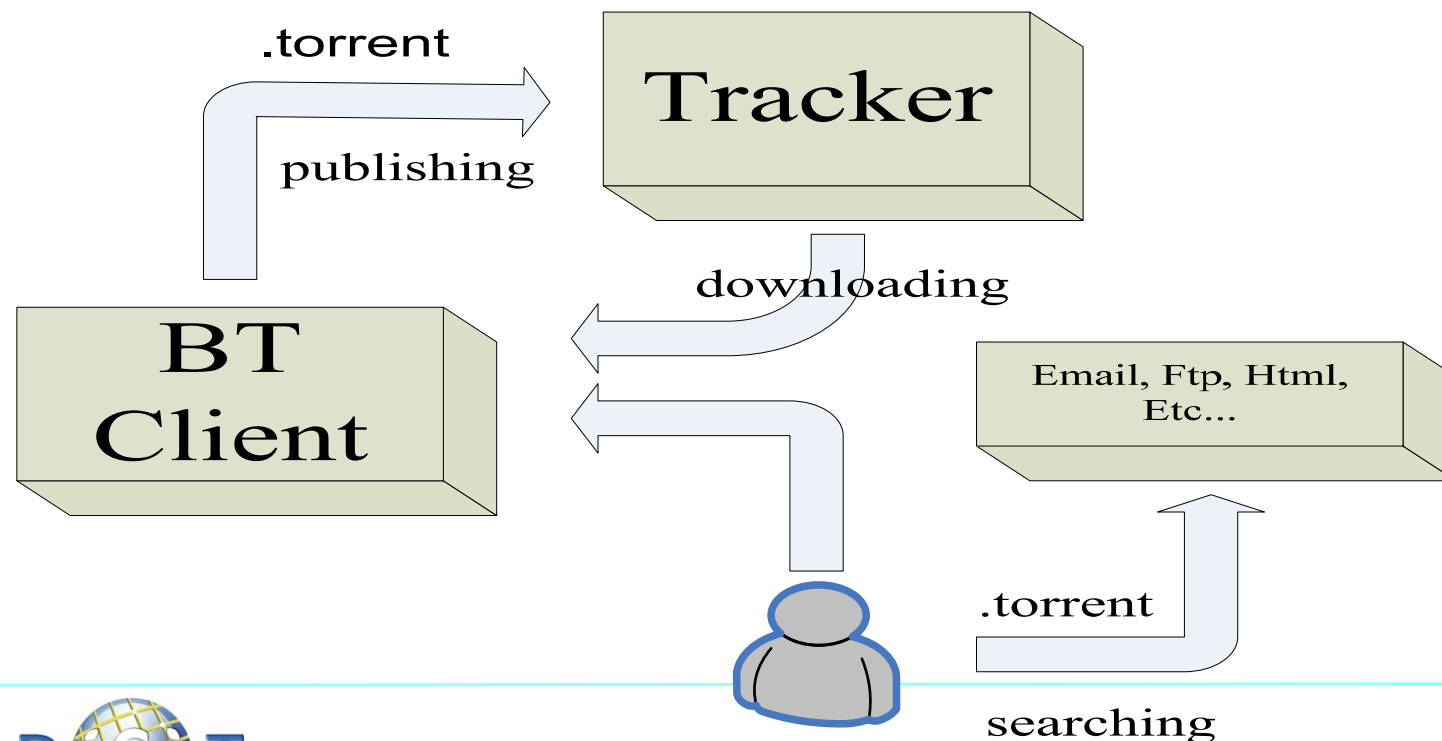
- Questa e' la seconda parte, la prima parte e' stata presentata in precedenza nel contesto del download multisorgente

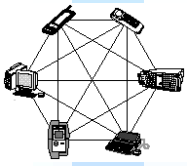


# BitTorrent based solutions

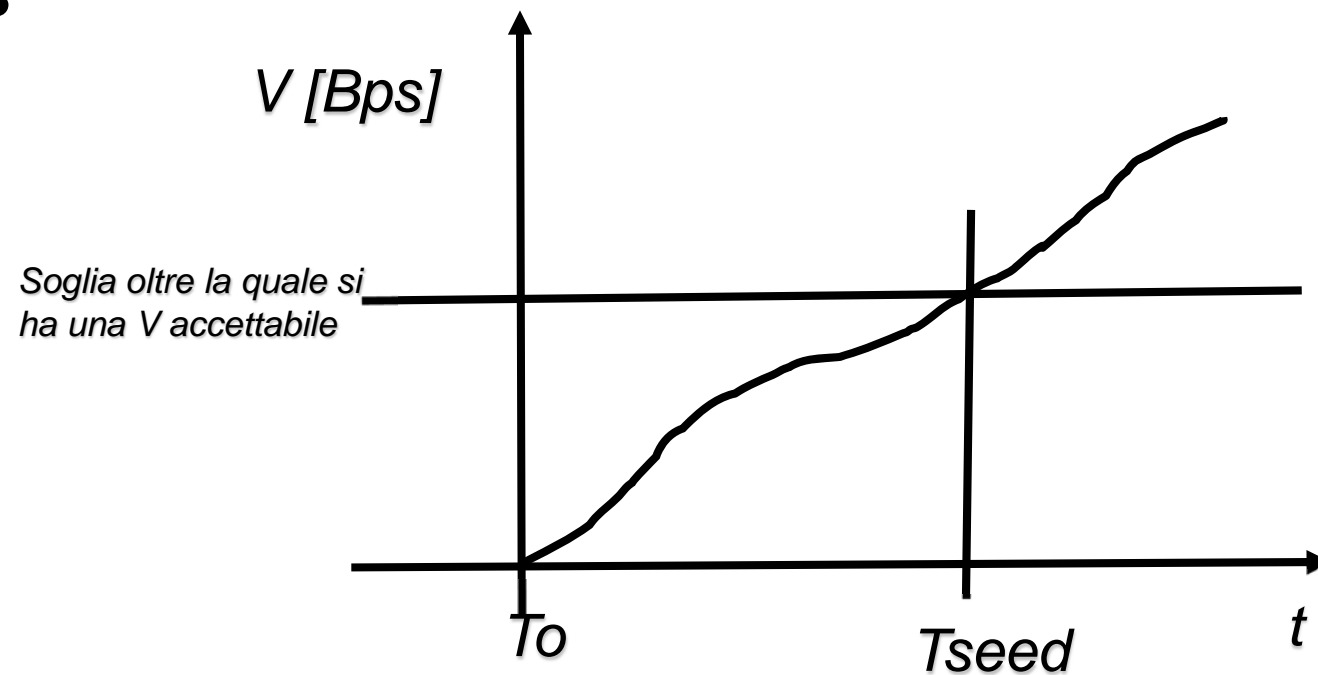
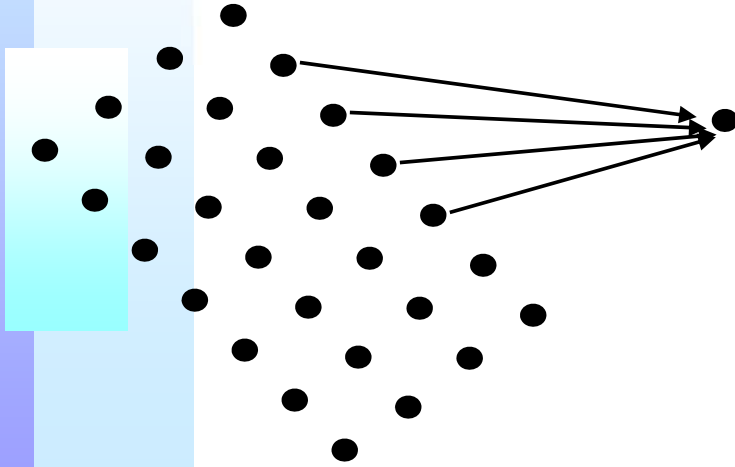


- When a file is published by a peer (initial seeding) a file (.torrent) is created and sent to a reference Tracker
- A peer can start the download having the BitTorrent file (can be obtained: from the tracker knowing the ID, via email, from HTML pages, ftp, MMS, etc. )
- The Tracker periodically updates the list of peers involved in hosting the file and notify them about the other active peers
- Tracker has the list of objects, the catalogue, and the metadata are limited, e.g., to the file name





# Time of Seeding of a given object in a given point

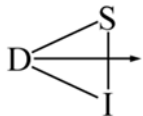


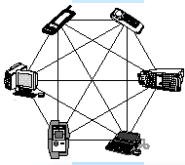


# BitTorrent limitations



- **no capabilities to provide support for**
  - ♣ **querying/indexing** of content on the basis of the metadata and related object/file cataloguing and querying for B2B and/or C2C (Consumer to Consumer).
    - ➔ The querying/indexing is delegated to external services,
    - ➔ the content type is not uniform so that the classification is hard and almost impossible
    - ➔ the Tracker has only capabilities of presenting the list of objects, the called catalogue and the metadata are limited to the file name;
  - ♣ **content protection** and DRM, to control the publication distribution and sharing of non certified/protected/authorized content



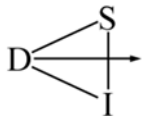


# BitTorrent limitations

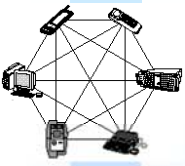


## ○ **no network control** for

- ♣ **removal** of content/files from the network, that would mean to remove them at least from the tracker;
- ♣ **fast notification of new files** (changed files) and thus of seeding of files among the network;
- ♣ **publishing and downloading** files in an automatic manner, via the integration of the P2P network facilities with the content production facilities;
- ♣ **monitoring activities** and user behavior on nodes.
  - ➔ In addition to the classical P2P network monitoring that can be performed on the tracker in a limited manner

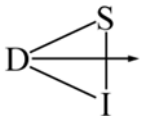


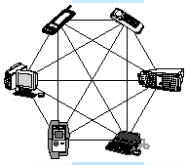




○ AXP2P

○ BitTorrent





# P2P Network of AXMEDIS



- **Requirements**

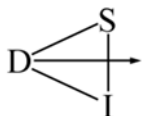
- ♣ P2P for B2B content distribution and sharing

- **BitTorrent Technical Requirements**

- Fast content download

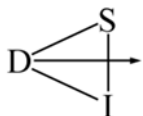
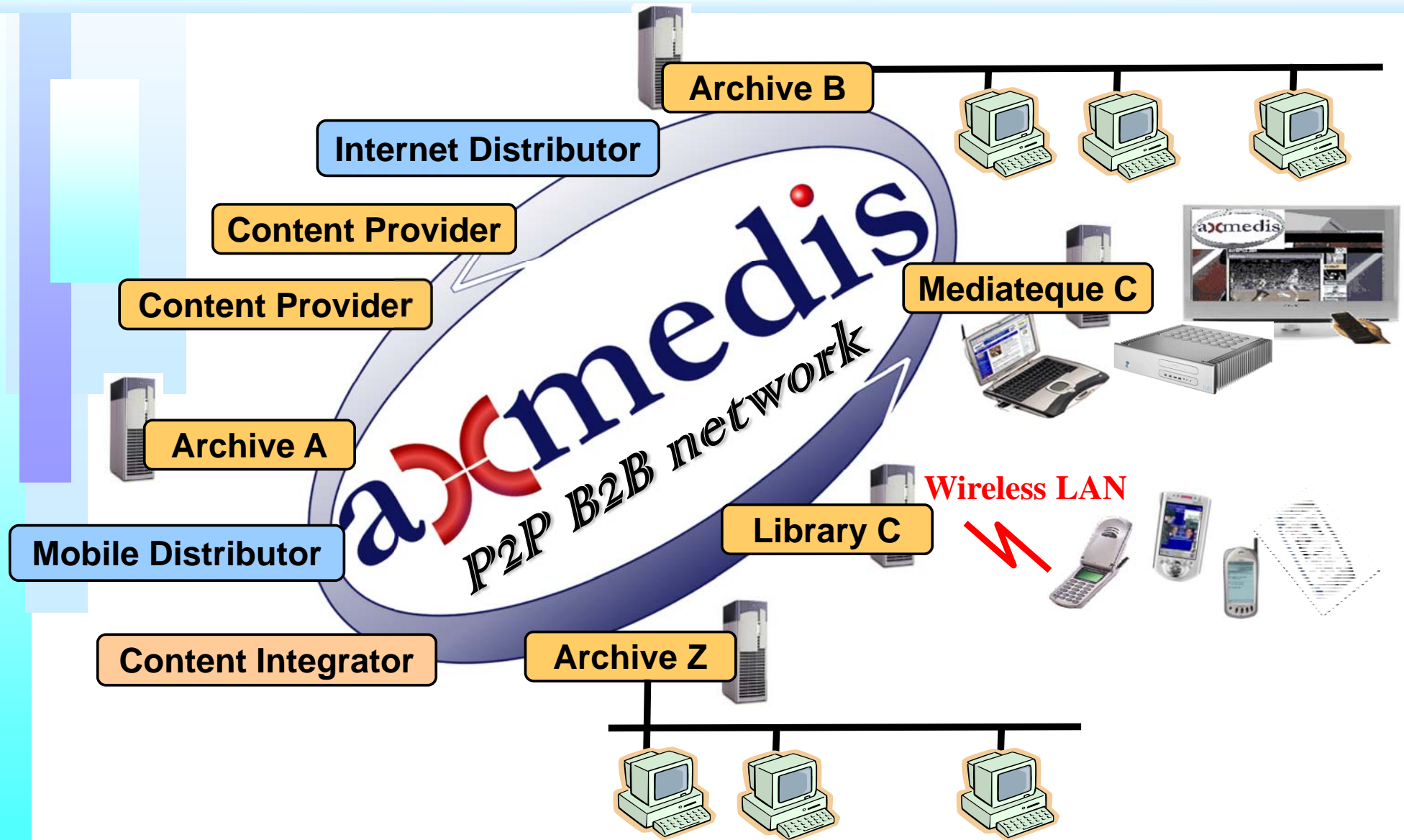
- **Additional Technical requirements:**

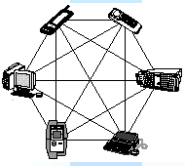
- ♣ Support to make query, similarly to other BT servers
- ♣ Immediate/fast high performances
  - Fast seeding for a number of given objects
- ♣ Control of the network
  - Delete/change the objects/content
- ♣ Certified metadata
- ♣ Secure in terms of IPR and DRM
- ♣ Automated control





# Content Sharing among Content Archives





# P2P Network of AXMEDIS, req.

## ○ Automated Control of P2P Network:

- ♣ **Sharing and publishing of content**

- ♣ **Control of the P2P network** avoiding the

  - **distribution of non certified/authorized** content (e.g., content with metadata inconsistent with resources),

  - **sharing of illegal files** (those that are shared without the corresponding authorizations of the content owner),

  - **access to P2P B2B facilities** to non authorized (or malicious) actors/users, that is registering the users;

- ♣ **monitoring the activities of the P2P network** (tools and users) in terms of

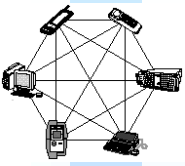
  - Performances: ?download rate for a given file in a given area,

  - content shared: when, where and by who

  - Queries on Query Server: log of performed queries

  - statistical information that may be used to better tune the service and understand the user behavior;

  - Status of the control nodes: workload, cpu, disk, etc.



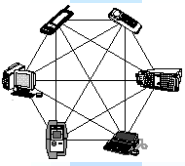
# P2P Network of AXMEDIS, req.

- **Automated Control of P2P Network:**
  - ♣ **querying on content** on the basis of a large set of metadata, including those to make search and queries on business/trading aspects: complex metadata, licensing rules and conditions, costs, etc.;
  - ♣ **set up of high quality services** of content distribution and sharing, CDN (Content Delivering Network).

This implies to guarantee the content download according to predictable performance, QOS (Quality Of Service); even when

    - ➔ a new content object/file is shared, thus when the P2P network may not contains enough replicas;
    - ➔ faults occur in the network and/or in the nodes;

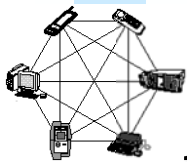




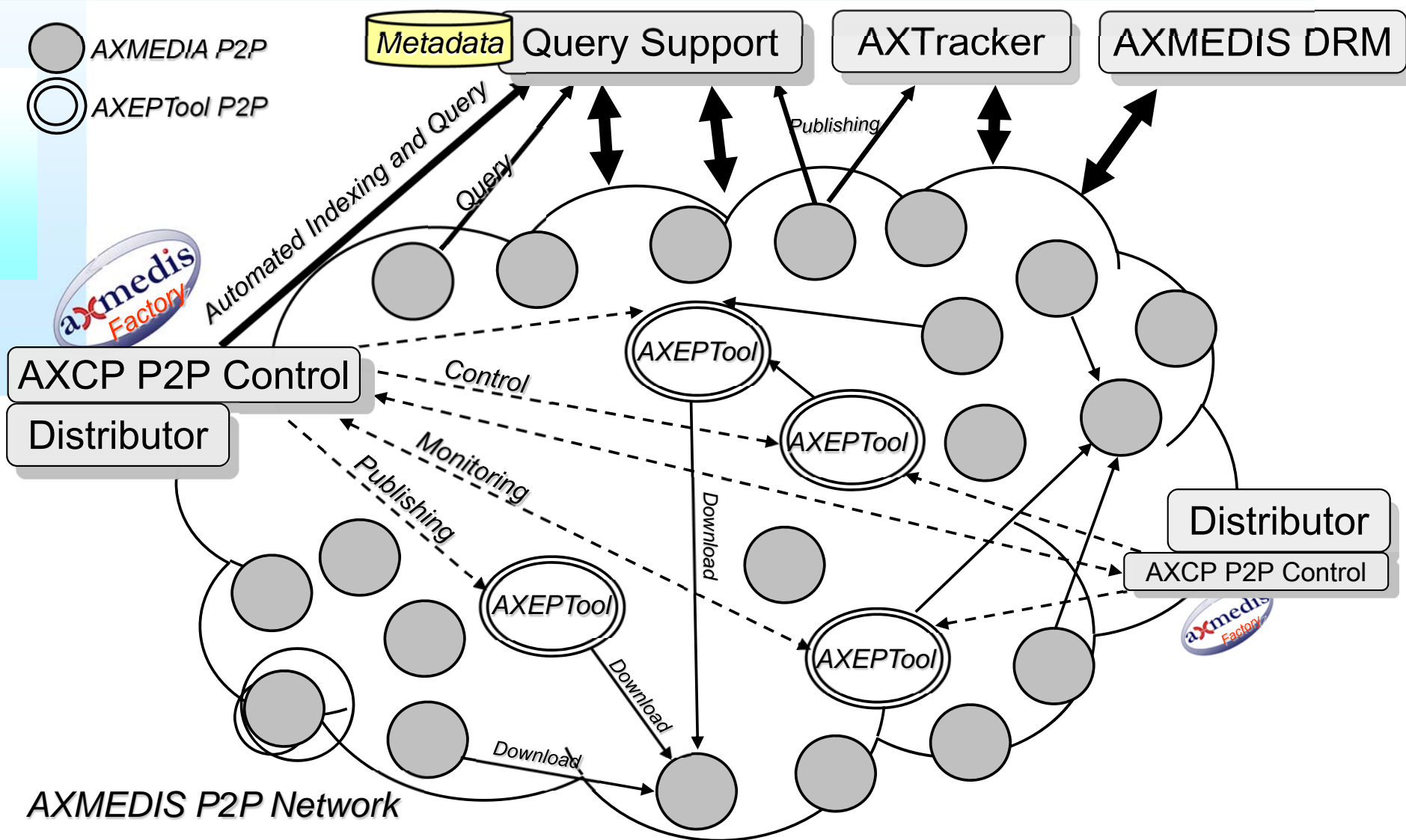
# AXMEDIS P2P solution

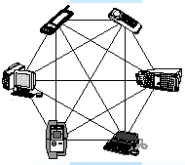
- **Satisfaction of all the above mentioned requirements for B2B P2P networks**
- **Extension of the BitTorrent solution**
  - ♣ Usage of the AXOID as unique identification
    - ➔ From AXOID to hash
  - ♣ Insertion of a classification and query support
  - ♣ Insertion of DRM support (only to protect digital files)
  - ♣ Insertion of control nodes for publication and monitoring
- **Addition of:**
  - ♣ P2P client tools for B2B: AXEPTool tool
  - ♣ P2P client tools for Consumers, B2C and C2C: AXMEDIA tool
  - ♣ Server for classification and query: AXMEDIS Query Support
  - ♣ Server for DRM: AXMEDIS DRM
  - ♣ GRID solution for P2P Network control, AXCP based





