

Paradigma P2P

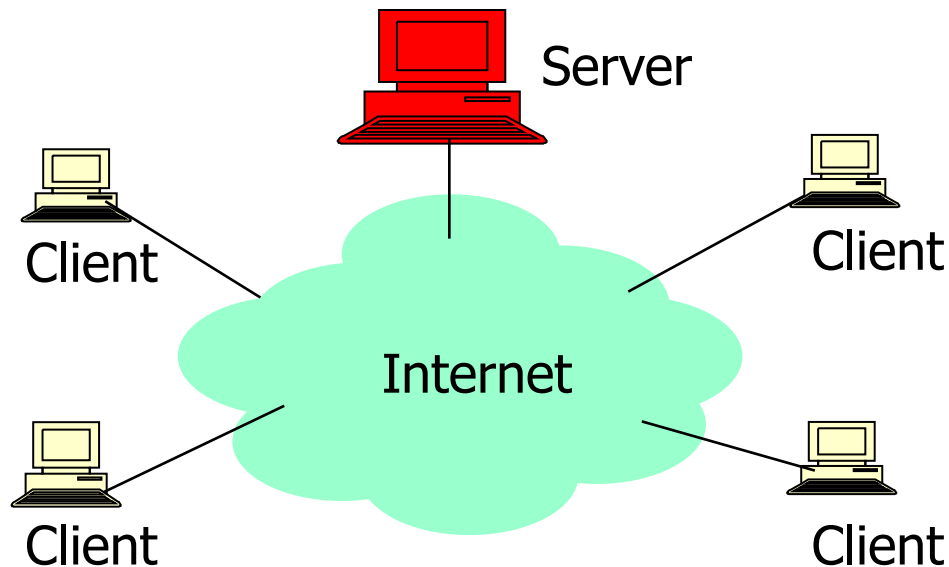
Lenuta Alboaie
adria@infoiasi.ro

Cuprins

- **Paradigma peer-to-peer(P2P)**
 - Preliminarii
 - Definitii
 - Caracterizare
 - Tipuri de aplicatii
 - Infrastructuri
 - Aplicatii

Preliminarii

...sa ne reamintim **modelul client/server**



Preliminarii

...sa ne reamintim **modelul client/server**

- Uzual, privim clientul ca fiind o componenta avind capacitati computationale reduse
- Serverul este mentinut si administrat in mod centralizat

Probleme ale arhitecturii client/server:

- Lipsa robustetii
- Lipsa sigurantei
- Lipsa scalabilitatii
- Vulnerabilitate la atac

Definitii

Peer = *one that is of equal standing with another*
(conform Webster)

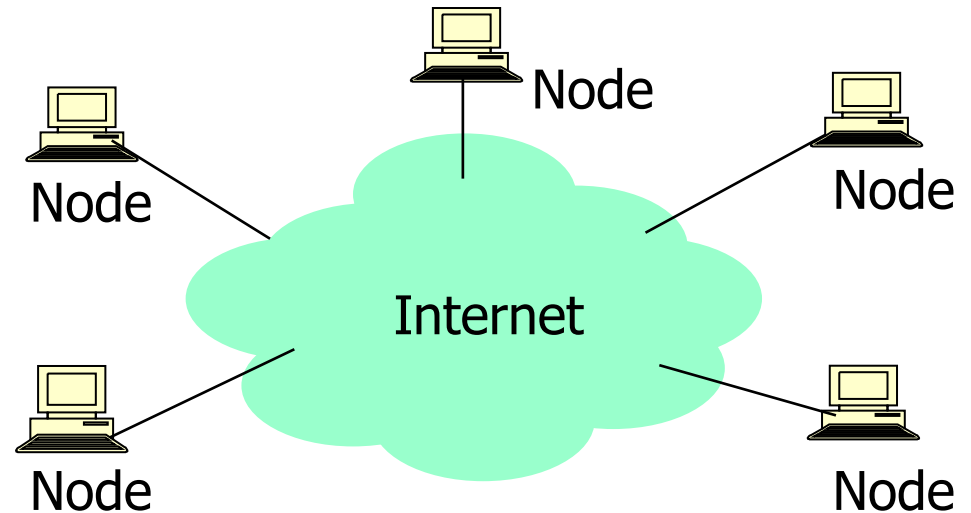
Peer-to-peer (P2P) = arhitectura de retea in care nodurile sunt relativ egale