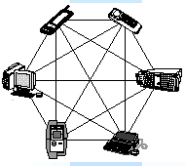
# AXMEDIS P2P network architecture





# P2P Network of AXMEDIS

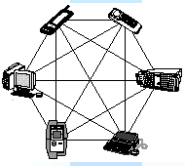
- **Tracker: AXTracker** is a modified BitTorrent Tracker that manages the AXMEDIS P2P network and community
- **Supernodes: AXEPTool** is a special P2P BitTorrent Client Node, suitable to play the role of a P2P Node for B2B activities such as producers, distributors, integrators, etc., for B2B content distribution.
- **P2P clients: AXMEDIA** is a specific P2P BitTorrent Client Node for final users content sharing and B2C (Business to Consumer) content distribution.
- **Control: AXCP GRID** is an instance of the AXCP GRID tool to control the activities of some AXEPTools
- **Query Server: AXQuery Support** is a server on which the user and the AXCP may perform queries



# AXMEDIS P2P Network

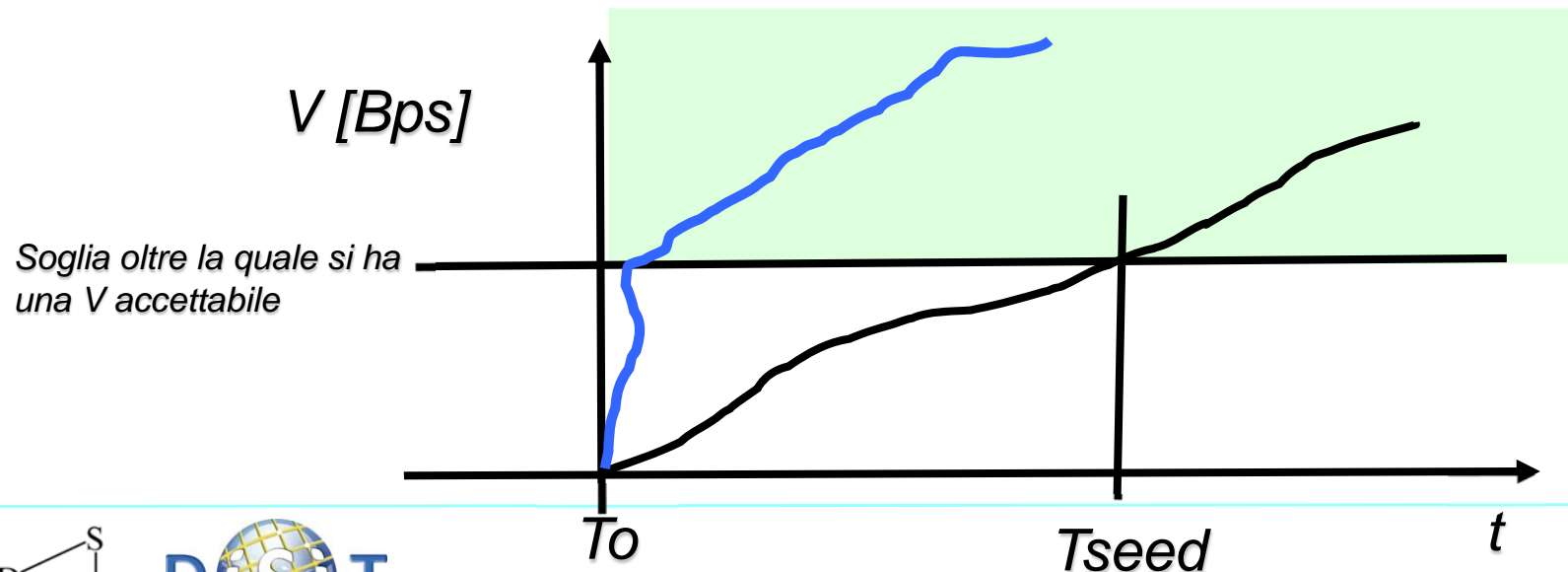
## ○ different kinds of P2P Nodes:

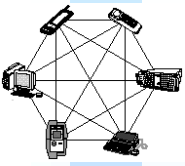
- ♣ content production, publications and sharing nodes in which a controlling tool (e.g., AXCP GRID) and at least one AXEPTool are joined;
- ♣ content sharing and distribution nodes which are constituted by an AXEPTool only (controlled and supervised by other AXCP GRID nodes);
- ♣ AXMEDIA P2P nodes for content sharing.



# AXP2P Tuning the service, req.

- **Monitoring the network** by means of the a uniform distribution of supernodes allows to measure the:
  - ♣ The velocity of download for each content for hours of the day along the time of service
  - ♣ This may be used to change the distribution of replicas on supernodes so that to work in the guaranteed area since  $T_0$

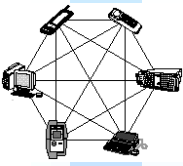




# P2P Network of AXMEDIS

- Exploitation of BitTorrent
  - ♣ Hierarchical BitTorrent solution, super-peers
- In addition technical support to:
  - ♣ MPEG-21 files and normal files
    - ➔ Certification of objects
    - ➔ Centralized query support for search of MPEG-21 files
  - ♣ Control of P2P network via one or more control servers, that may be grid or not.
  - ♣ perform measures on the Tracker
  - ♣ control the P2P status





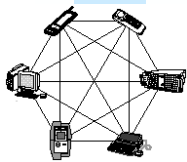
# AXMEDIS P2P network: Benefits



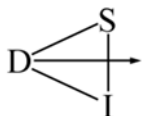
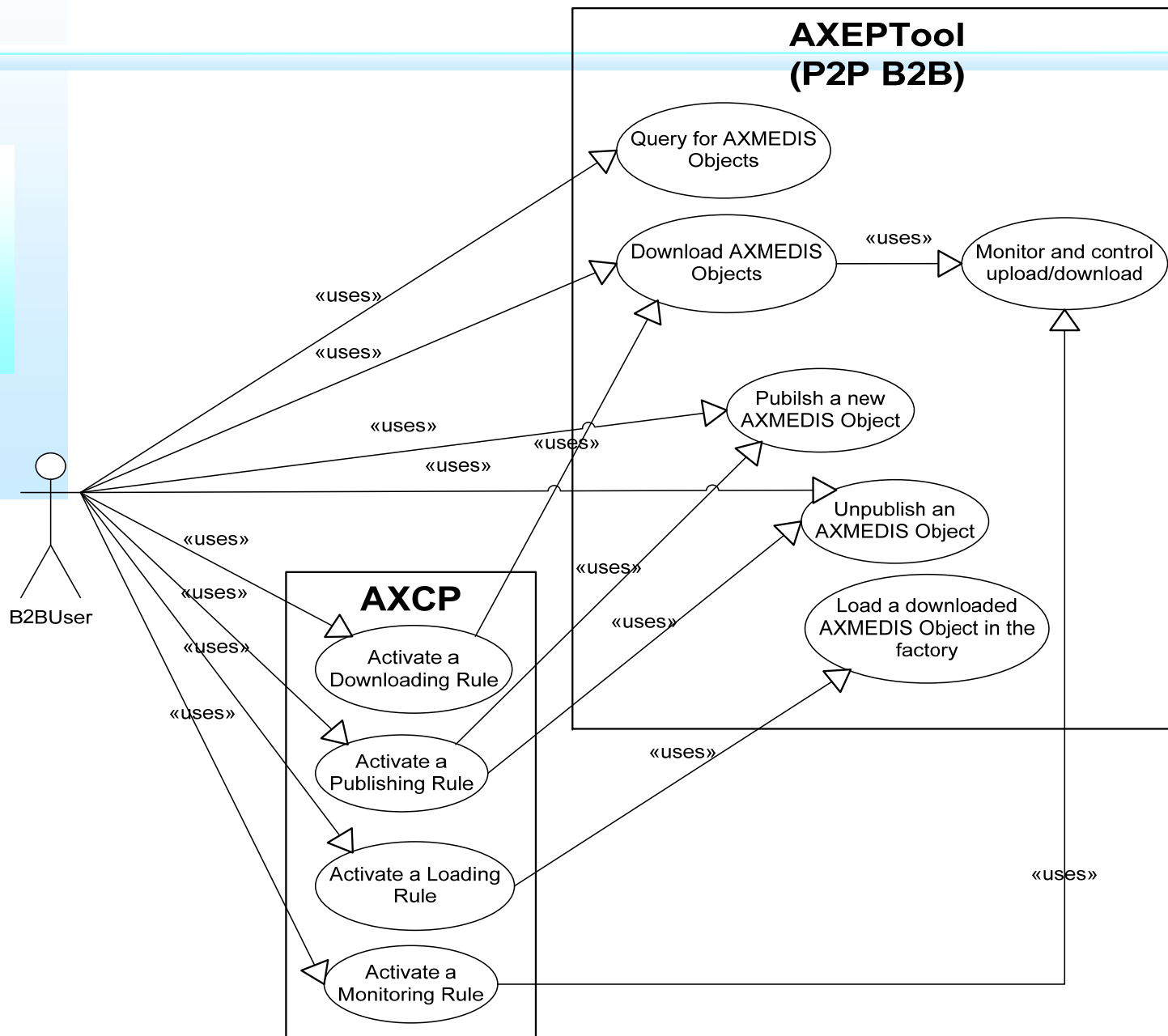
- Fast seeding of the P2P network
  - ♣ Asking to superpeers to start downloading of new objects
  - ♣ Immediate notification of new objects
- Guaranteed quality of service
  - ♣ With the deterministic number of superpeers and the possibility of knowing their networking capabilities
- Possibility of deleting object from the network
  - ♣ Delete of objects on superpeers via GRID
  - ♣ Delete of objects on the Tracker (to be done)
- Possibility of having multiple GRIDs controlling the network for publication of objects
- Performance control:
  - ♣ Control of tracker performance
  - ♣ Control of superpeer performance, in terms of networking, seeding, space on disk, etc.
  - ♣ Definition of policies of LRU on the network, optimization of content location

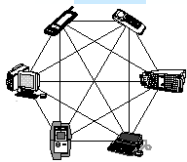




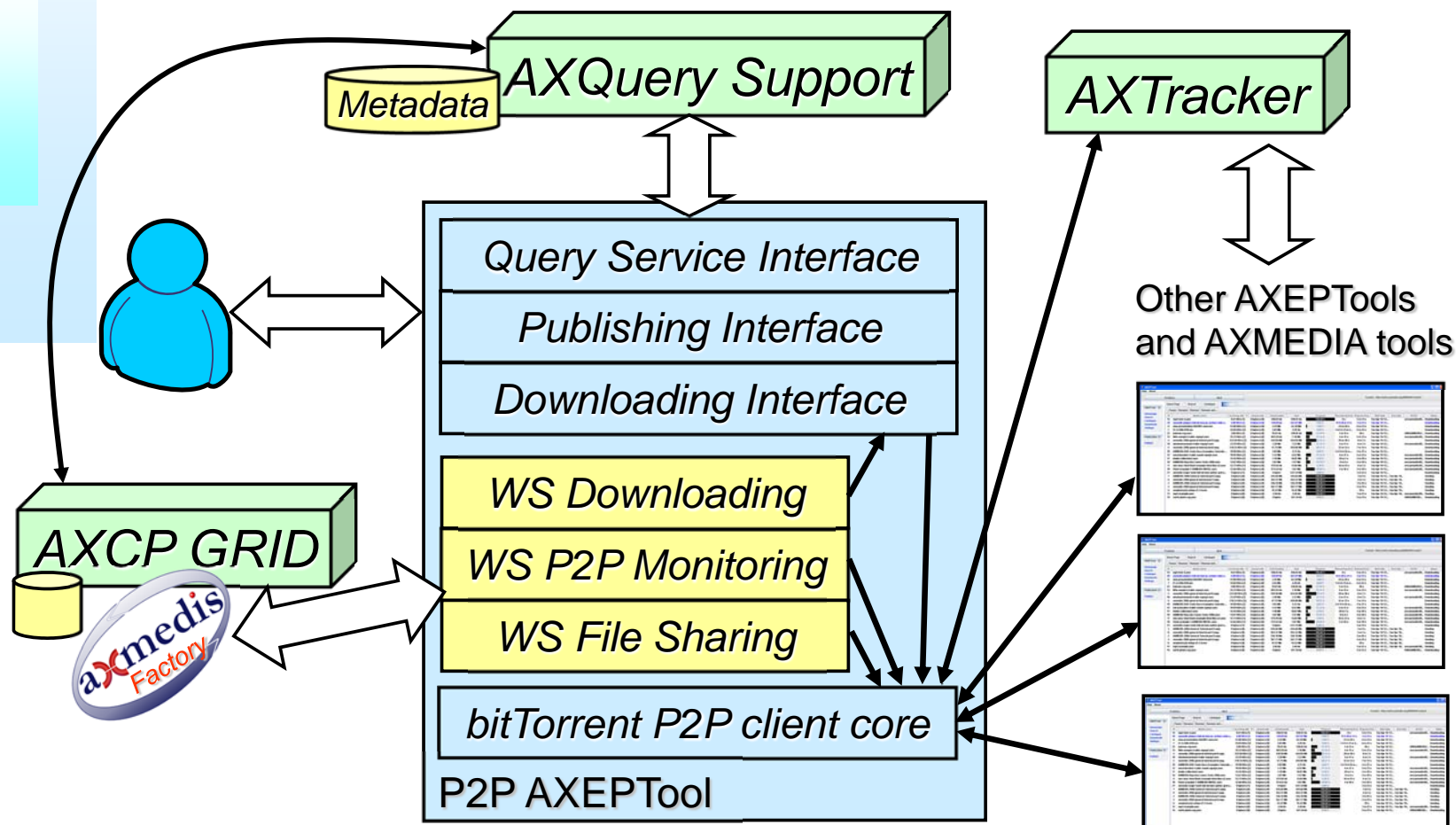


# Use Cases per AXEPTool





# AXEPTool P2P client



Derivato da Azureus/Vuze



# AXEPTool, P2P BT client, downloading



AXEPTool

Help About

Previous Next

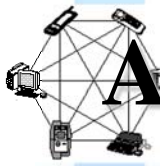
Tracker: <http://axtrk.axmedis.org:8080/AXTrackv2/>

Home Page Search Catalogue Downloads

Pause Resume Remove Remove and ...

#	Media name	Incoming rate	Upload rate	Downloaded	Size	Progress	Remaining time	Elapsed time	Start date	End date	AXOID	Status
14	mp4-test-4.axm	9.61 KB/s [1]	0 bytes/s [0]	450.92 kb	450.92 kb	100.00 %	0 s	5 m 23 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
20	axmedis-players-full-version-pc-activex-skib-v...	4.80 KB/s [1]	0 bytes/s [0]	120.69 kb	267.81 Mb	0.04 %	15 h 50 m 41 s	3 m 53 s	Tue Apr 10 12:...			Downloading
8	siae-presentation-feb2007-new.exe	41.86 KB/s [2]	0 bytes/s [0]	2.25 Mb	62.28 Mb	3.60 %	24 m 28 s	6 m 24 s	Tue Apr 10 12:...			Downloading
7	FC-6-i386-DVD.iso	33.84 KB/s [3]	0 bytes/s [0]	2.83 Mb	3.28 Gb	0.08 %	1 d 4 h 13 m 4...	6 m 42 s	Tue Apr 10 12:...			Downloading
22	batman-reg.axm	3.06 KB/s [1]	0 bytes/s [0]	70.81 kb	330.82 kb	21.39 %	1 m 24 s	38 s	Tue Apr 10 12:...		URN:AXMEDIS:F...	Downloading
13	little-vampire-trailer-mpeg4.axm	25.33 KB/s [2]	0 bytes/s [0]	893.55 kb	7.18 Mb	12.14 %	4 m 15 s	5 m 23 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
4	axmedis-2006-general-tutorial-part5.mpg	243.50 KB/s [3]	0 bytes/s [0]	159.58 Mb	454.05 Mb	35.14 %	20 m 38 s	8 m 2 s	Tue Apr 10 12:...			Downloading
10	whatwomenwant-trailer-mpeg4.axm	23.29 KB/s [2]	0 bytes/s [0]	1.28 Mb	7.22 Mb	17.71 %	4 m 21 s	6 m 2 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
5	axmedis-2006-general-tutorial-part6.mpg	218.24 KB/s [3]	0 bytes/s [0]	47.75 Mb	459.80 Mb	10.37 %	32 m 13 s	7 m 14 s	Tue Apr 10 12:...			Downloading
19	AXMEDIS-DVD-Tools-Docs-Examples-Tutorials...	19.98 KB/s [2]	0 bytes/s [0]	1.03 Mb	4.31 Gb	0.02 %	2 d 14 h 55 m ...	4 m 37 s	Tue Apr 10 12:...			Downloading
12	miraclemaker-trailer-movie-mpeg4.axm	19.95 KB/s [2]	0 bytes/s [0]	1.12 Mb	6.52 Mb	17.23 %	4 m 37 s	5 m 23 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
17	Audio-Collection2.axm	14.45 KB/s [2]	0 bytes/s [0]	1.10 Mb	18.07 Mb	6.10 %	20 m 2 s	4 m 39 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
16	AXMEDIS-Req-Use-Cases-Tests-2006.axm	13.67 KB/s [1]	0 bytes/s [0]	1.07 Mb	7.57 Mb	14.15 %	8 m 6 s	4 m 39 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
9	xim-ansc-html-flash-example-html-files-v2.axm	12.73 KB/s [1]	0 bytes/s [0]	575.95 kb	13.04 Mb	4.30 %	16 m 43 s	6 m 2 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
18	Flash-example-2-AXMEDIS-WEDEL.axm	12.66 KB/s [1]	0 bytes/s [0]	574.52 kb	1.87 Mb	29.86 %	1 m 46 s	4 m 39 s	Tue Apr 10 12:...		urn:axmedis:00...	Downloading
21	axmedis-major-tools-full-version-author-grid-v...	0 bytes/s [1]	0 bytes/s [0]	0 bytes	537.76 Mb	0.00 %		3 m 53 s	Tue Apr 10 12:...			Downloading
1	AXMEDIS-2006-General-Tutorial-part2.mpg	0 bytes/s [0]	0 bytes/s [0]	445.66 Mb	445.66 Mb	100.00 %		3 m 9 s	Tue Apr 10 12:...	Tue Apr 10...		Seeding
0	axmedis-2006-general-tutorial-part1.mpg	0 bytes/s [0]	0 bytes/s [0]	363.37 Mb	363.37 Mb	100.00 %		4 m 4 s	Tue Apr 10 12:...	Tue Apr 10...		Seeding
2	AXMEDIS-2006-General-Tutorial-part3.mpg	0 bytes/s [0]	0 bytes/s [0]	336.10 Mb	336.10 Mb	100.00 %		3 m 55 s	Tue Apr 10 12:...	Tue Apr 10...		Seeding
3	axmedis-2006-general-tutorial-part4.mpg	0 bytes/s [0]	0 bytes/s [0]	367.77 Mb	367.77 Mb	100.00 %		4 m 25 s	Tue Apr 10 12:...	Tue Apr 10...		Seeding
6	axeptool-p2p-setup-v3-2-6.exe	0 bytes/s [0]	0 bytes/s [0]	15.32 Mb	15.32 Mb	100.00 %		39 s	Tue Apr 10 12:...	Tue Apr 10...		Seeding
11	mp4-example.axm	0 bytes/s [0]	0 bytes/s [0]	3.46 kb	3.46 kb	100.00 %		1 m 47 s	Tue Apr 10 12:...	Tue Apr 10...	urn:axmedis:00...	Seeding
15	earth-photo-reg.axm	0 bytes/s [0]	0 bytes/s [0]	0 bytes	161.16 kb	0.00 %		5 m 23 s	Tue Apr 10 12:...		URN:AXMEDIS:...	Downloading





# AXMEDIS P2P Query Support



Home Page Search Catalogue Downloads Settings

**axmedis**

**AXMEDIS P2P Query Portal**

QuickQuery

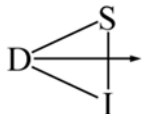
☐ Find Any Word

Metadata

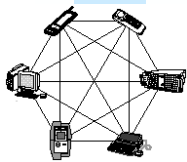
Match any ☒ or all ☐ query criteria

Abstract

(c) AXMEDIS 2006-2007







# AXTracker Catalogue

AXEPTool

Help About

Previous

Next

Home Page

Search

Catalogue

Downloads

Settings

AXEPTool

Homepage

Search

Catalogue

Downloads

Settings

Publication

Publish

AXMEDIS Tracker Catalogue

AXMEDIS Objects

File Name	File Size	AXOID
<a href="#">002.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:3b347184-4d25-40b0-9a5e-f6a85f3b8c96
<a href="#">241.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:921236a3-27ad-40d4-9698-11c0040e8b9a
<a href="#">Actors-Collection.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:c679fc59-5747-4d32-a254-68a9fd1275d8
<a href="#">aica-SLIDES.axm</a>	21.52 Mb	urn:axmedis:00000:obj:befe4f13-c968-4c10-95ce-2a6b850d33ec
<a href="#">allori.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:c4200ed7-bcd6-4abd-bf86-462745e96b33
<a href="#">Audio-Collection2.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:ee3c134d-b227-4a6b-9ad1-328978f28321
<a href="#">ax-glass-buttons-gifs.axm</a>	33.67 Kb	urn:axmedis:00002:obj:c1e4671f-1714-3215-848e-90c9c9390659
<a href="#">ax11_botticelli.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:3d7016b8-10dc-4591-ba58-d04eea2063d2
<a href="#">ax16_religious.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:15aaa627-61df-4d99-b6d3-17fec12f70d9
<a href="#">AXMEDIS-Demonstrators-Proto.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:1b9d41c8-e4a2-4646-b443-c0a2a03c8d95
<a href="#">AXMEDIS-Editor-Lesson-1.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:899333e6-51da-4321-877c-82a2fa47706a
<a href="#">AXMEDIS-Framework-and-Tools.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:abb57ad5-231d-498e-837c-8002a10e45fc
<a href="#">AXMEDIS-Req-Use-Cases-Tests-2006.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:fcc93ff8-f9a2-475c-85e6-5419e397888d
<a href="#">AXMEDIS-Specification-2006.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:4d06212b-eed6-4fab-9aca-df957e571c64
<a href="#">axpresentation.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:24c1f60d-5553-4dd3-b978-6c8211fc5e4c
<a href="#">batman-reg.axm</a>	0.0 Bytes	urn:axmedis:fffff:obj:c6921986-7e79-3753-863b-d6870d143530
<a href="#">clubitalia2.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:18ca1843-b034-4ee7-ae8f-b6c08cbe7951
<a href="#">demo_chap4.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:44c379a1-7621-4593-8257-00dd27227b99
<a href="#">dsi magnifier.axm</a>	0.0 Bytes	urn:axmedis:00000:obj:9295f310-61a3-49ef-aa5c-78706f637438

Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2012-2013

105

# Example of AXEPTool monitoring



P2P Report - Windows Internet Explorer

H:\AXCOM\AXCP-platform\Report-07-10-22.htm

File Edit View Favorites Tools Help

Google

Go

Bookmarks

Popups okay

Check

AutoLink

AutoFill

Send to

Settings

P2P Report

P2P Report

Page

Tools

## Summary of AXMEDIS P2P Node Status

### Node Profile (.....)

AXEPTool Version	Online since	Operating System	Username	JVM memory used
3.2.30	Wed Oct 24 18:48:46 CEST	Windows 2003 OS.Arch:ix86 OSVersion:5.2	Administrator	291.24 MB

### Report of activity on P2P node at .....

Object n.	Filename	Size	Download (%)	Start date	End date	AXOID	Status	Number of seeds	Number of peers
1	VRS-1-0c.axm	246.40 MB	100	Monday, October 22, 2007 17:18:23	Monday, October 22, 2007 20:27:11	um:axmedis:00002:obj:e6ac1ee5-7be8-38bc-b17a-29688acee123	Seeding	6	8
2	raw_ec3.axm	262.24 MB	100	Monday, October 22, 2007 18:21:17	Monday, October 22, 2007 20:18:12	um:axmedis:00000:obj:dd43b5be-f6c6-4bcf-9091-65ccba0b807f	Seeding	12	74
3	Xim-garden-P6201687-Mobile.axm	51.05 KB	100	Monday, October 22, 2007 17:10:50	Monday, October 22, 2007 19:36:18	um:axmedis:00000:obj:02ad02c9-1c66-4803-bd39-c73f6129721e	Seeding	10	48
4	VRS-059-Signalers.axm	54.78 MB	100	Monday, October 22, 2007 18:21:32	Monday, October 22, 2007 18:59:54	um:axmedis:00002:obj:7596a254-552b-3790-92d7-82564d6ba46a	Seeding	134	348
5	Xim-garden-P6201658-HiRes.axm	677.01 KB	100	Monday, October 22, 2007 17:10:22	Monday, October 22, 2007 18:50:17	um:axmedis:00000:obj:55033b54-63fc-48ba-af77-c8d025b8785e	Seeding	390	456
6	Xim-garden-P6201675-PDA.axm	116.89 KB	100	Monday, October 22, 2007 17:10:40	Monday, October 22, 2007 18:34:33	um:axmedis:00000:obj:2f476004-b695-4187-a4e9-cea51bee189d	Seeding	5	9
7	Xim-garden-P6201663-PDA.axm	120.11 KB	100	Monday, October 22, 2007 17:10:27	Monday, October 22, 2007 18:26:23	um:axmedis:00000:obj:2bf0c77c-4f6f-4cbf-adfc-459a8141015c	Seeding	456	563
8	REVIEW2007_18_OCT_REVIEW2007_18_Oct_tiscalitest_Akkordzither.axm	546.71 KB	100	Monday, October 22, 2007 18:21:43	Monday, October 22, 2007 18:24:33	um:axmedis:00000:obj:0db9bf4d-2b00-43ac-b39e-61ab118aba88	Seeding	76	82
9	REVIEW2007_15_OCT_B002_4_image.jpg_MASTER_REVIEW2007_IvanOCT_15.axm	255.96 KB	100	Monday, October 22, 2007 18:21:20	Monday, October 22, 2007 18:24:07	um:axmedis:00000:obj:e5fe2d44-979c-4611-993f-9d2c145a5652	Seeding	45	58
10	VRS-demo-0c-mpeg4.axm	61.27 MB	100	Monday, October 22, 2007 17:09:47	Monday, October 22, 2007 18:13:28	um:axmedis:00000:obj:e4497340-ff6c-4c4d-b6f9-2c86f3826823	Seeding	12	85
...	...	...	...	...	...	...	...	...	...

Number of objects	Total Disk Space	Free Disk Space	Total Objects Size
2172	219.83 GB	192.82 GB	14.63 GB

AXMEDIS 2007 - Thu Oct 25 2007 12:48:42 GMT+0200 (ora legale Europa occidentale)



Done

My Computer

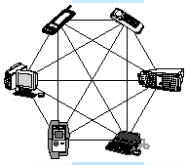
100%





# P2P Network control functionalities

- *publishing*(AXEPToolURL, FileURI, AXTrackerURL)
- *download*(AXEPToolURL, FileName or AXOID or BitTorrentURL)
- *listContent listPublished*(AXEPToolURL)
- *listContent listDownloaded*(AXEPToolURL)
- *infoStatus status*(AXEPToolURL, FileName or AXOID)
- *controlDownload*(AXEPToolURL, FileName or AXOID)
- *listAXOID query*(AXQuerySupportURL, Query)
- *listContent catalogue*(AXTrackerURL)
- *delete*(AXEPToolURL, FileName or AXOID)
- *FileURI get*(AXOID or FileName)
- *listAXEPTool listAXEPTool*(AXTrackerURL)
- *infoNode statusNode*(AXEPToolURL)
- *infoTracker statusTracker*(AXTrackerURL, period)

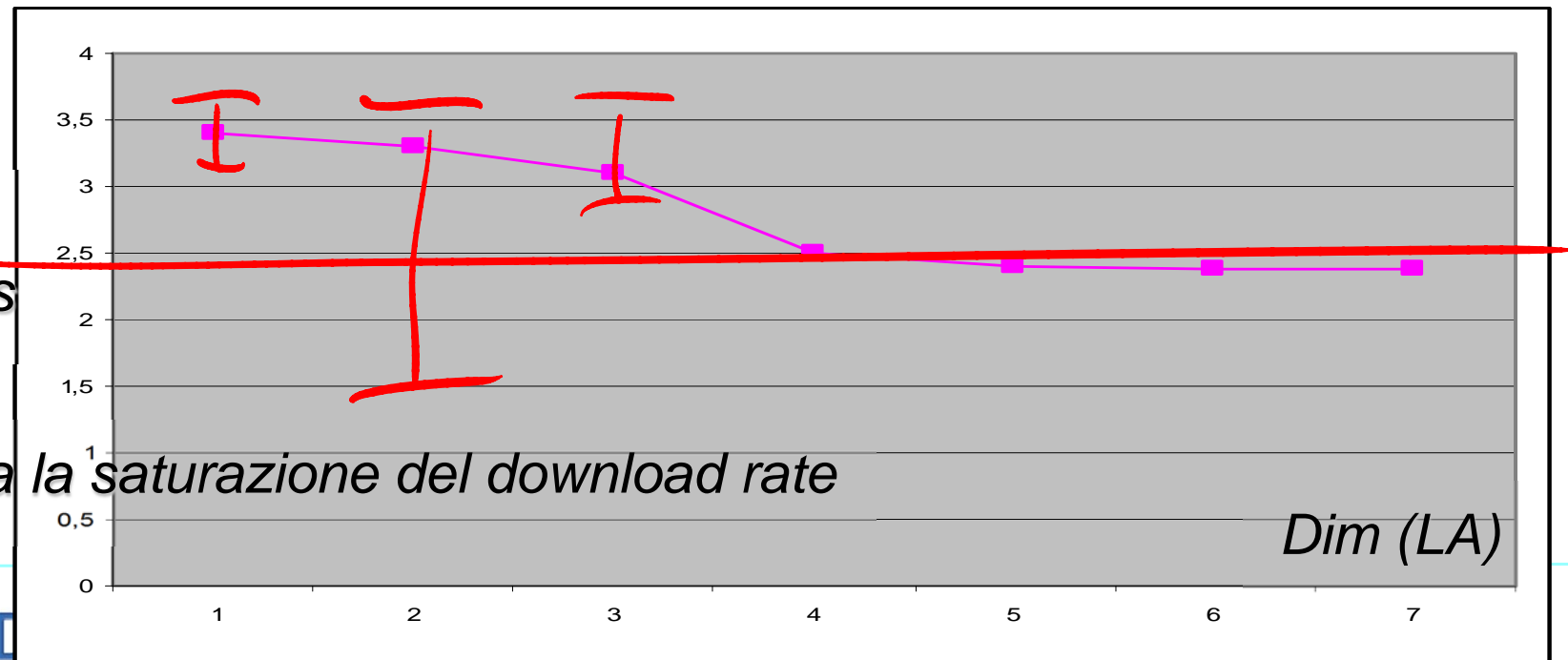


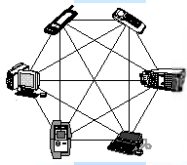
# P2P Experiments

- Publishing an object on all the AXEPTools of the P2P Network for example to accelerate the seeding of a given object on the network:
  - ♣ *Axoid=getAXOID(MyFile);*
  - ♣ *publishing(MyAXEPTool, MyFile, TheAXTracker);*
  - ♣ *LA = listAXEPTool(TheAXTracker);*
  - ♣ *For each la of LA: download(la, Axoid);*

□ *Minuti per  
XXX Mbytes*

□ *Raggiunta la saturazione del download rate*



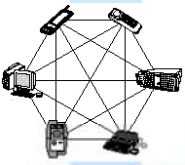


# Programming P2P Experiments



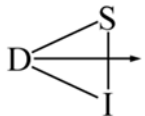
- Notifying the publication of new objects/files in the network among the different AXCP GRIDs/AXEPTools controlled;
- Removing/deleting of objects/files from the network, at least from the AXEPTool nodes and from the AXTrackers and AXQuery Support;
- Monitoring the status of the P2P Network: discovering which are the most active/virtuous AXEPTools, their capabilities, how many downloads have been performed, how many segments have been provided and for whose objects/files, then they are active, etc. This allows to perform specific analysis to assess the reputation of Business actors on the basis of their behavior on the corresponding AXEPTool;
- Controlling the content seeded by the AXEPTools, for example constrained them to become an exact replica of each other (uniforming the seeding distribution), or imposing some distribution for the content in the network of the AXEPTools on the basis of the content distribution and statistical analysis;
- Activating automated queries for obtaining, downloading and posting these objects into the database of specific content collection on the basis of complex queries. So that, these active queries can be periodically activated to verify if some new content satisfy the criteria and in the positive case, the automated download can be activated as well;
- Activating automated publishing on the P2P Network of accessible collections from the AXCP GRID and crawling facilities

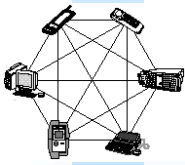




○ P2P IPTV

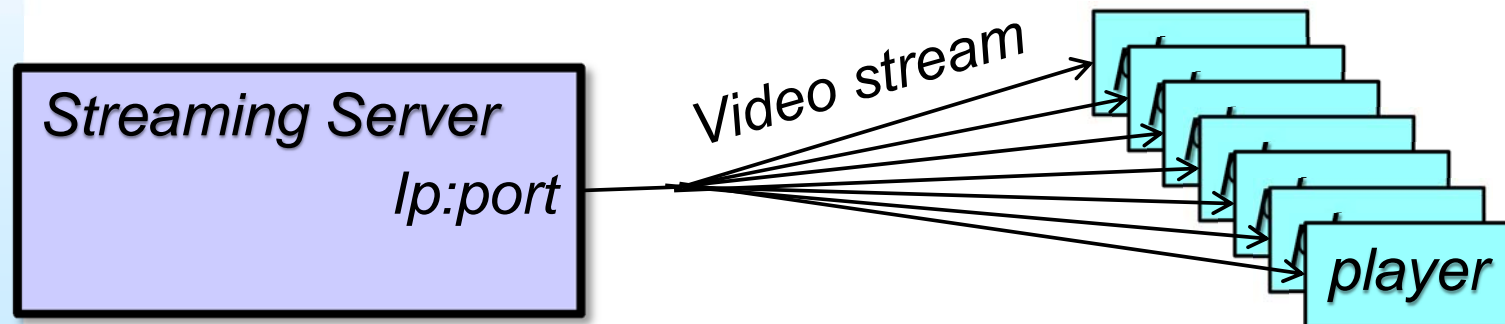
○ P2P VOD





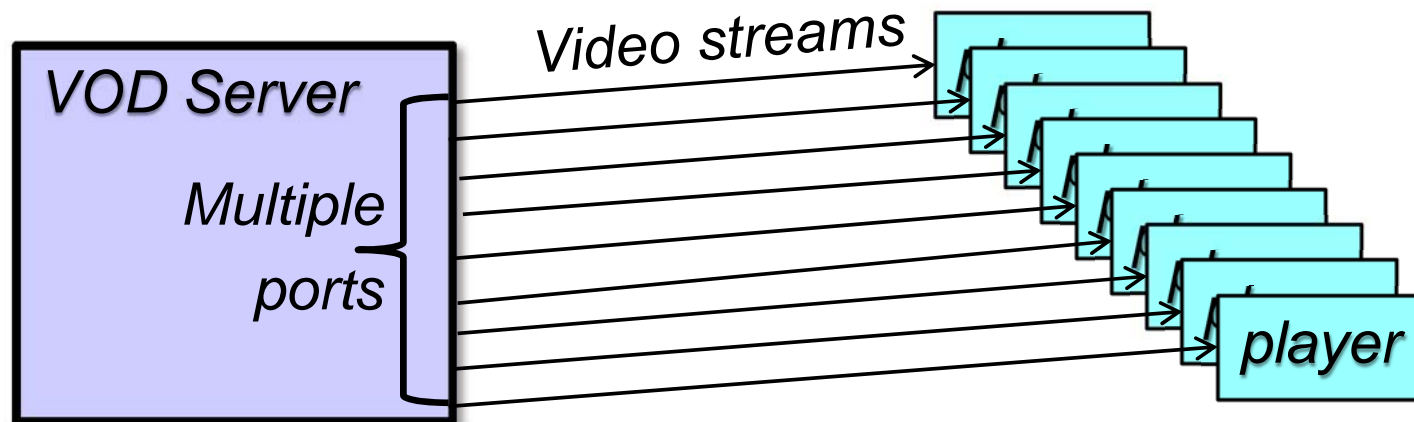
# IPTV vs VOD

- **IPTV: TV via Internet Protocol**



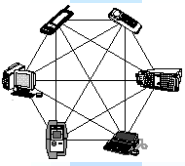
$O(1)+O(n)$  if QoS

- **VOD: Video on Demand**



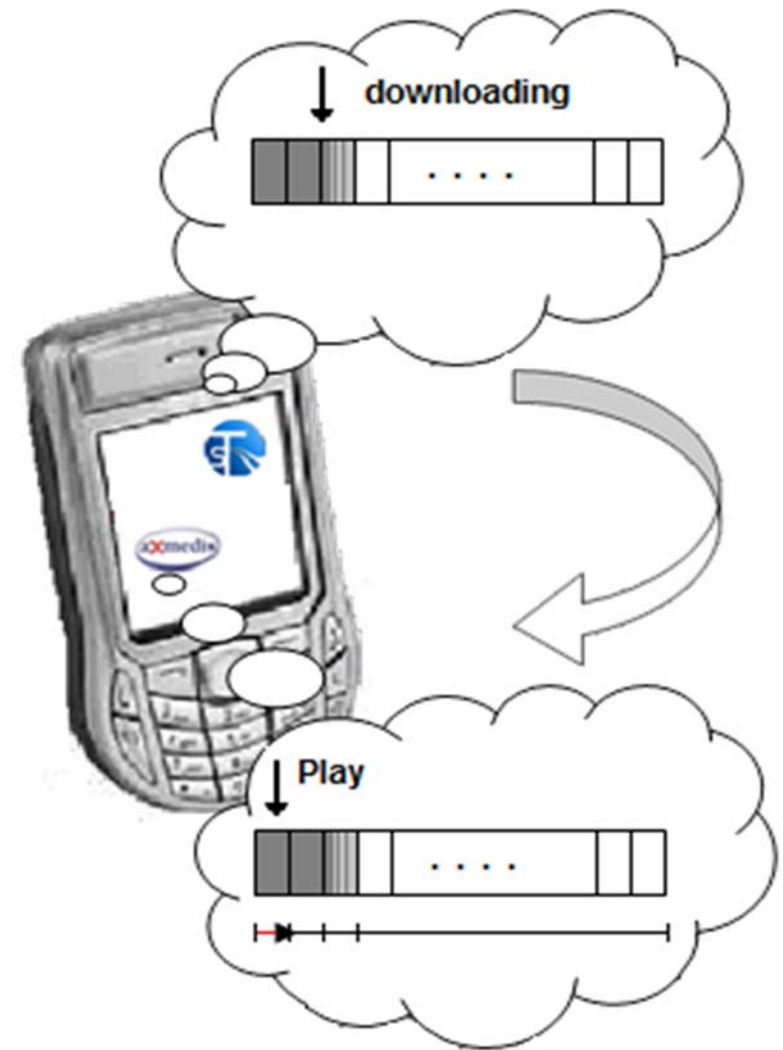
$O(N)$  in any case: 300bps x 10.000 Clients



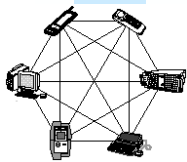


# P2P Progressive Download

- The cost of streaming is very high since the number of streams/user supported depends on the max bandwidth supported by the server:
  - ♣ e.g., 10Mbps  $\rightarrow$  50x200 kbps, YouTube...
- P2P solution reduces the costs of distribution while the user has to wait for the download to play the content.
- Progressive P2P supports the play while P2P downloading.





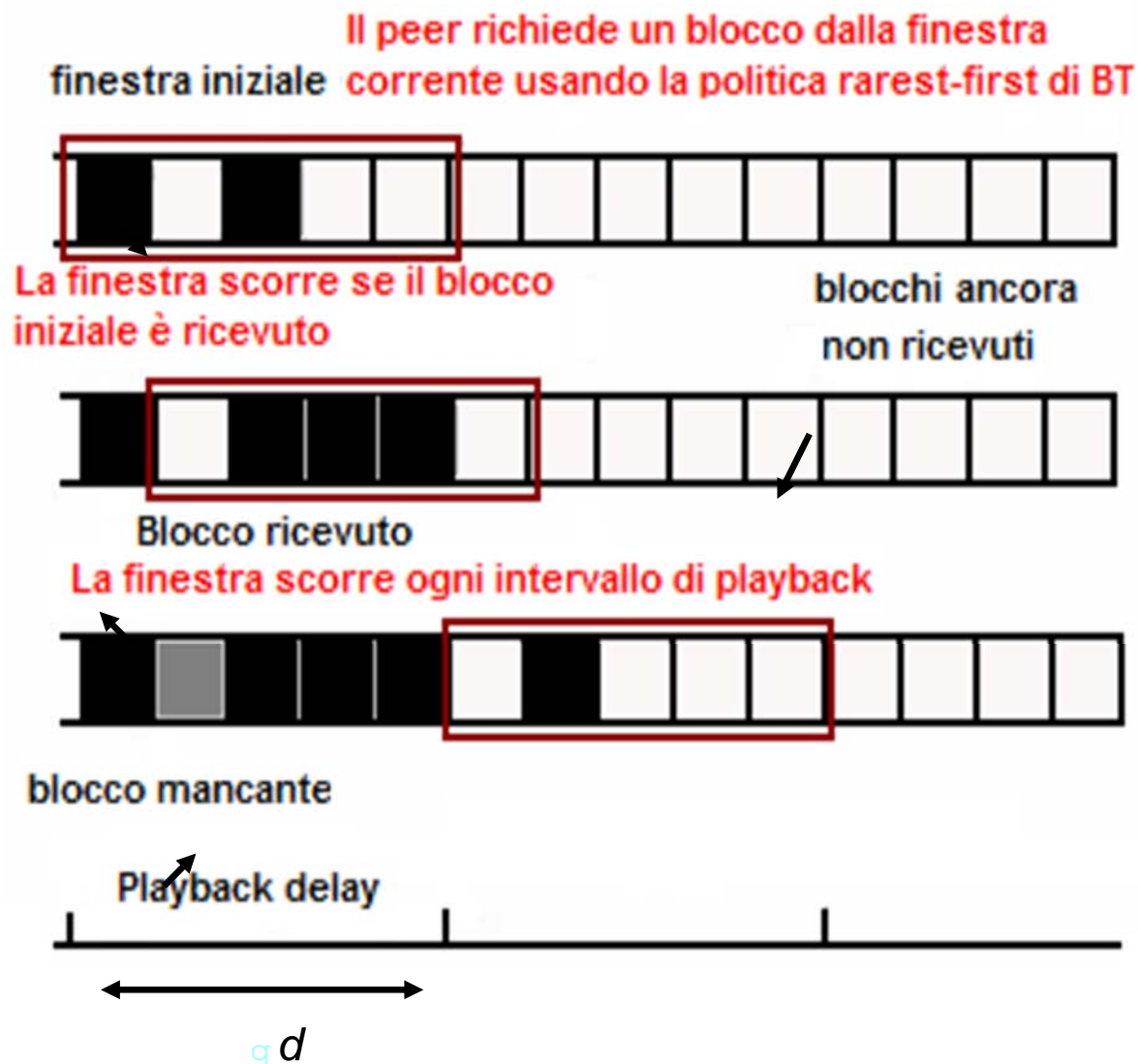


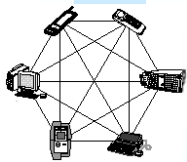
# BitTorrent: progressive download

Individuare un compromesso tra la Rarest First Policy e il download sequenziale.