- In sensul ca fiecare nod este in principiu capabil sa realizeze functii specifice retelei
- In practica, multe dintre nodurile retelei pot realiza asemenea functii

Definitii

Sistemele P2P, in sens strict, sunt sisteme complet distribuite

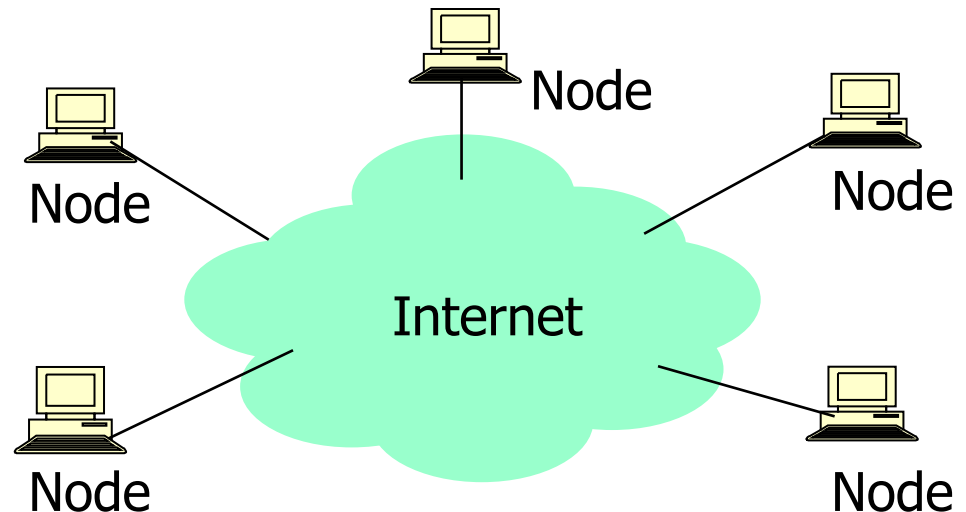
- Toate nodurile sunt total echivalente, in termeni de functionalitate si a activitatilor pe care le pot desfasura



Obs. **Sisteme P2P pure**, sunt rare (e.g. Gnutella), majoritatea sunt hibride avind supernoduri sau servere cu diferite roluri (cautare de date, control)