➤ finestra di segmenti definite “*urgenti*”

Fondamentale la scelta della dimensione della finestra.





# Applicazione SymTorrent:

## Adattamento al download sequenziale

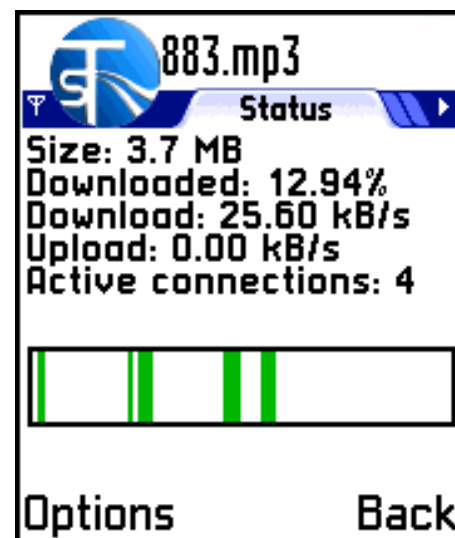
q Riproduzione del concetto di finestra di segmenti:

- La dimensione ottimale della finestra deve rispettare la seguente equazione

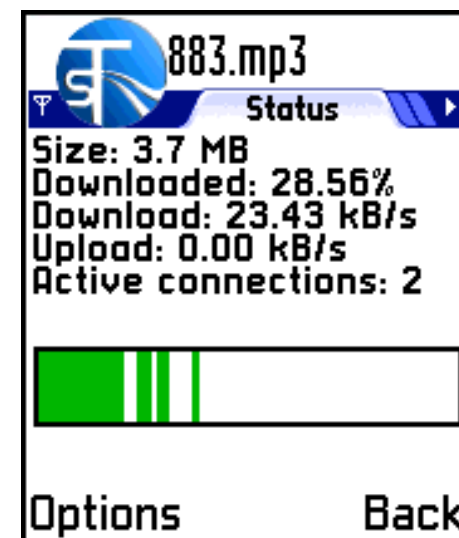
$$w = \frac{d * b}{c}$$

- q  $w = n^{\circ}$  blocchi nella finestra
- q  $d = \text{playback delay (secondi)}$
- q  $b = \text{BitRate file (Kb/s)}$
- q  $c = \text{dimensione blocco in byte}$

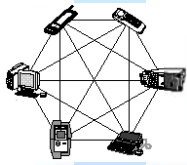
- Estrazione del BitRate dall'header del file (formato WAVE e MP3);
- Uso della *rarest first policy* nella finestra;
- Gestito lo spostamento della finestra;



q **download non sequenziale (SymTorrent originale)**



q **download sequenziale**



# Implementazione Multimedia Streaming

- Crea un'interazione tra le applicazioni: garantita la contemporaneità dei servizi forniti dalle due
- La strategia di buffering implementata fornisce all'utente una stima del tempo di attesa necessario per garantire una riproduzione fluida del media:

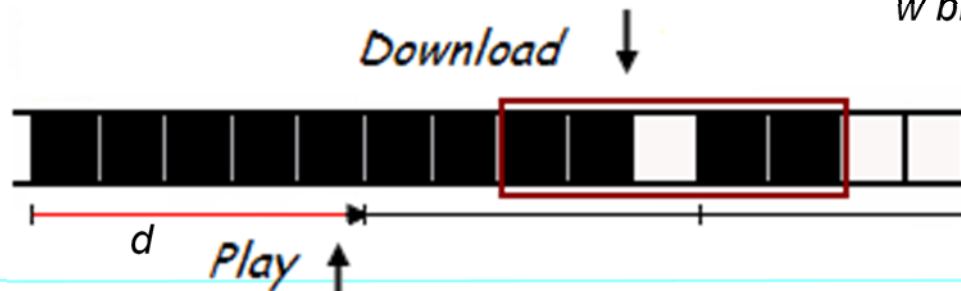
→ La formula  $w = \underline{d} * b$  garantisce la riproduzione dei primi  $d$  secondi

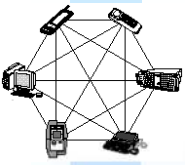
→ Terminati i  $d$  secondi, è necessario rispettare la relazione

$$S = \frac{w * c}{DR} \leq d$$

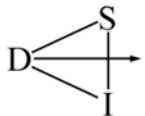
$DR$  = velocità media di download del file dalla rete

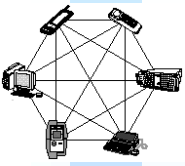
$S$  = secondi necessari a scaricare i  $w$  blocchi successivi





# Monitoraggio reti P2P





## ○ Monitoraggio reti P2P

- ♣ Analisi del traffico dei download

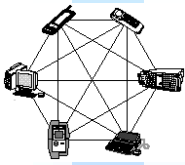
## ○ Motivazioni:

- ♣ Misurare l'effettivo gradimento di un prodotto di mercato tramite la sua diffusione sul P2P
- ♣ Quantificare il fenomeno di download illegale, per eventuale ripartizione dei diritti forfettari
- ♣ Statistiche sulla distribuzione di contenuti digitali

## ○ Come:

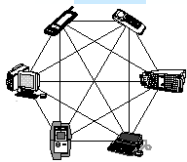
- ♣ Fare delle analisi periodiche
- ♣ Analizzare i risultati territoriali
- ♣ Produrre dei report anche per estrapolazione



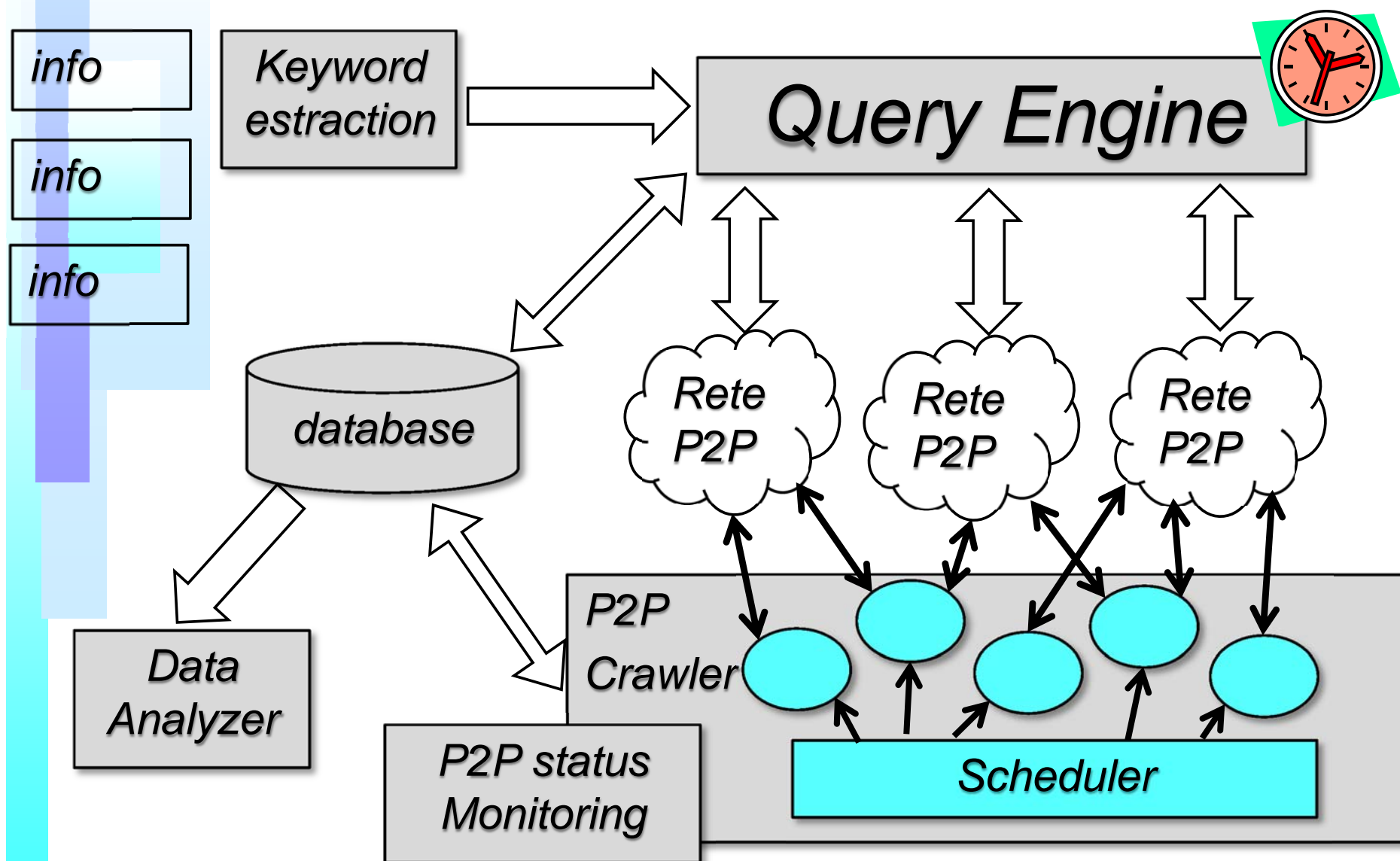


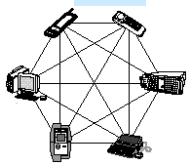
# Requisiti P2P Monitoring

- Fare un'analisi della rete e della diffusione dei download sulla base di keyword prelevate identificate dal committente
  - ♣ Keyword prese dalle top 20, top 10, eventi, etc.
  - ♣ Nuovi prodotto di mercato
  - ♣ Nuove uscite dei film
  - ♣ Etc.
  
- Periodicamente fare questa analisi per:
  - ♣ comprendere il trend
  - ♣ Calcolare un volume di scaricamenti o pervasivita'

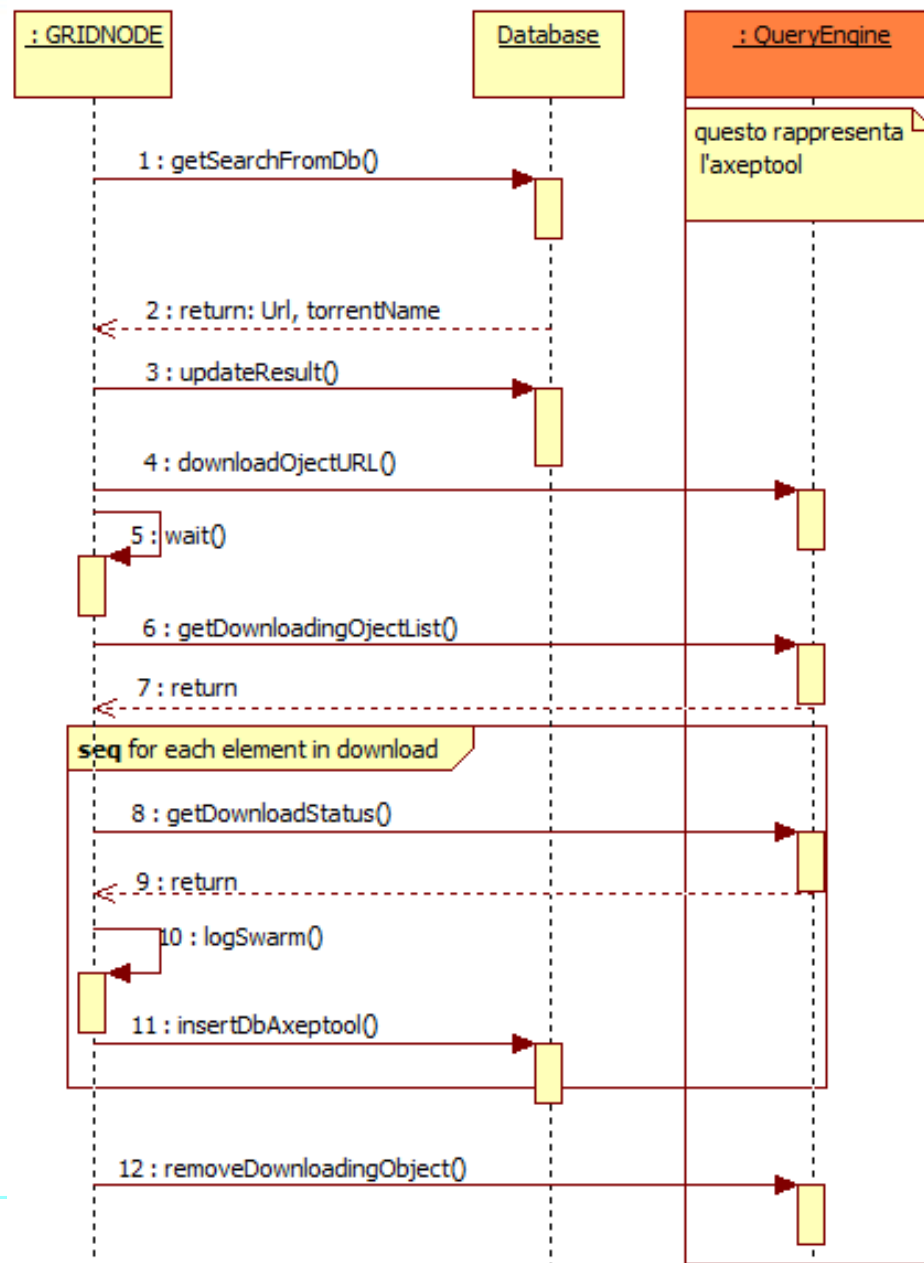
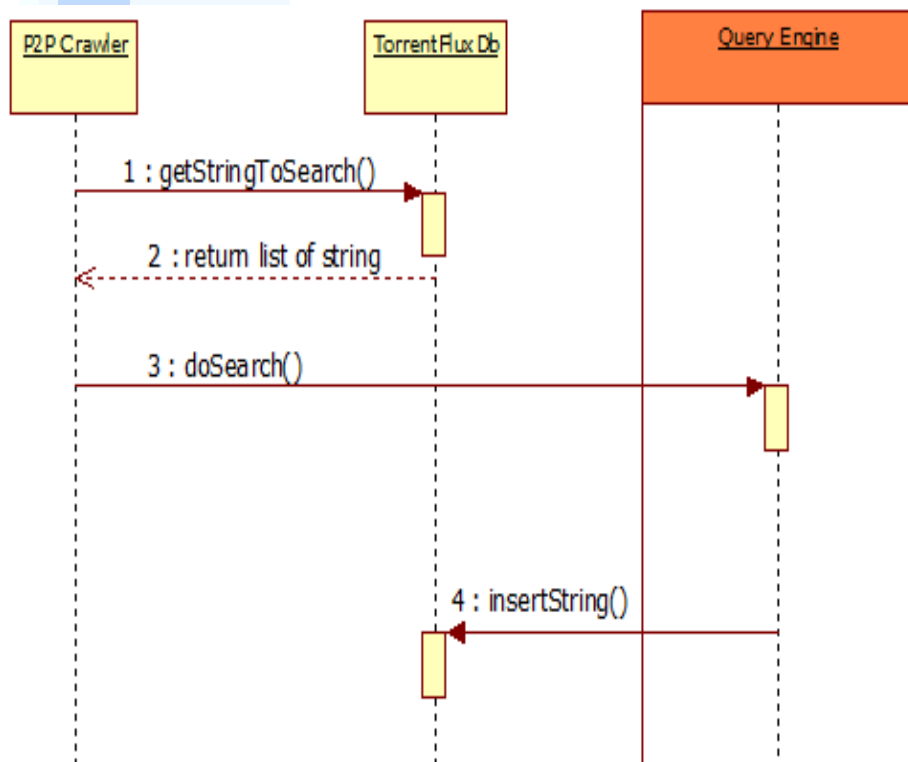


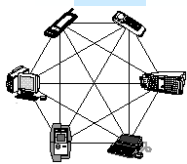
# Architettura





# Sequence Diagrams

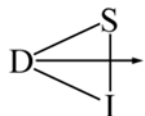


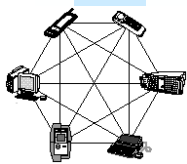


# Example of track record

[New Setting](#) | [Current setting](#) | [Display results](#)

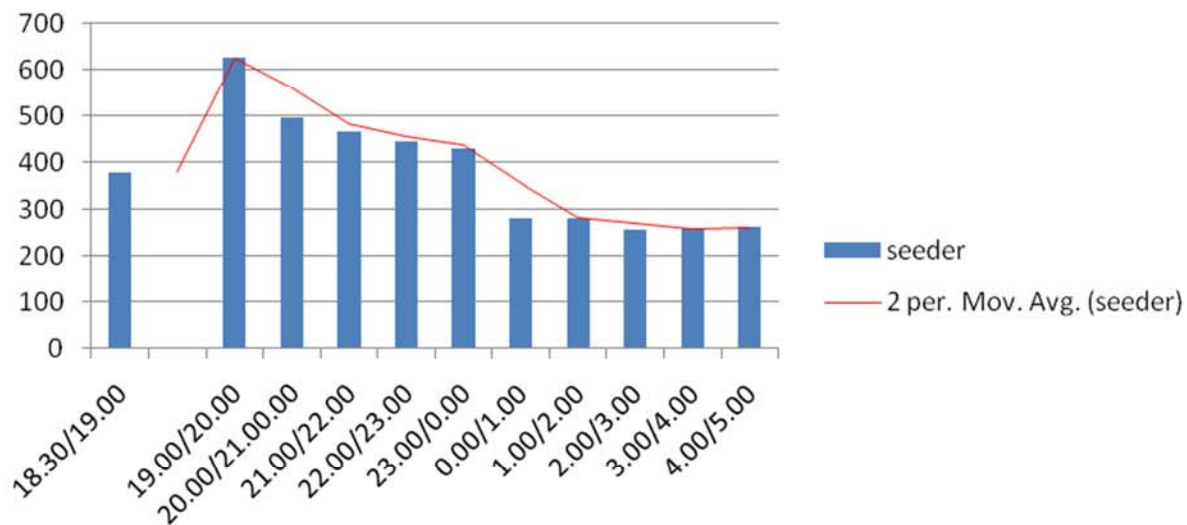
DateTime	TorrentName	IP	Percentage	Country
2010-09-17 18:21:06	james blunt - back to bedlam (with covers) a dhz.inc release	84.101.124.129	100	France
2010-09-17 18:21:05	james blunt - back to bedlam (with covers) a dhz.inc release	88.180.134.126	100	France
2010-09-17 18:21:05	james blunt - back to bedlam (with covers) a dhz.inc release	86.197.28.245	100	France
2010-09-17 18:21:05	james blunt - back to bedlam (with covers) a dhz.inc release	90.39.205.250	100	France
2010-09-17 18:19:52	now thats what i call music 70 (with covers) a dhz.inc release	94.202.16.4	100	United Arab Emirates
2010-09-17 18:19:52	now thats what i call music 70 (with covers) a dhz.inc release	86.152.66.116	100	United Kingdom
2010-09-17 18:19:52	now thats what i call music 70 (with covers) a dhz.inc release	86.7.70.69	100	United Kingdom
2010-09-17 18:19:52	now thats what i call music 70 (with covers) a dhz.inc release	85.210.162.249	100	United Kingdom
2010-09-17 18:19:48	gary go-gary go 2009 kompletlywyred dhz inc release	78.248.38.75	100	France
2010-09-17 18:19:46	top 40 singles usa 13 03 2010 dhz inc release	88.88.190.104	100	Norway
2010-09-17 18:19:43	david gray - greatest hits (with covers) a dhz.inc release	92.13.13.67	100	United Kingdom
2010-09-17 18:19:43	[2008] viva la vida or death and all his friend - coldplay	123.243.72.176	100	Australia
2010-09-17 18:19:43	david gray - greatest hits (with covers) a dhz.inc release	88.166.31.92	100	France
2010-09-17 18:19:43	david gray - greatest hits (with covers) a dhz.inc release	86.150.113.230	100	United Kingdom
2010-09-17 18:19:42	[2008] viva la vida or death and all his friend - coldplay	90.155.135.179	100	Russian Federation
2010-09-17 18:19:41	top 40 singles usa 13 06 2009 kompletlywyred dhz inc release	110.159.158.171	100	Malaysia
2010-09-17 18:19:41	va - a state of trance 2010 (mixed by armin van buuren) (2010)	91.107.15.112	100	United Kingdom
2010-09-17 18:19:41	[2008] viva la vida or death and all his friend - coldplay	92.136.170.108	100	France
2010-09-17 18:19:40	va - a state of trance 2010 (mixed by armin van buuren) (2010)	89.133.40.135	100	Hungary
2010-09-17 18:19:22	top 40 singles usa 01 11 2008 dhz inc release	78.12.80.31	100	Italy
2010-09-17 18:19:22	top 40 singles usa 01 11 2008 dhz inc release	68.107.15.244	100	United States
2010-09-17 18:19:21	david gray - greatest hits (with covers) a dhz.inc release	86.126.0.6	100	Romania