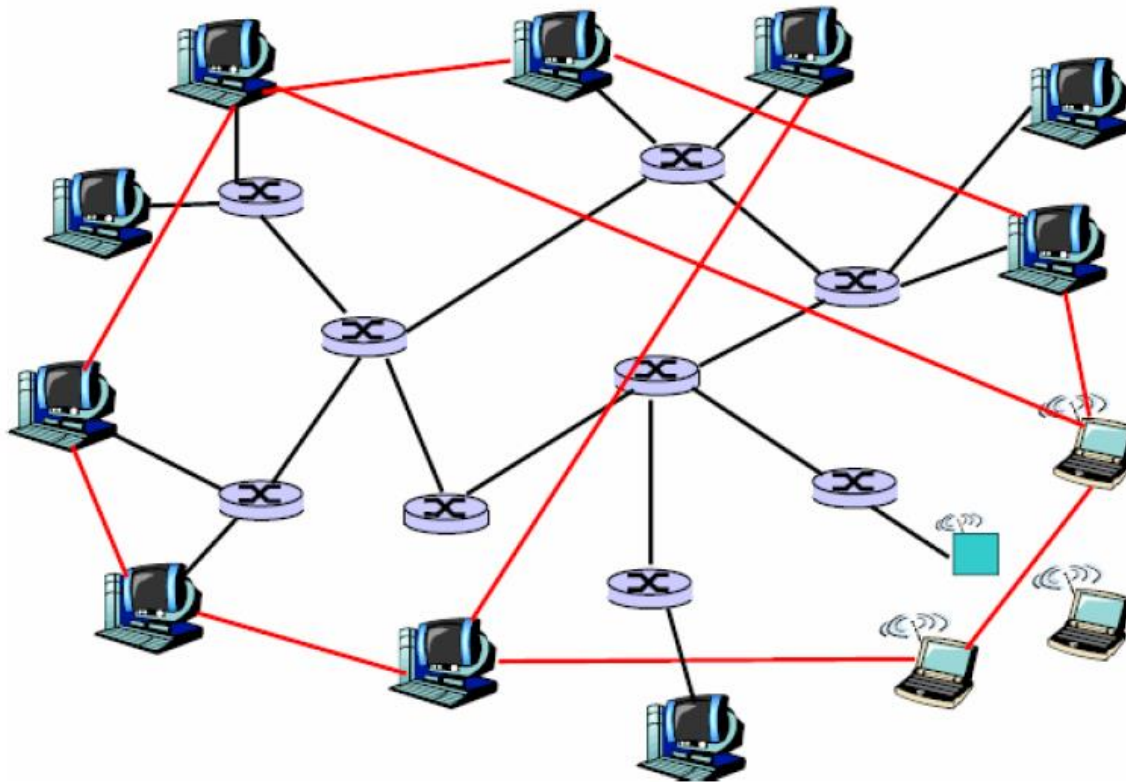
Definitii

- Nodurile
 - Pot consuma si oferi date
 - Orice nod poate initia o conexiune
- Nu exista sursa de date centralizata =>
 - Forma de democratie pe Internet
 - Protectie *copy-right* amenintata



Definitii

- “**P2P**” este clasa de aplicatii care se bazeaza pe resursele (de stocare, de procesare, continut, prezente umane) disponibile la marginile (edges) Internet-ului



*Edges of the Internet
(overlay networks)*

Definitii

“**P2P** is a class of applications that take advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet. Because accessing these decentralized resources means operating in an environment of unstable and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant, or total autonomy from central servers”

“A distributed network architecture may be called a **P2P** network if the participants share a part of their own resources. These shared resources are necessary to provide the service offered by the network. The participants of such a network are both resource providers and resource consumers”

Caracterizare

Caracteristici definitorii:

- Partajarea resurselor computationale prin interschimb **direct** si mai putin prin intermediari oferite de o autoritate centralizata (server)
 - Serverele centralizate pot fi folosite insa pentru a realiza activitati specifice (initializarea retelei P2P, adaugarea de noi noduri in retea,...)
 - Ideal, nodurile participa activ si unilateral la realizarea de operatii ca localizarea & caching-ul nodurilor/continutului, dirijarea informatiilor, managementul resurselor transferate etc.

Caracterizare

Caracteristici definitorii:

- Abilitatea de a trata instabilitatea si variatiile conctivitatii retelei, **adaptindu-se automat** la erorile survenite sau la dinamicitatea nodurilor
 - Topologia retelei P2P e adaptiva si toleranta la defecte, nodurile auto-organizindu-se in vederea mentinerii conectivitatii si performantei retelei

Caracterizare

Reteaua P2P este una suprapusa (*overlay*) peste cea fizica

- Se situeaza la nivelul aplicatie => flexibilitate
- Muchiile virtuale sunt conexiuni TCP sau pointeri la adrese IP
- Mentinerea rețelei P2P se face prin verificarea periodica a conectivitatii (*ping*) ori a existentei (mesaje “*still alive?*”)
- Cand un nod pica, sistemul P2P ar putea stabili noi muchii
- Proximitatea (fizica) a nodurilor nu e importanta
- Reteaua P2P poate fi structurata sau nu