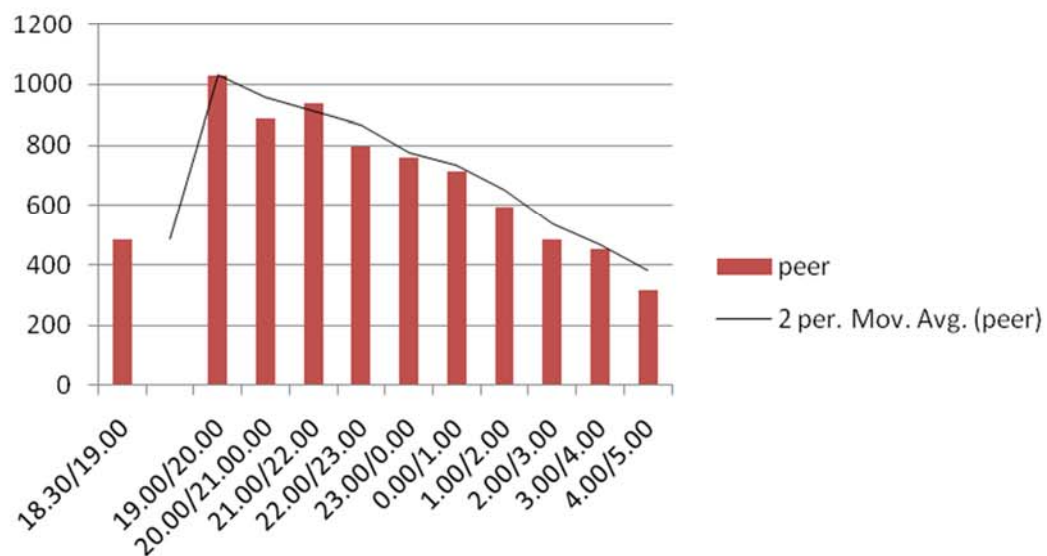


# Andamento dei Seeder/peer

seeder



peer



seeder + peer

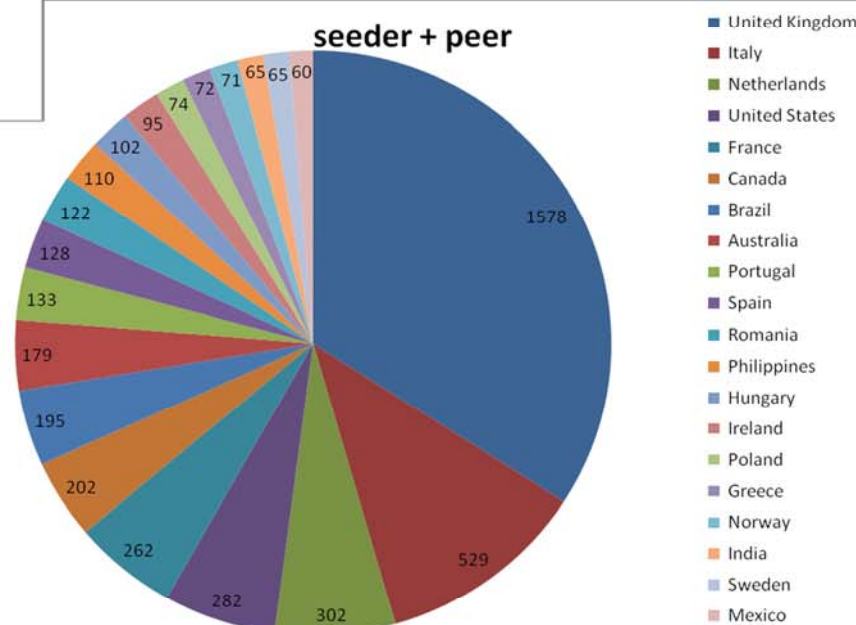
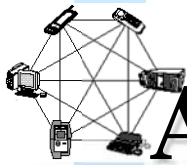
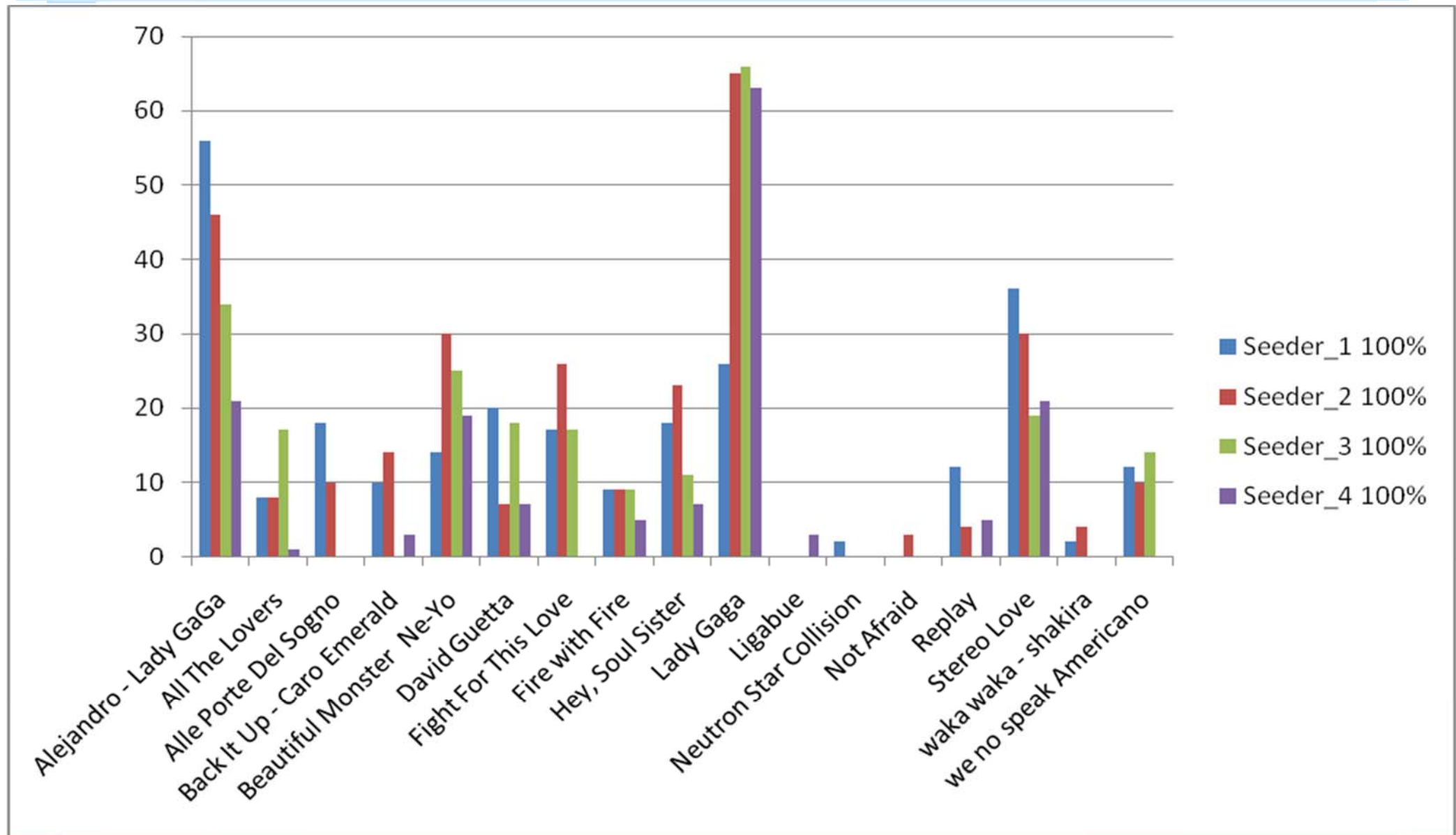


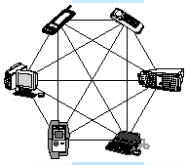
Figura 6.1 degli utenti per paese



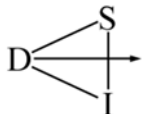


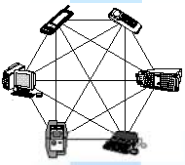
# Andamento dei Seeder nel tempo x key





- Analisi della situazione Italiana
  - ♣ Su quelle keyword
  - ♣ Per quel periodo
  - ♣ Senza estrapolazione
- Dai 2 siti P2P (TorrentPortal, TorrentBox)
  - ♣ In media 318000 record salvati
  - ♣ 5600 IP individuati per ogni 10 ore di esecuzione
  - ♣ Circa 2200 seeder
  - ♣ 46 file effettivamente condivisi



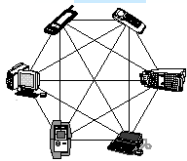


# ○ P2P, Db DHT

# ○ Distributed

# Trust





# P2P reciprocal Trusting for DRM

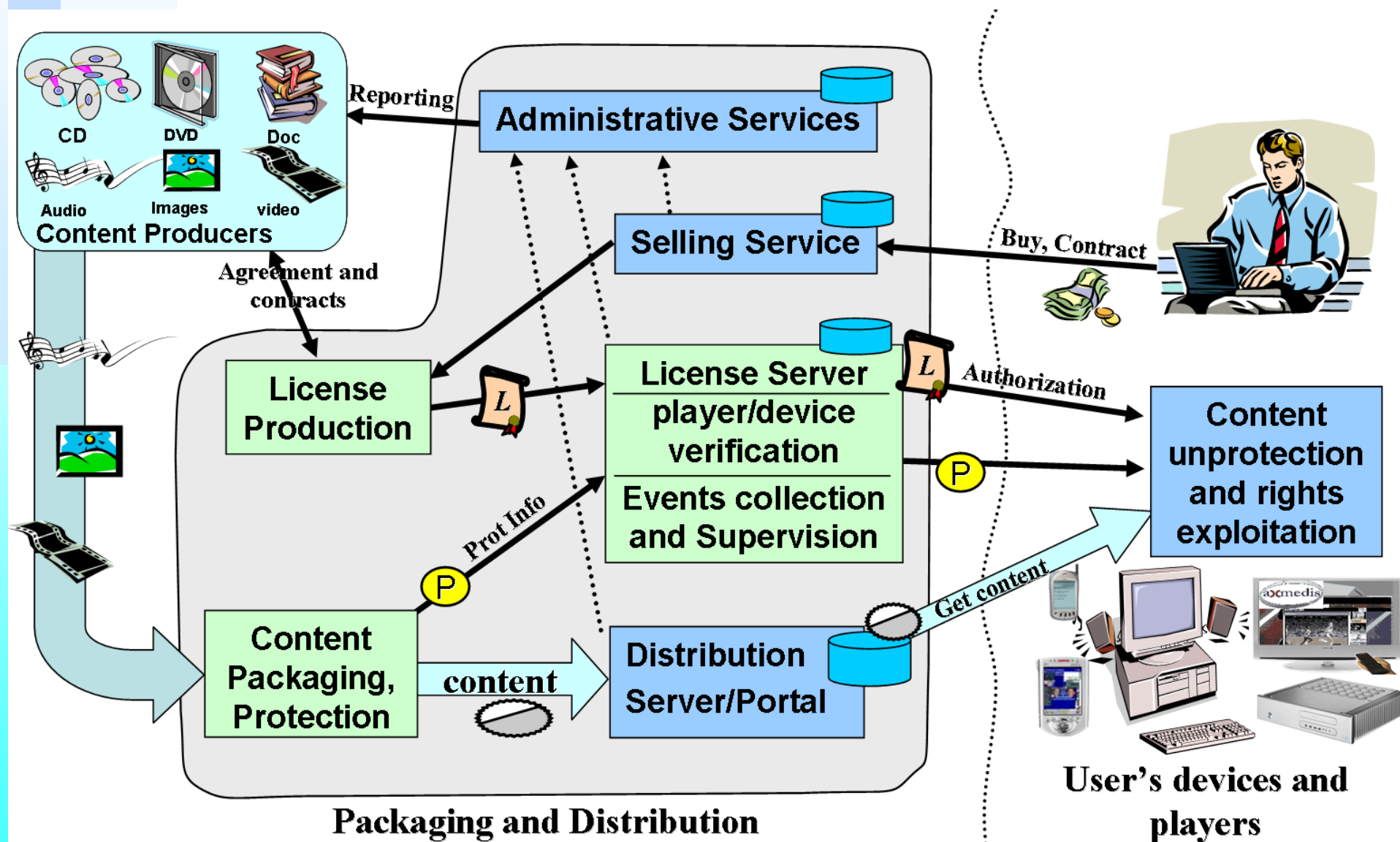
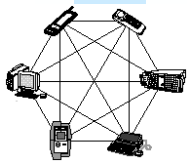


- le prestazioni del DRM AXMEDIS sono fortemente condizionate dal numero di chiamate contemporanee;
  - ♣ Ad esempio, in certi casi, ogni 8 richieste di autorizzazione in contemporanea, 2 non hanno esito positivo, a causa dei ritardi nell'elaborazione;
  - ♣ il DRM AXMEDIS ha complessità computazionale lineare 1:N (1 server:N client), e.g., 10 alla 12.  $O(R)$ ,  $O(U \cdot C)$ , un milione di utenti per un milione di content.

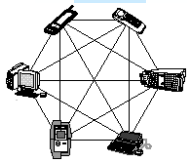
## Obiettivi:

- **Scalabilità:** definire e sviluppare una soluzione scalabile che consenta di far fronte a carichi elevati sul sistema, decentralizzando le risorse da aggiungere, in modo da poter soddisfare un elevato numero di richieste contemporanee.
- **Sicurezza:** adottare politiche adeguate alla nuova soluzione, mantenendo gli stessi standard di sicurezza









# Richiami sul DRM

## ○ AXMEDIS e'

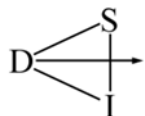
- ♣ un'infrastruttura distribuita multiplatforma per la creazione, protezione, distribuzione e consumo delle risorse digitali

## ○ Digital Rights Management, DRM:

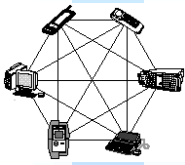
- ♣ meccanismi di gestione dei diritti di risorse digitali, grazie alla possibilità di rendere protette, identificabili e tracciabili le risorse stesse

## ○ DRM AXMEDIS fornisce

- ♣ una serie di strumenti che consentono di
  - ➔ certificare utenti e dispositivi,
  - ➔ gestire i diritti e
  - ➔ controllarne lo sfruttamento da parte degli utenti finali







# DRM AXMEDIS/MPEG-21

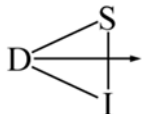
- Procedure fondamentali del DRM:

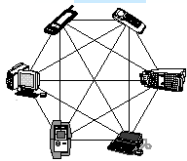
- **Server side**

- *registrazione di: utente e tool type*
- *protezione e registrazione del contenuto*
- *registrazione della licenza*

- **Client side**

- *certificazione del tool: salvataggio dei dati relativi al tool utilizzato*
- *verifica del tool: controllo sull'integrità*
- *autorizzazione: concessione del grant all'utente finale*





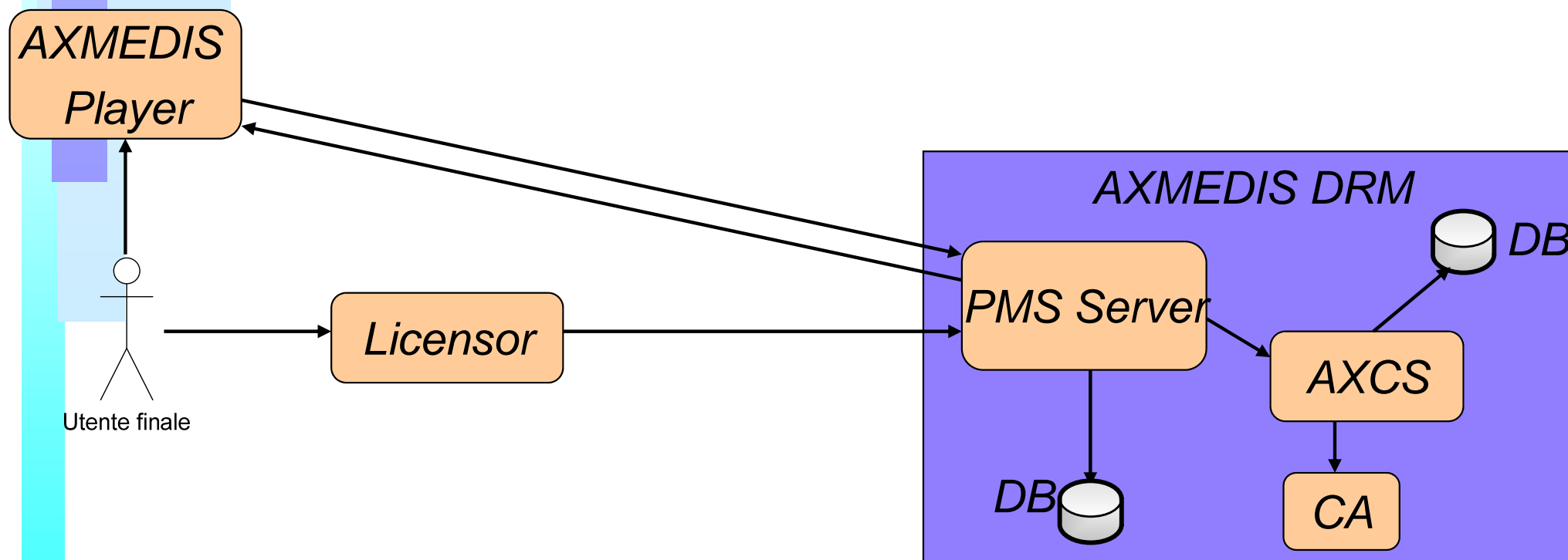
# DRM AXMEDIS Esempio

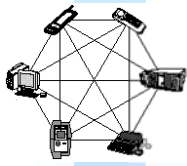
*Autorizzazione*

*Verifica del tool*

*Registrazione della licenza*

*Certificazione del tool*



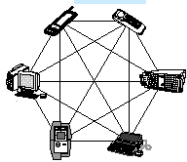


# Scelta della struttura decentralizzata

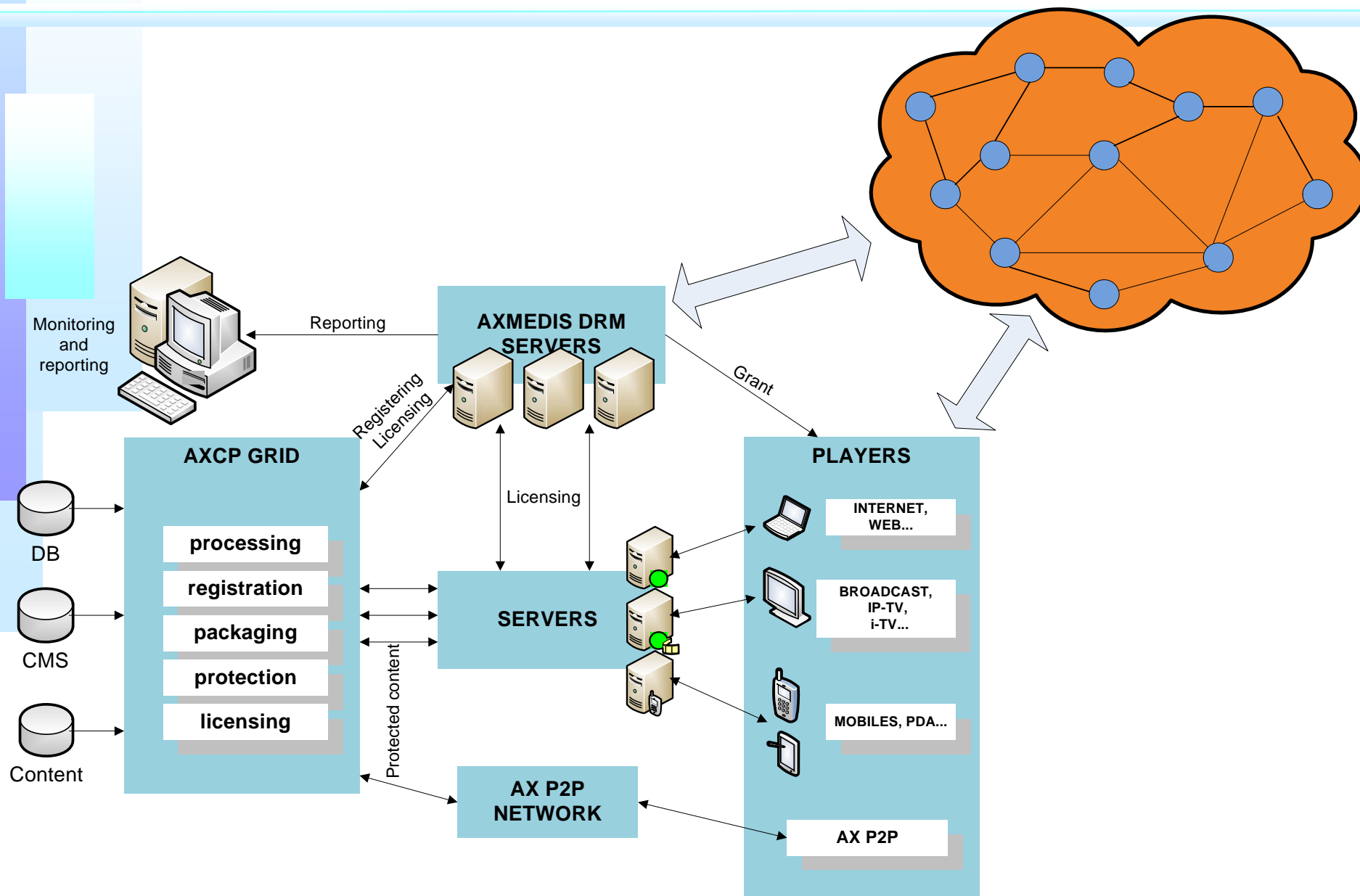
- **Si concretizza la decentralizzazione** scegliendo di delegare l'esecuzione di alcune procedure chiave ad una DHT su rete P2P
  - ♣ DHT: Distributed Hash Table:
  - ♣ Classe di sistemi distribuiti decentralizzati che partizionano l'appartenenza di un set di chiavi tra i nodi che fanno parte della rete P2P;
  - ♣ hanno complessità computazionale logaritmica
- **Perché Bamboo DHT:**
  - *due insiemi di vicini: leaf set e routing table*
  - *interfacce put/get predefinite: XML-RPC*
  - *gestione del churn*
  - *open source*

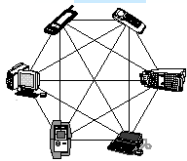




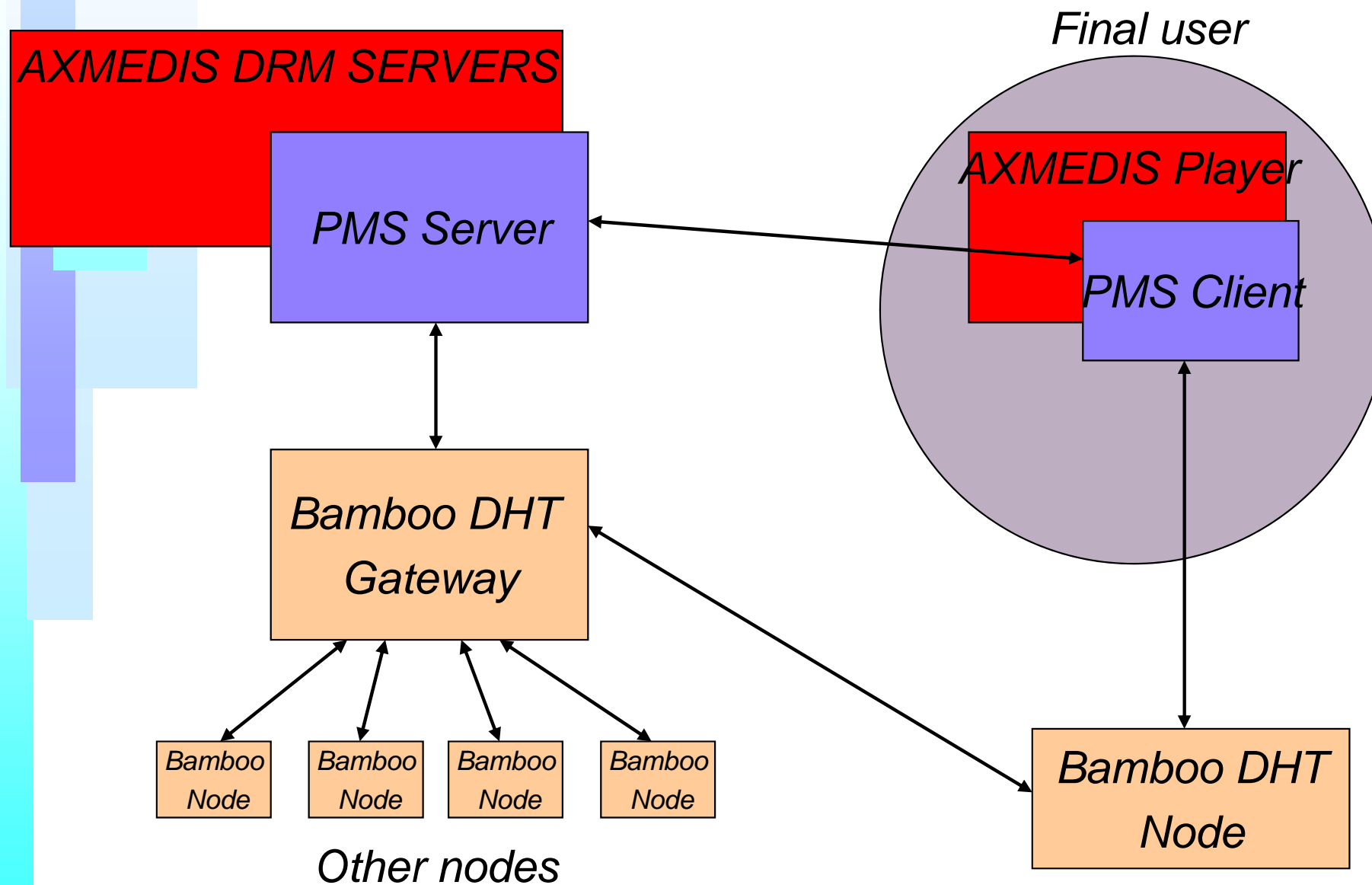


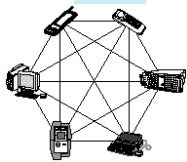
## RETE P2P



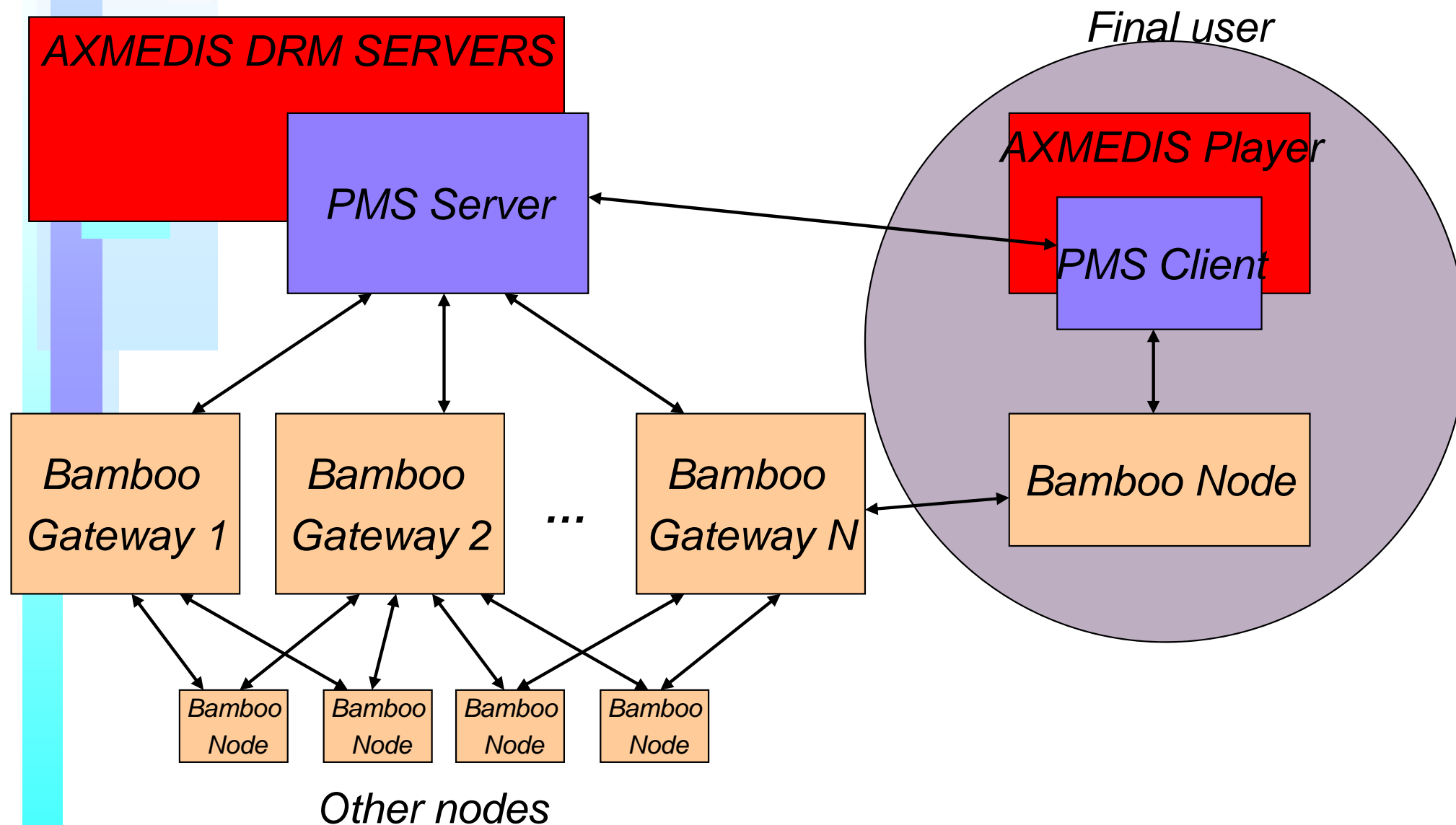


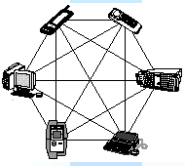
# Interazioni tra nodi e DRM





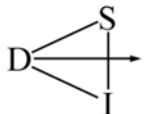
# Esempio di deployment 'ottimo'

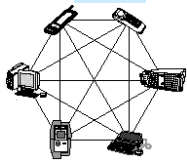




# Salvataggio dati sulla DHT

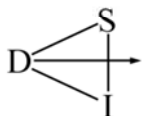
- **avviene** nei casi di
  - ♣ certificazione del tool e
  - ♣ registrazione della licenza.
- **si effettua** durante
  - ♣ la verifica del tool e
  - ♣ l'autorizzazione per l'utente finale.
- **Il PMS Server**, completate le rispettive procedure, effettua
  - ♣ una connessione ad un nodo Bamboo Gateway per trasmettere le informazioni utili.



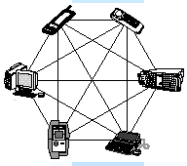


# Recupero dati dalla DHT

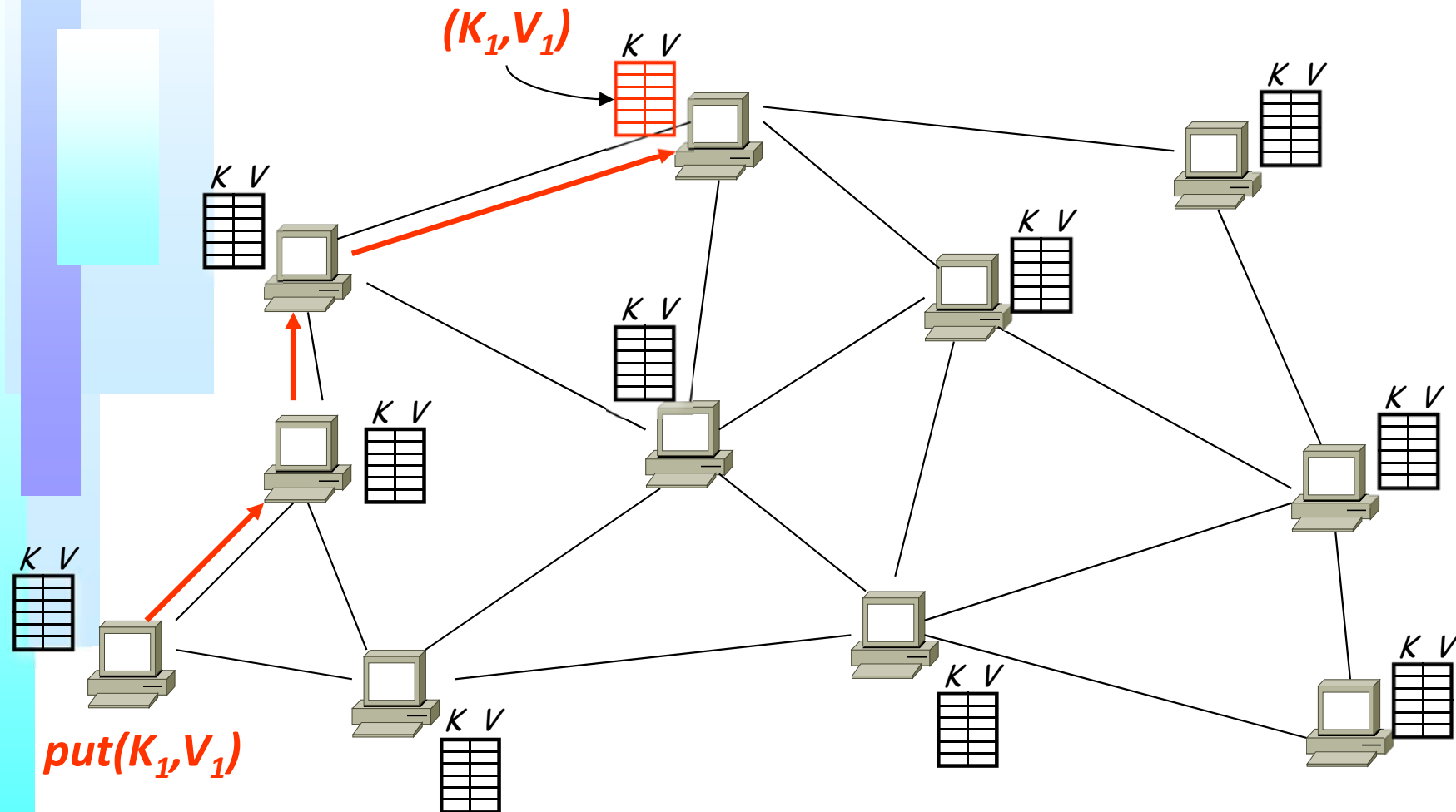
- **Si effettua, come scelta prioritaria,**
  - ♣ ogni volta che si eseguono le procedure di verifica e autorizzazione.
- **Spetta al PMS Client effettuare un lookup sulla DHT,**
  - ♣ in base alla chiave specifica per la relativa procedura.
- **Quando viene trovata una corrispondenza sulla rete P2P,**
  - ♣ la procedura viene espletata direttamente,
  - ♣ senza dover ricorrere ai server di DRM AXMEDIS.



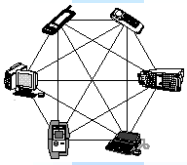




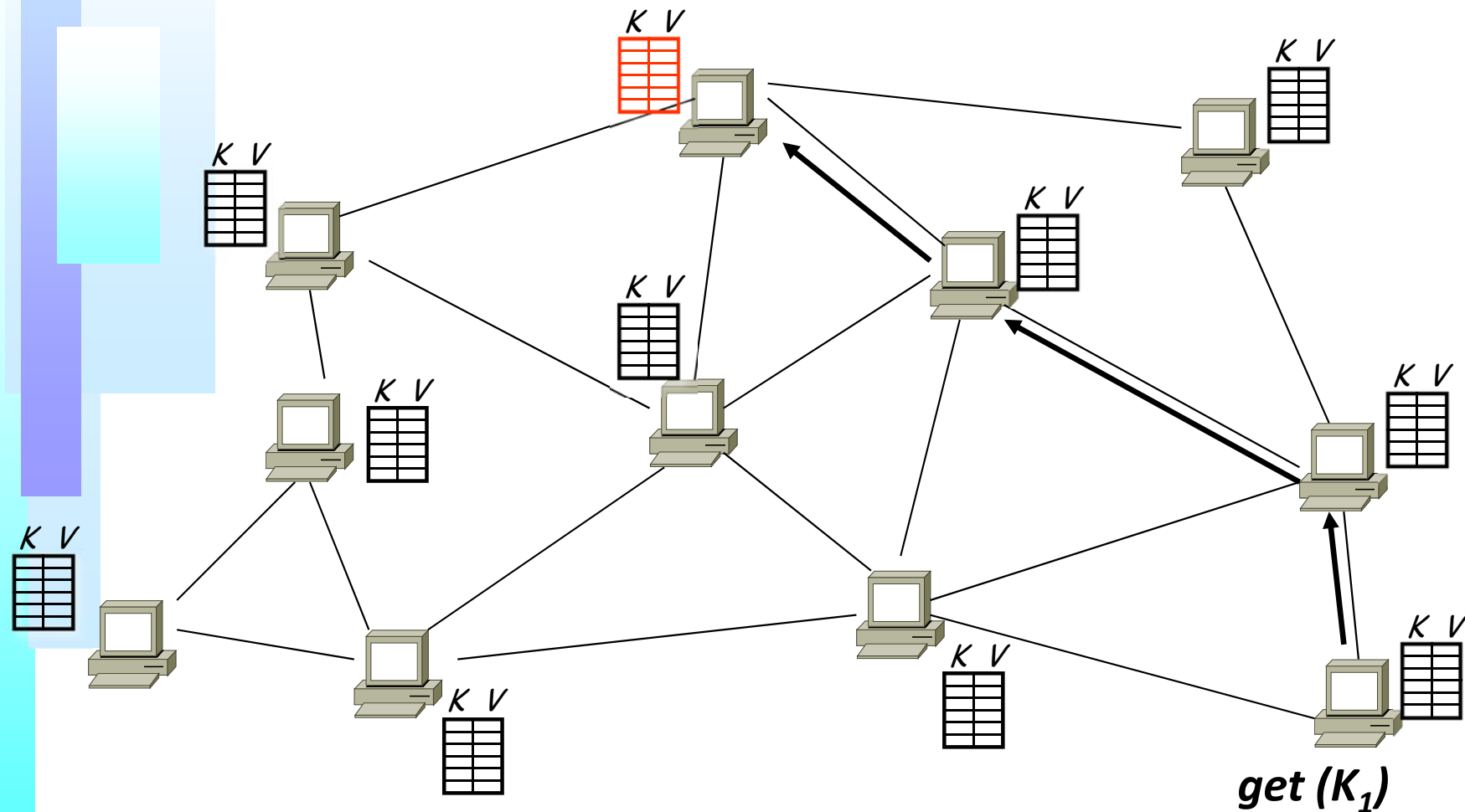
# DHT in action: put()



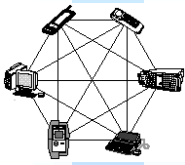
*Operation: take key as input; route msgs to node holding key*



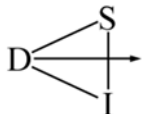
# DHT in action: get()