Scopuri si beneficii

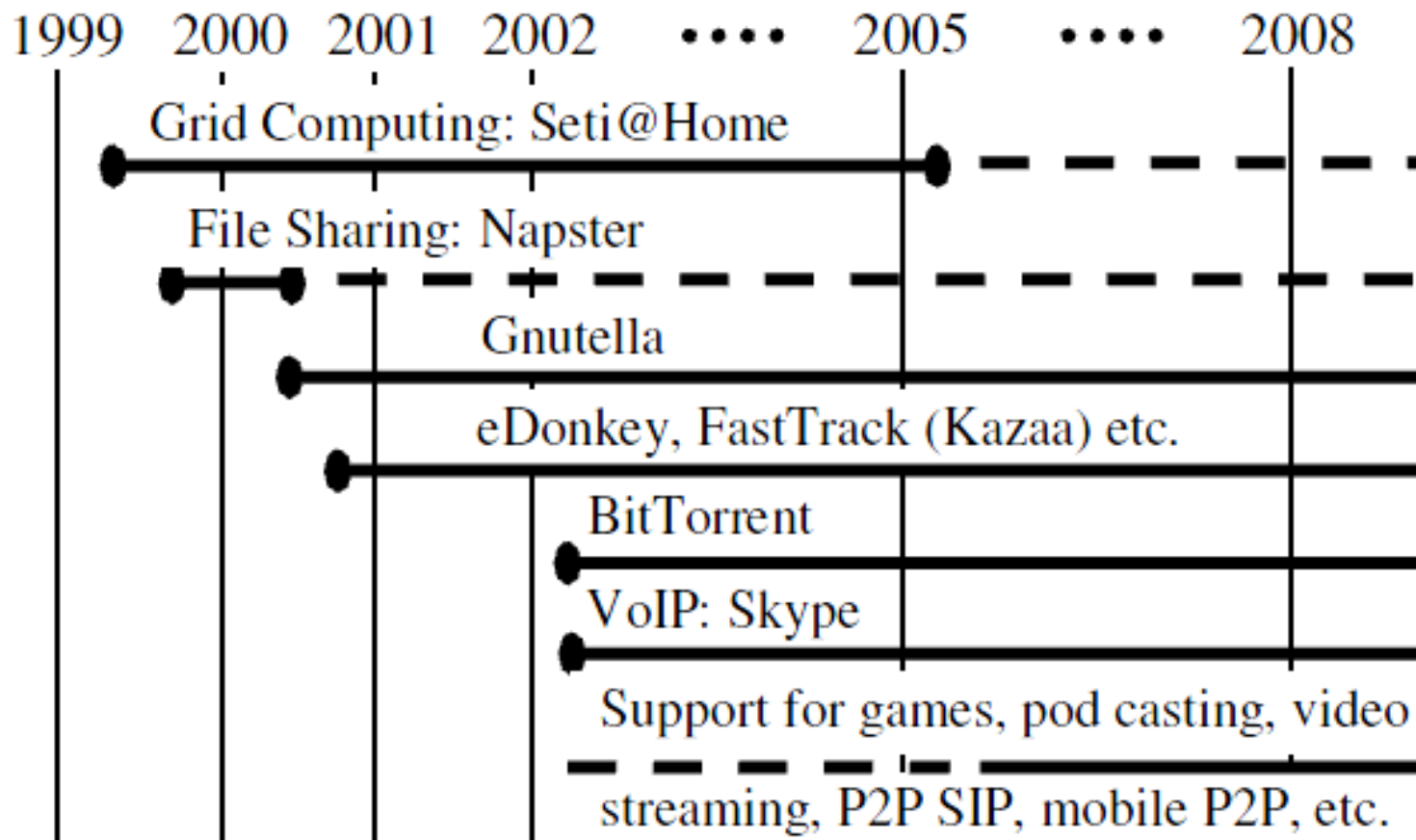
- **Utilizarea eficienta a resurselor**
 - Latimea de banda neutilizata, resurse de stocare, putere de procesare disponibile la marginile (*edges*) rețelei
- **Scalabilitate**
 - Fara informatii centralizate, fara *bottleneck* –uri (de comunicare si de calcul)
 - Agregarea resurselor se face in mod natural odata cu utilizarea sistemului
- **Siguranta** (eng. Reliability)
 - Existenta de copii a datelor
 - Distribuie geografica
 - Nu mai exista “*single point of failure*”
- **Administrare usoara**
 - Nodurile se auto-organizeaza
 - Cresterea tolerantei la erori si a echilibrarii incarcarii
 - Cresterea autonomiei
- **Anonimitatea**
 - Greu de realizat intr-un sistem centralizat
- **Dinamism**
 - Mediu dinamic
 - Colaborare si comunicare ad-hoc