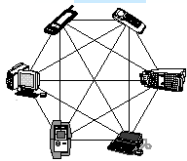


*Operation: take key as input; route msgs to node holding key*



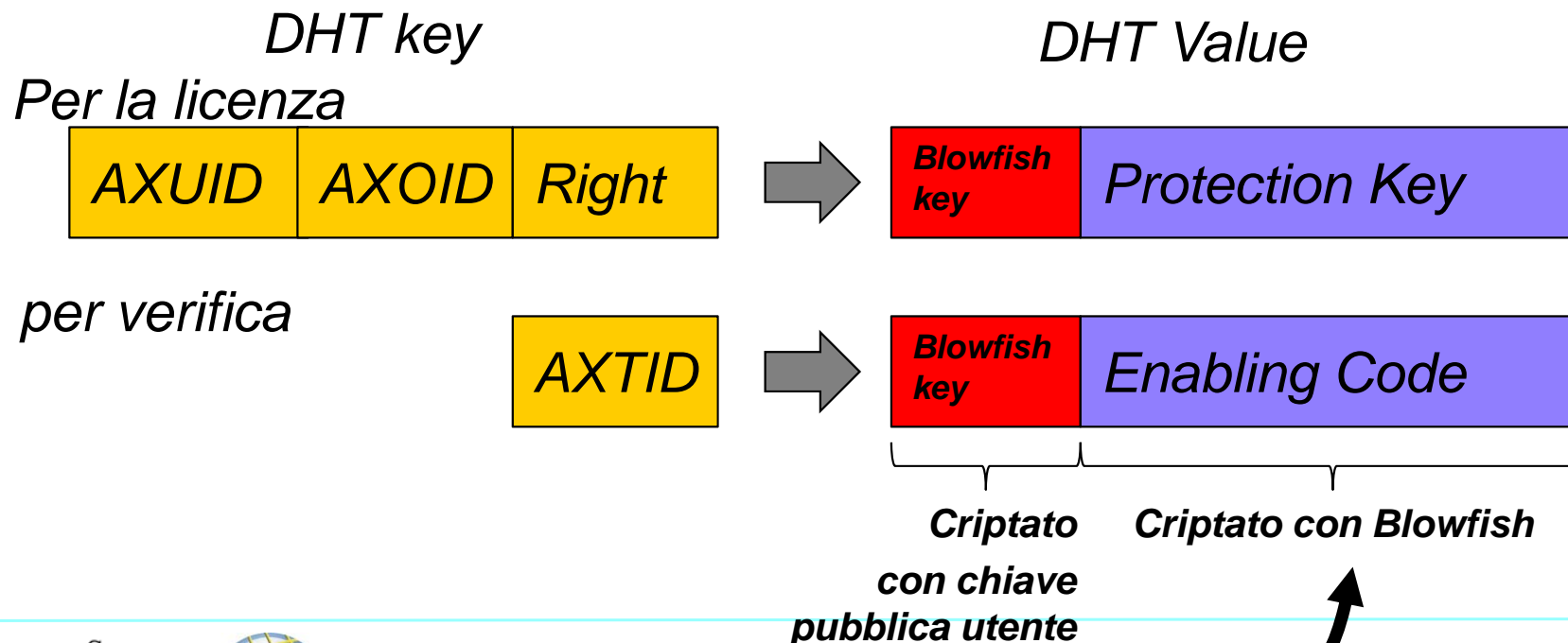
- n keyspace composto da stringhe di 160 bit
  - ♣ Al massimo di hanno u numero di hop pari al Log in base due di 160
  - ♣ Costo della ricerca:  $O(\log N)$
- Se si vuole immettere nella DHT un contenuto caratterizzato dai parametri *filename* e *data*, viene inizialmente calcolato l'hash del filename, ad esempio tramite un algoritmo SHA
- Bamboo è un sistema peer to peer che riassume alcune delle caratteristiche di Chord e Tapestry. È stato scritto in Java da Sean Rhea della UC Berkeley, ma si basa fortemente sui progetti OceanStore e Libasync, ed è open source

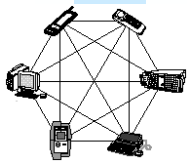




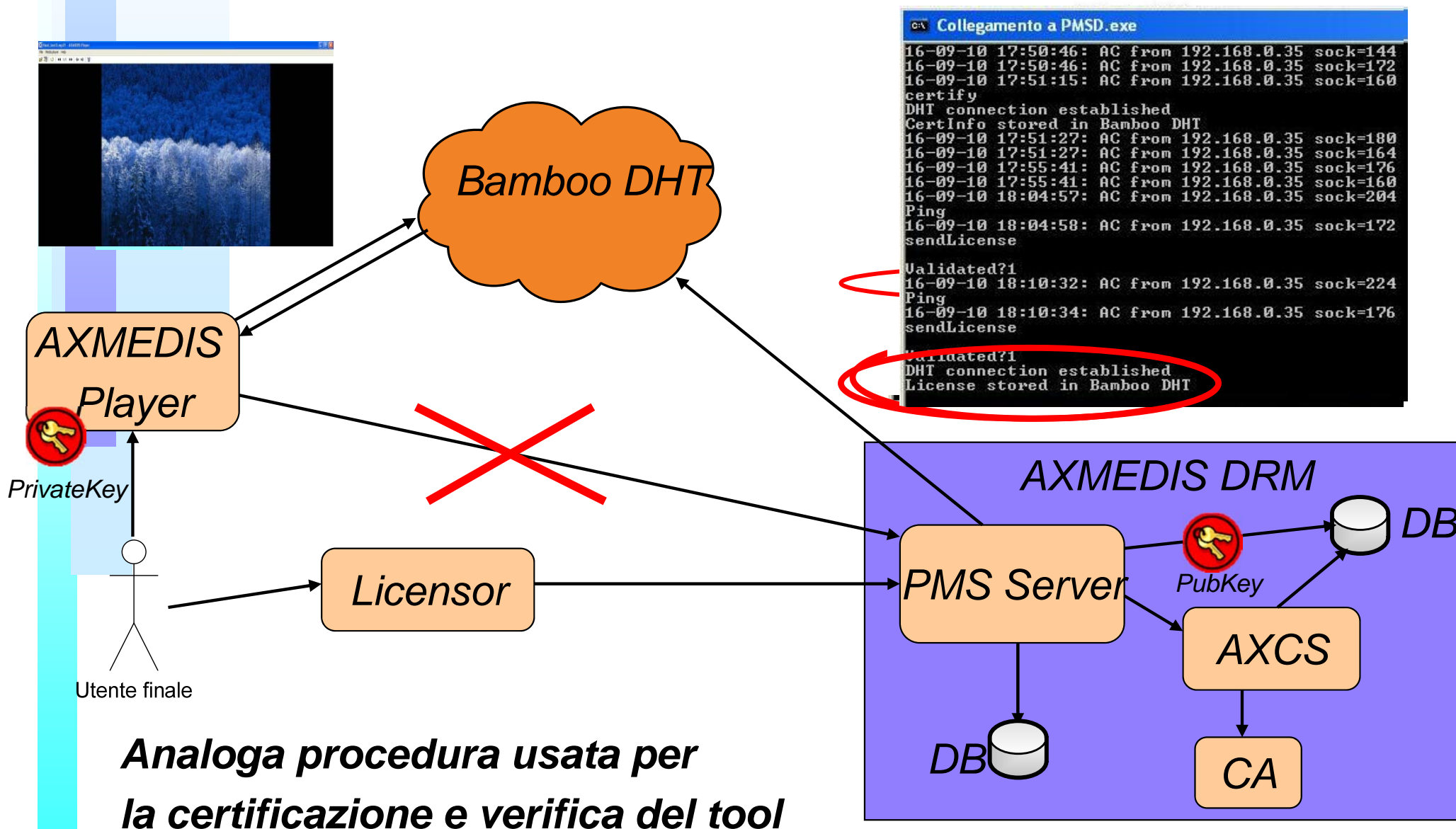
# Sicurezza dei dati

- Sicurezza ottenuta tramite meccanismi di crittazione: Blowfish e RSA.
- Si ottiene quindi un doppio sistema di cifratura, che impedisce agli utenti non autorizzati l'utilizzo delle informazioni riservate.

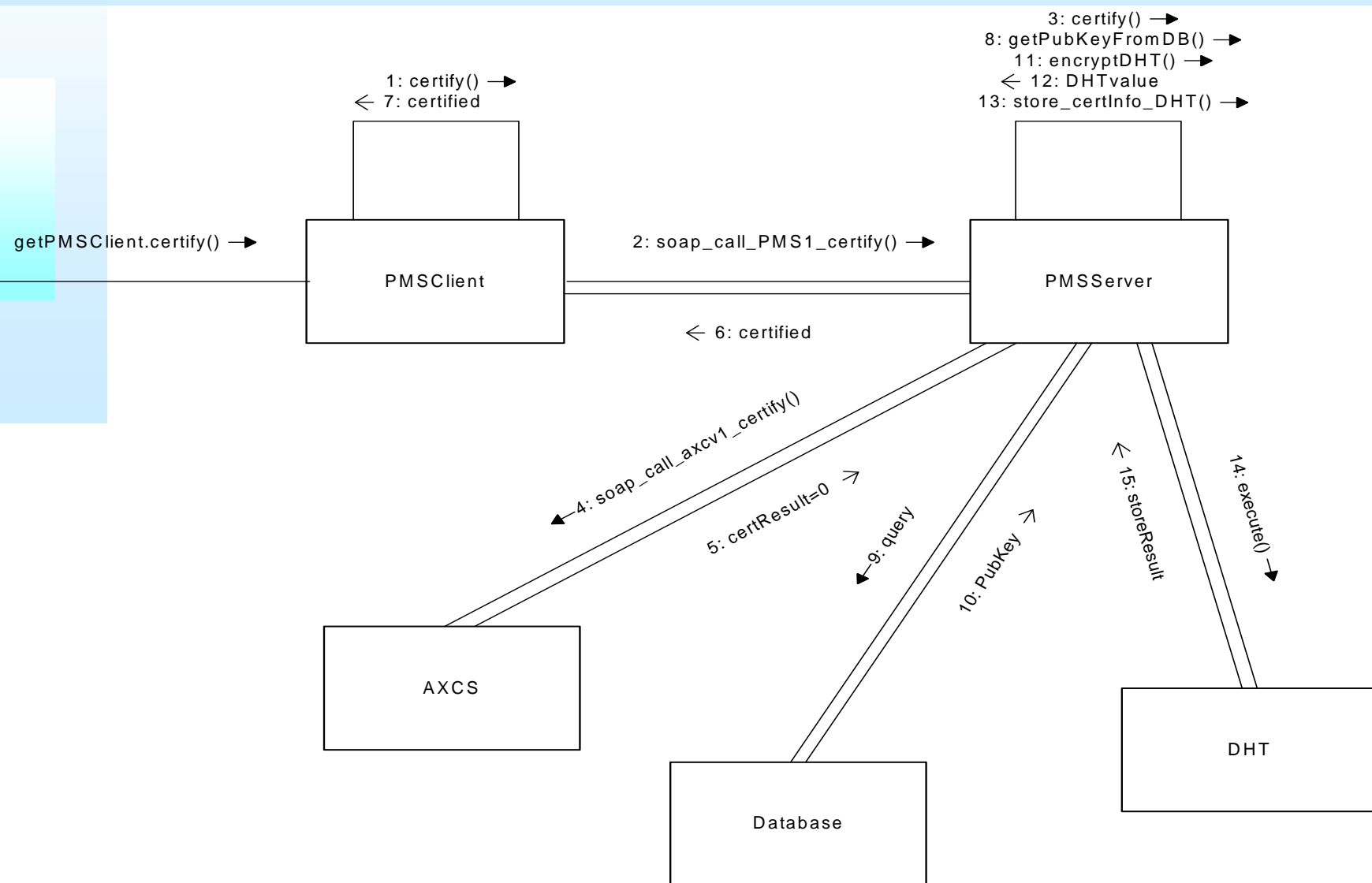
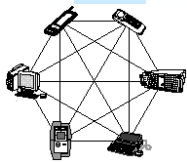




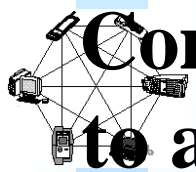
# Scenario di test (1)



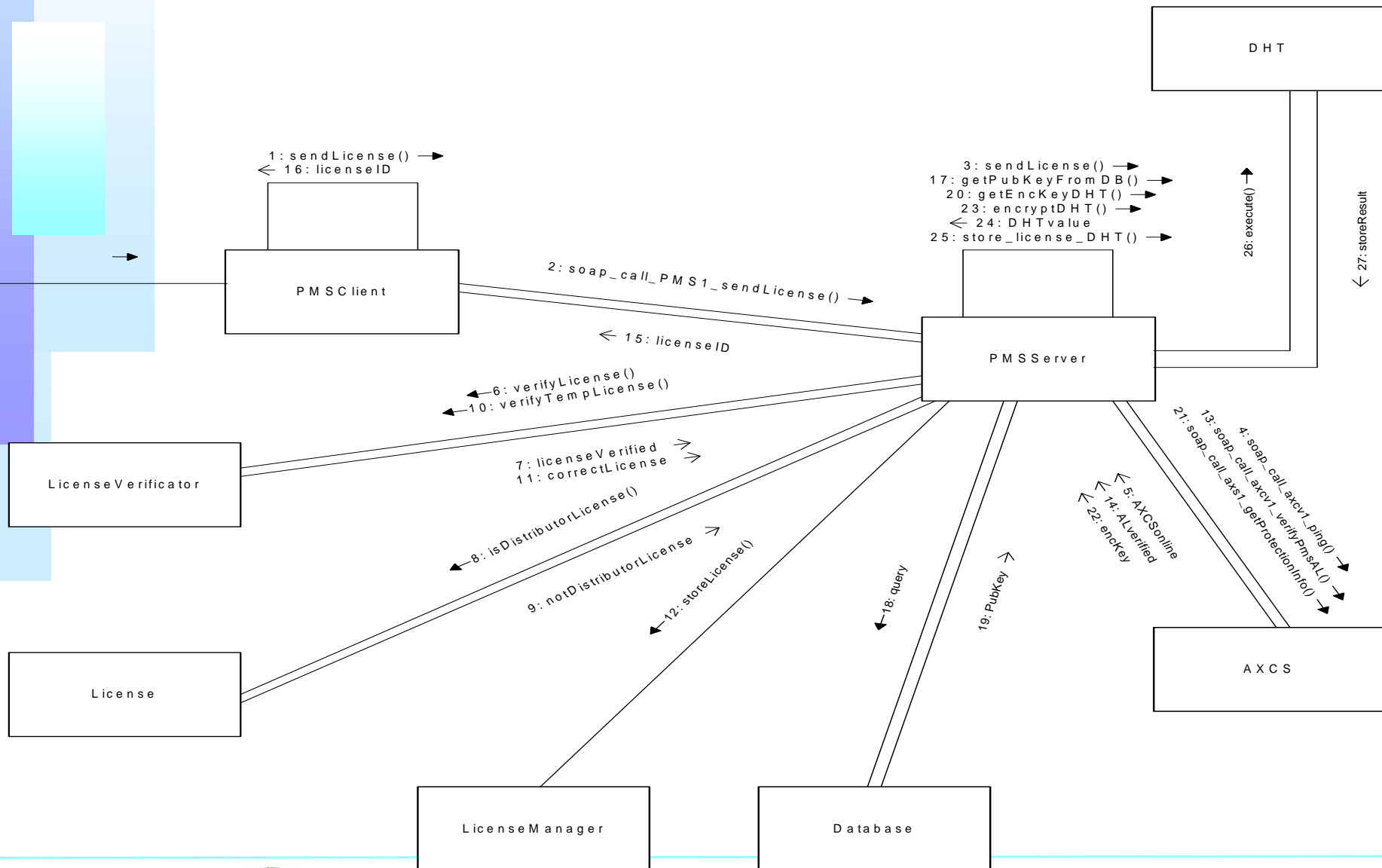
**Analoga procedura usata per la certificazione e verifica del tool**

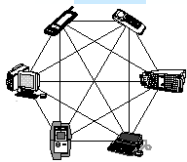




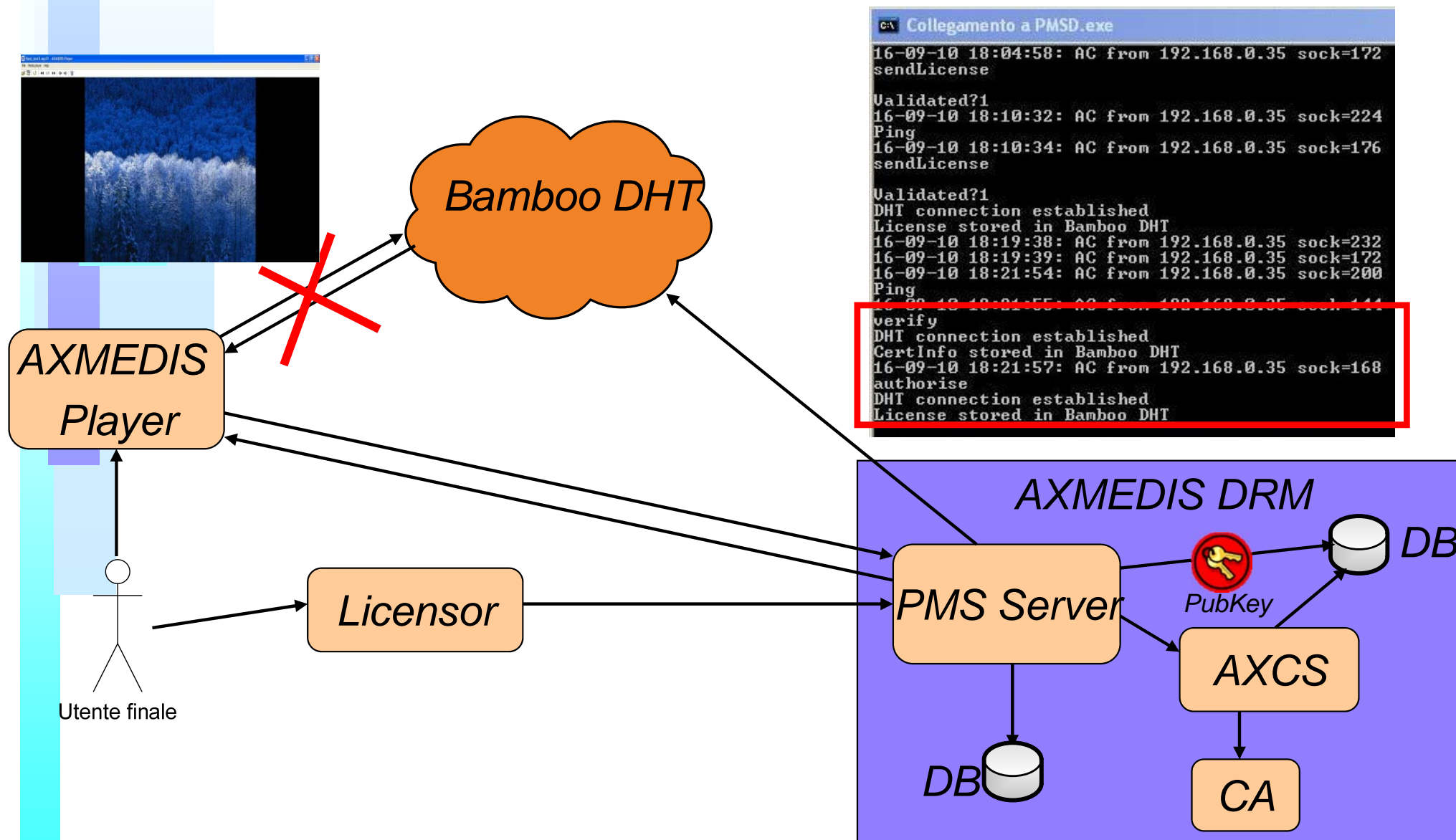


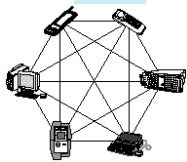
# Computing and saving protection information related to a grant authorization formalized in a license, in the case of DRM P2P.