Dezavantaje/Probleme

- Arhitecturile P2P sunt probabilistice
 - Localizare impredictibila a resurselor
 - Resursele sunt volatile
- Inexistenta unui control centralizat
 - Probleme privind impunerea unei autoritati asupra aplicatiilor, continutului si utilizatorilor
 - Dificultati in detectarea si identificarea utilizatorilor (aspecte anti-sociale)
- Folosirea in mod abuziv a unor resurse/dispozitive
- Incurajarea folosirii sistemelor P2P in scop abuziv si ilegal (drepturile de autor asupra continutului digital)
- Lipsa increderii la nivel comercial, de afaceri
- Probleme de securitate (cursul viitor)

Example...

Timeline of Popular Peer-to-Peer Protocols



Tipuri de aplicatii

- **Comunicare & colaborare**

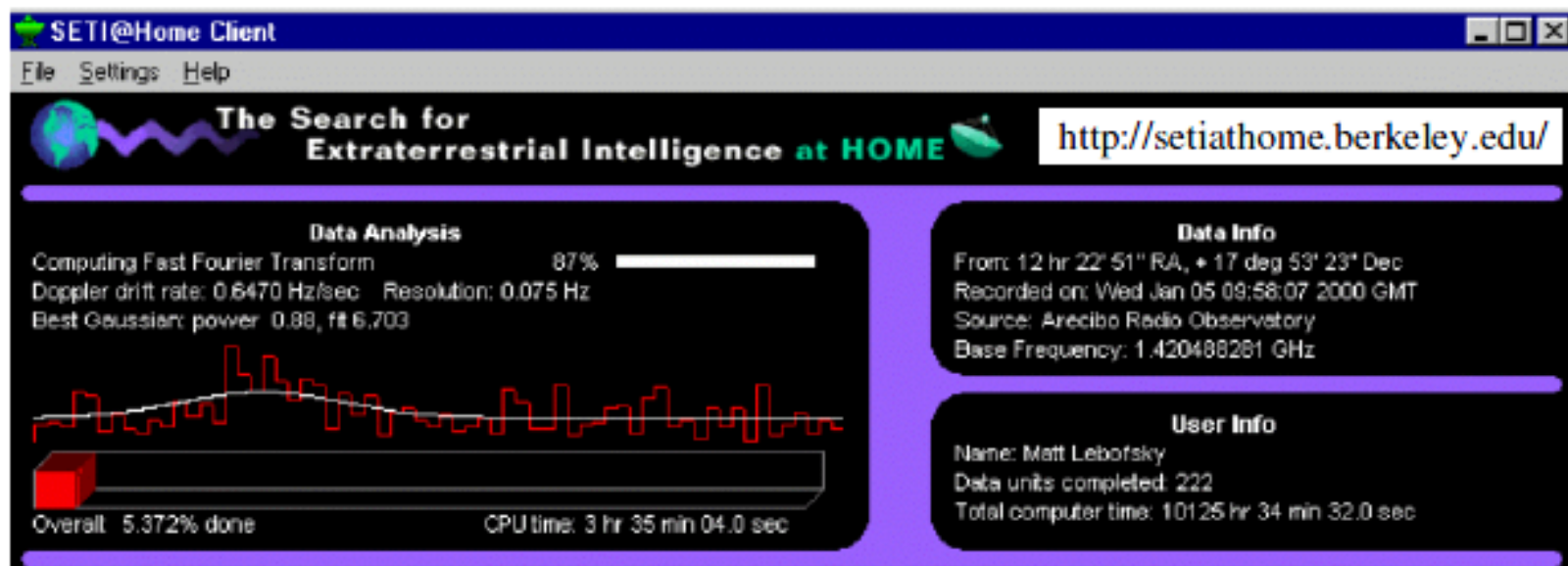
- Sisteme ce ofera o infrastruktura pentru facilitarea comunicarii & colaborarii directe, deseori in timp real, intre noduri
 - Sisteme conversationale (chat, mesagerie instantanee):
IRC (Internet Relay Chat), ICQ (1996), YM!, MSN Messenger, Skype

- **Calcul distribuit**

- Sisteme ce folosesc puterea computationala a nodurilor disponibile (cicli de procesor)
 - Rezolvarea unor probleme prin divide-et-impera:
SETI@home (*Search for Extra-Terrestrial Intelligence*-Berkeley), genome@home
 - Reteaua P2P reprezinta un gen de Grid computational (...curs viitor)

Tipuri de aplicatii| Calcul distribuit - Exemplu

SETI@Home: A Public-Resource Computing Experiment



- ❑ Radio telescope signal analysis has insatiable appetite for computing power
- ❑ Usage of computers in homes and offices around the world has provided unprecedented computing power
- ❑ Grid computing application via peer-to-peer approach under central control

Tipuri de aplicatii

- **Suport pentru serviciile Internet**

- Sisteme multicast P2P (e.g. Cirrus – Adobe Flash <http://labs.adobe.com/technologies/cirrus/>), aplicatii de securitate (impotriva atacurilor DoS sau a virusilor) etc. -> (cursul viitor)

- **Sisteme de stocare (baze de date)**

- Proiectare de sisteme de baze de date distribuite bazate pe infrastructuri P2P
 - Modelul LRM (*Local Relational Model*)
 - PIER – motor scalabil de interogare distribuita
 - Edutella – proiect open-source pentru interogari si stocare de *meta-date* (P2P pentru Semantic Web)