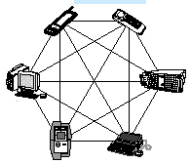
# Scenario di test (2)



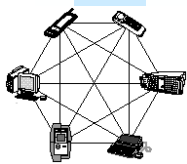


# Considerazioni su P2P DRM

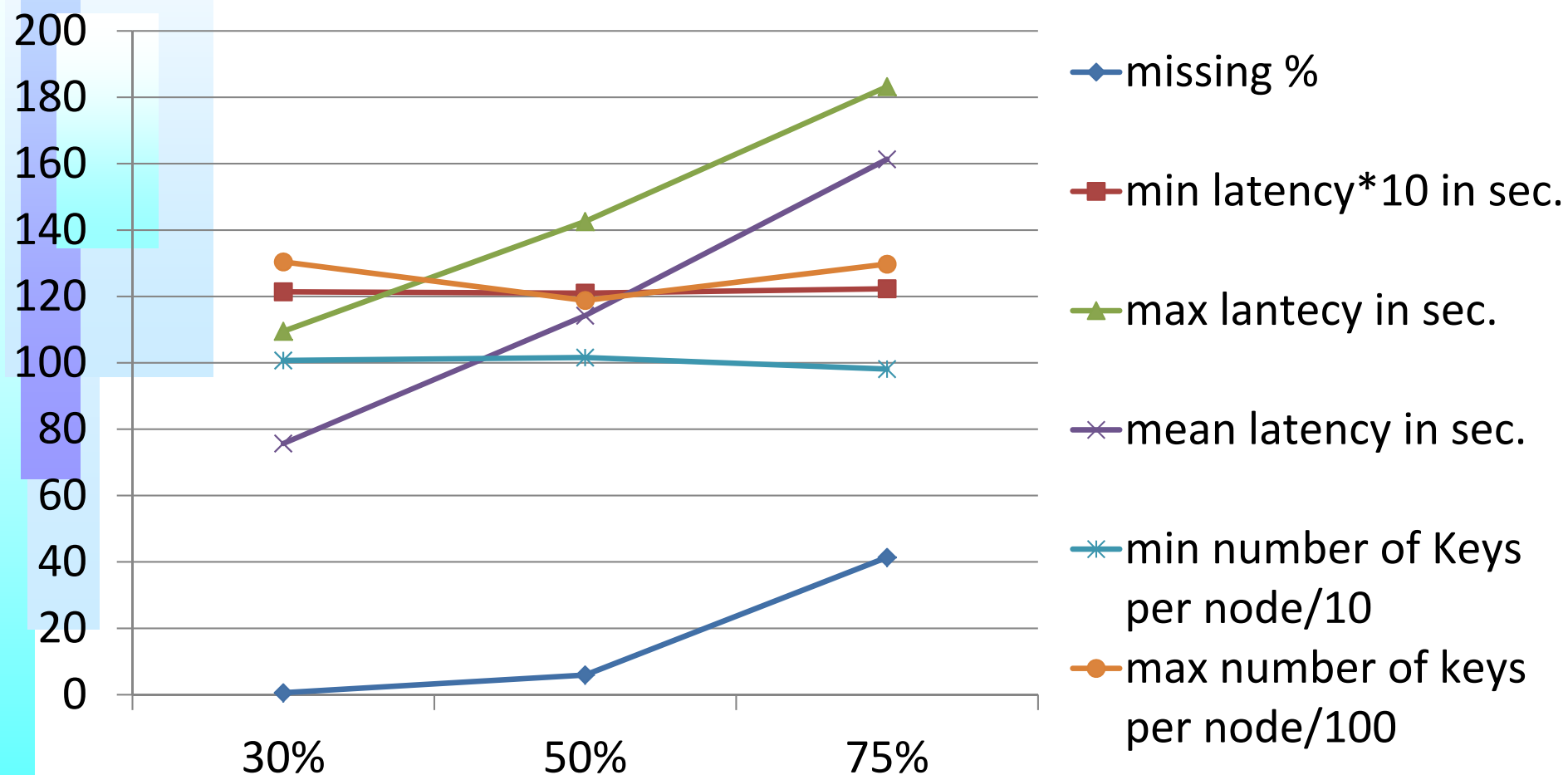
	DRM AXMEDIS TRADIZIONALE	DRM P2P
<b>SICUREZZA</b>	SI	SI
<b>TRUST MANAGEMENT</b>	SI	SI
<b>RIDONDANZA DEI DATI</b>	NO	SI
<b>SCALABILITA'</b>	NO	SI
<b>EFFICIENZA</b>	Lineare	Logaritmica
<b>Costo di storage</b>	$O(L=C*U)$	$O(L=C*U)$ + duplications on DHT
<b>Costo di accesso, transazione</b>	$O(U*\text{devices})$ , caso peggiore	$O(1) \rightarrow O(U*D)$ sul server oppure su DHT $O(\log_2 N)$
<b>FAULT TOLERANCE</b>	NO	SI

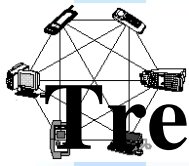


Parameter	Description	Range of values
<b>L</b>	Size of the leaf node list	8-400
<b>Ltime</b>	Time to update the leaf node list	25-250s
<b>K</b>	Number of replicas per node	4-200
<b>N</b>	Number of nodes involved in the network	1000-10000
<b>Delay</b>	Delay of transmission	30-50 ms
<b>Jitter</b>	variability over time of the packet latency across a network	10%
<b>SendQueueLength</b>	Max dimension of the output buffer	0.5 Mbyte

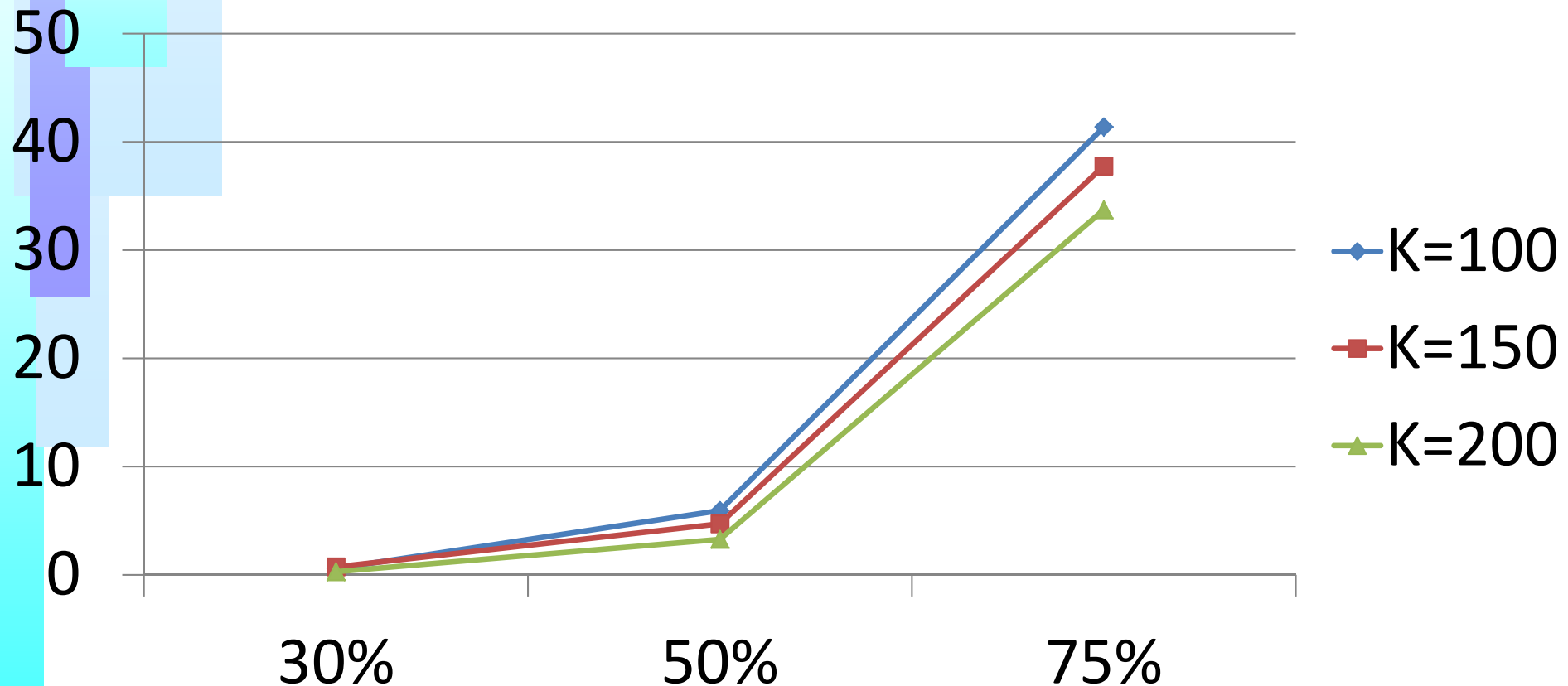


**(a)  $L=200$ ,  $K=100$ ;**

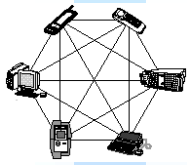




**Trend of the percentage of missing estimated on the DHT P2P for the cases of a churn of 30%, 50% and 75%, and K equal to 100, 150, and 200.**

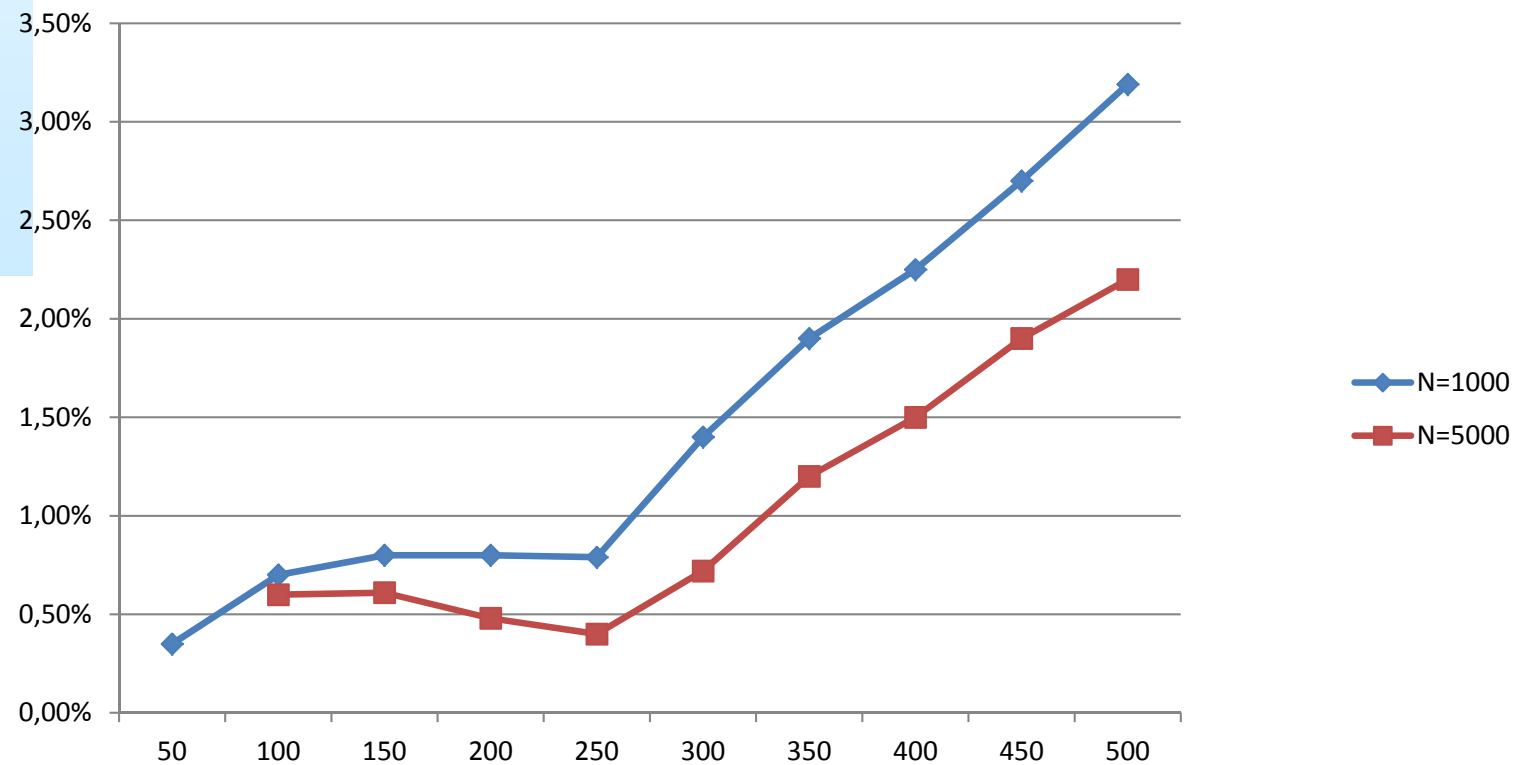






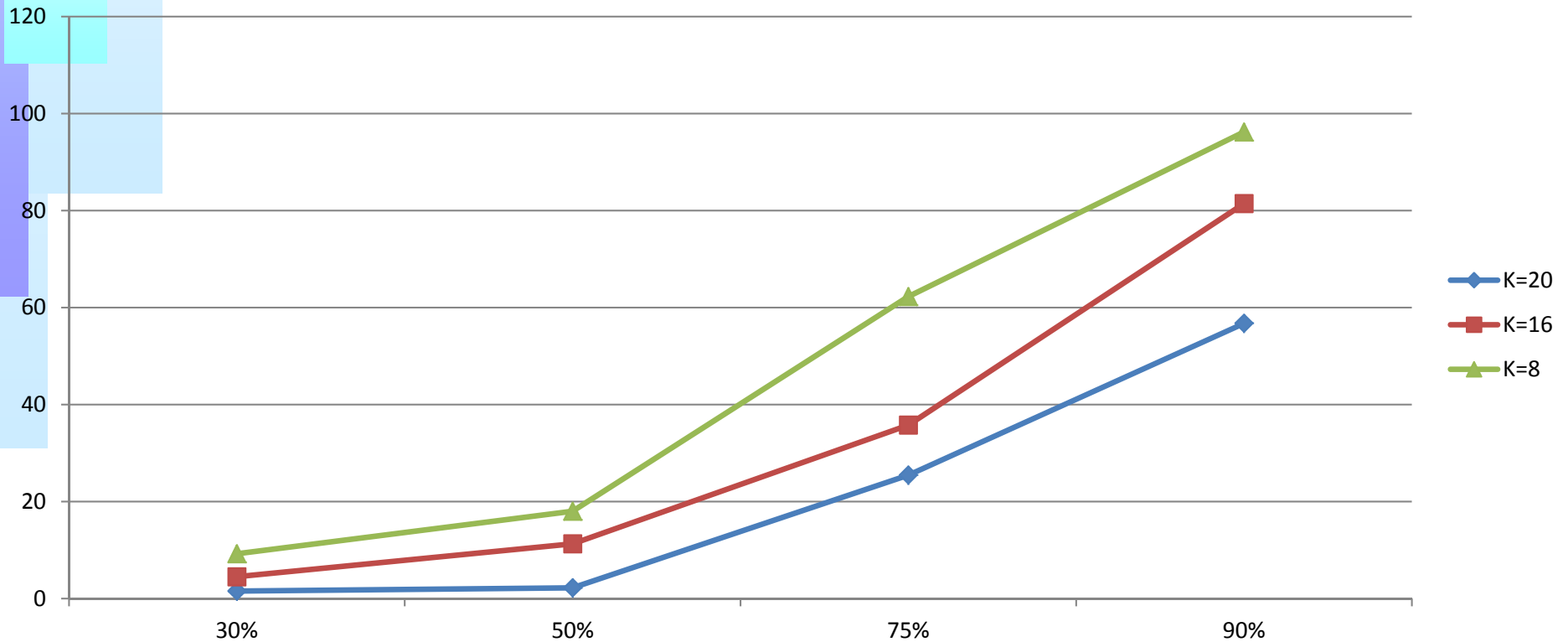
# Trend of the percentage of missing estimated on the DHT P2P

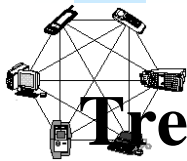
## With respect to the Ltime value for the cases of a churn of 30%, and different values of the number of nodes, N, with the same number of replica



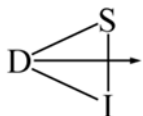
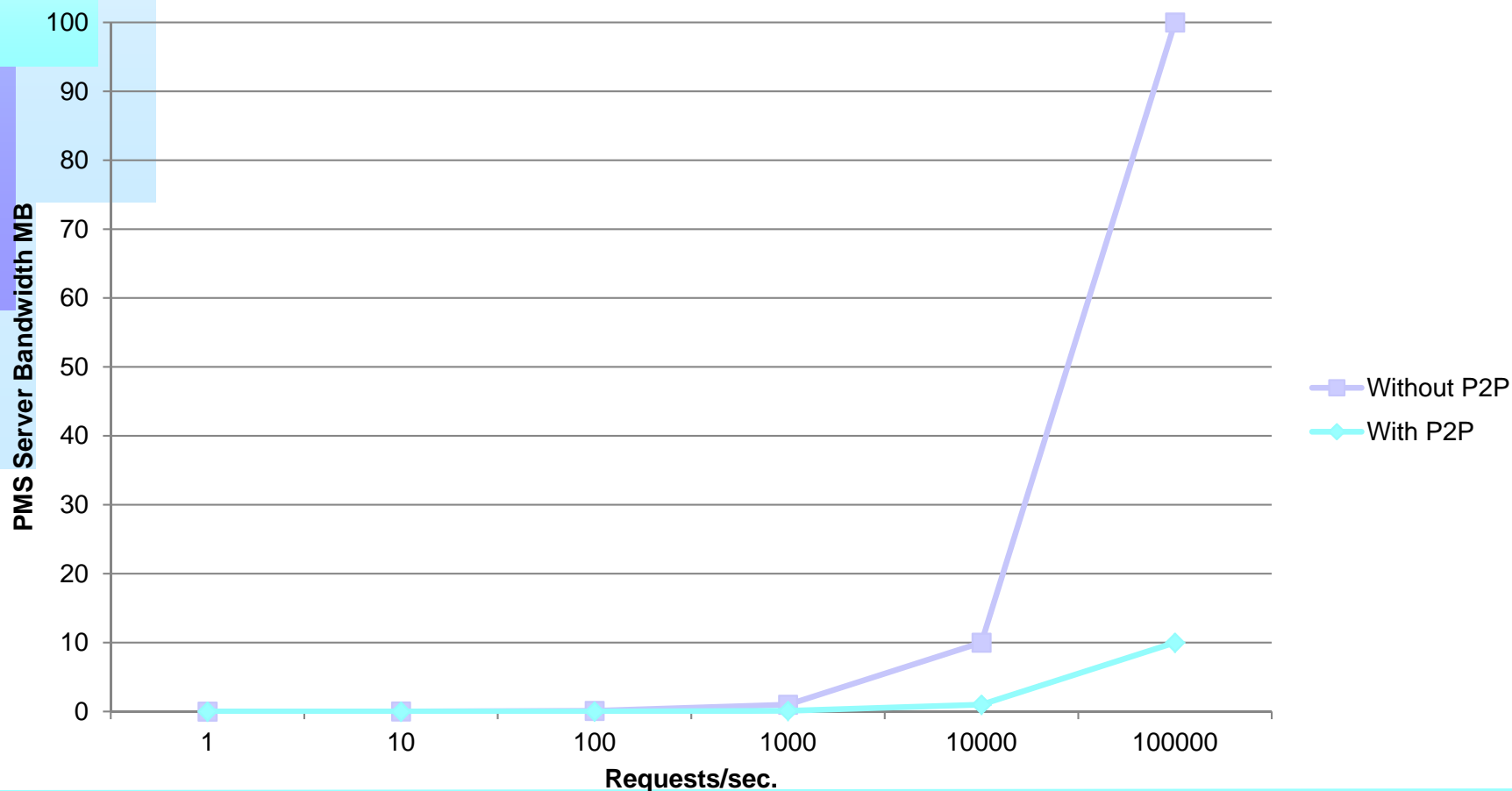


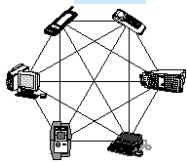
# Trend of the percentage of missing estimated on the DHT P2P with respect to different percentage of churn, for $K$ equal to 20, 16 and 8





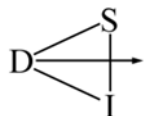
# Trend of needed bandwidth for the PMS Server with respect to the number of requests per second of verification/authentication coming from the clients both with and without P2P solution, for $K=100$ .

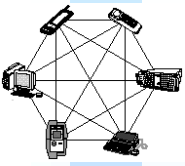




# Bibliografia

- P. Bellini, P. Nesi, F. Pazzaglia, "Exploiting P2P Scalability for Grant Authorization in Digital Rights Management Solutions", International Journal Multimedia Tools and Applications, Springer press, 2013. , DOI 10.1007/s11042-013-1468-y, Pub on line April 2013,
- Emanuele Bellini , Paolo Nesi, "[A Trust P2P network for the Access to Open Archive resources](#)", WORLD LIBRARY AND INFORMATION CONGRESS: 75TH IFLA GENERAL CONFERENCE AND COUNCIL, 23-27 August 2009, Milan, Italy, <http://www.ifla.org/annual-conference/ifla75/index.htm>
- P. Bellini, I. Bruno, D. Cenni, P. Nesi, D. Rogai, "P2P Architecture for Automated B2B Cross Media Content Distribution", [Automated Production of Cross Media Content for Multi-Channel Distribution, 2007. AXMEDIS '07. Third International Conference on](#), AXMEDIS 2007, IEEE press, 28-30 Nov. 2007 Page(s):105 - 112, Digital Object Identifier 10.1109/AXMEDIS.2007.31





# Bibliography

- Colouris Book come menzionato nelle prime slide del corso
- progetto Jxta ( <http://www.jxta.org> )
- progetto MyJxta2 ( <http://myjxta2.jxta.org> )
- Bernard Traversa, *Project JXTA 2.0 Super-Peer Virtual Network*, Maggio 2003.
- AXMEDIS P2P solution extending BitTorrent  
♣ [www.axmedis.org](http://www.axmedis.org)