Tipuri de aplicatii

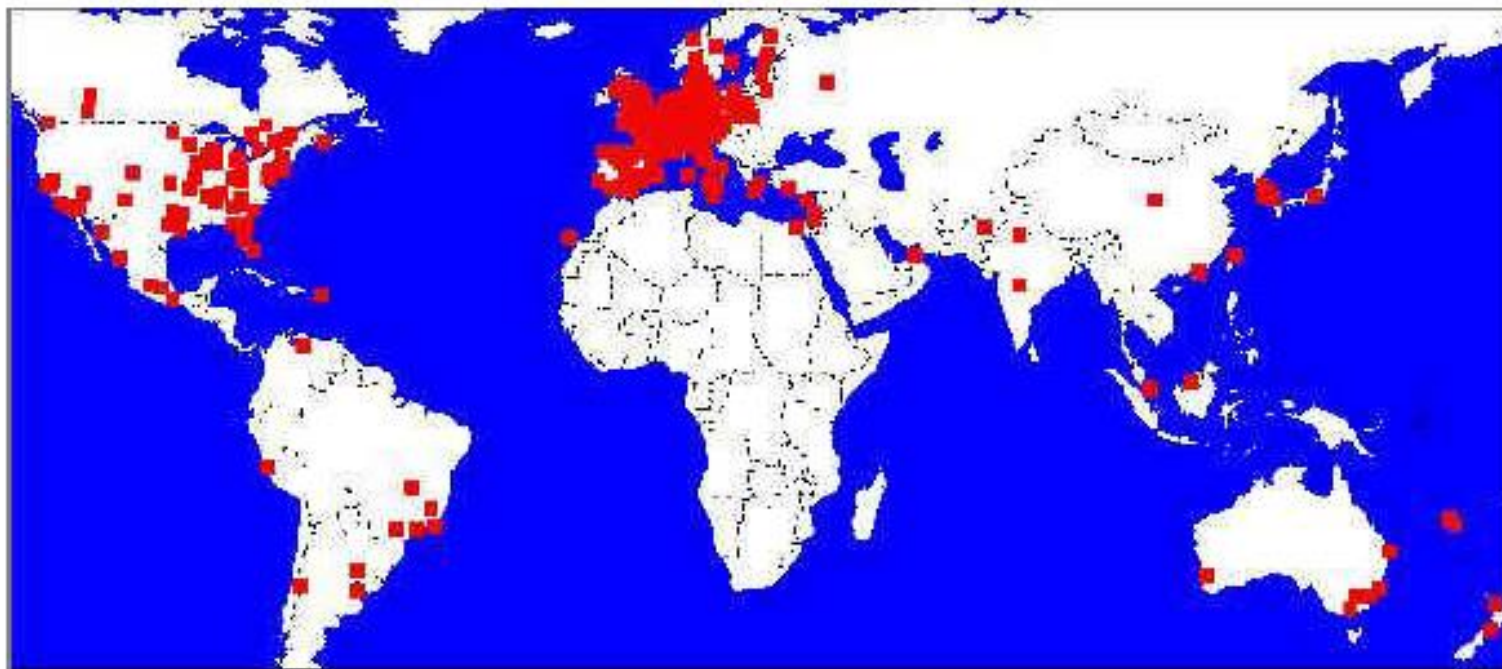
- **Distribuire de continut digital**

- Sisteme & infrastructuri pentru partajarea resurselor digitale (multimedia si alte date) intre utilizatori
 - Aplicatii pentru partajarea fisierelor (e.g. Napster, Gnutella, KaZaA, Freenet, BitTorrent, eDonkey etc.)
 - Medii de stocare distribuita pentru publicarea, organizarea, indexarea, cautarea si regasirea datelor in maniera securizata & eficienta
(PAST, Chord, Groove, Mnemosyne, Avalanche,...)

Tipuri de aplicatii

- Distribuire de continut digital | Exemplu

P2P File-Sharing: Fast distribution of large files



Example: Harry Potter III early propagation after 2 hours on May 28th 2004 (Source: www.itic.ca/DIC/News/archive.html)

Tipuri de aplicatii

- **Distribuire de continut prin P2P**
 - Sisteme P2P de “interschimb de fisiere”
 - Nodurile transfera un fisier la un moment dat
 - Se ofera facilitati pentru realizarea unei retele P2P si pentru cautarea&transferul de fisiere intre noduri
 - Nu se ofera suport pentru securitate, disponibilitate si persistenta
 - Exemple: Napster, KaZaA, Gnutella

Tipuri de aplicatii

- **Distribuire de continut prin P2P**
 - Sisteme P2P pentru publicarea & stocarea continutului
 - Utilizatorii pot publica, stoca si distribui continut digital, pe baza unor drepturi de acces (privilegii)
 - Se focalizeaza asupra securitatii si persistentei
 - Unele ofera si facilitati privind colaborarea intre utilizatori
 - Exemple: Scan, Groove, Freenet, MojoNation, Tangler

Tipuri de aplicatii

- **Distribuire de continut prin P2P**
 - **Infrastructuri** pentru:
 - **Dirijare & Localizare:**
Chord, Can, Pastry, Tapestry, Kademila
 - **Anonimitate:**
Onion Routing, ZeroKnowledge, Freedom, Tarzan
 - **Managementul reputatiei:**
Eigentrust, PeerTrust

Infrastruc.(localizare & dirijare)

- Mecanismele de localizare si dirijare ce pot fi adoptate depind de:
 - Topologia
 - Structura
 - Gradul de centralizare
- ale rețelei suprapuse, acoperitoare (*overlay network*)

Infrastruc.(localizare & dirijare)

- **Aspecte privind centralizarea**
 - **Arhitecturi pur descentralizate**: toate nodurile realizeaza exact aceleasi activitati, jucind simultan roluri de servere si clienti, fara a beneficia de o coordonare centrala
 - Nodurile se numesc **servents** (SERVents + clieENTS)

Infrastruc.(localizare & dirijare)

- **Aspecte privind centralizarea**
 - **Arhitecturi partial centralizate**: unele noduri au un rol mai important (e.g., stocind indecsi locali pentru fisierele partajate)
 - Nodurile devin **supernoduri** conform politicilor fiecarui sistem P2P
 - Rolul de supernod este stabilit dinamic
 - **Arhitecturi descentralizate hibride**: exista un server central facilitind interactiunea intre noduri, mentinind cataloage de meta-date ale fisiereilor
 - Serverele pot identifica si verifica nodurile de stocare
 - Sistemele se mai numesc *broker mediated*

Infrastruc.(localizare & dirijare)

- **Aspecte privind structura rețelei:**
 - **Nestructurata:** plasarea conținutului este complet independentă de topologia rețelei suprapuse
 - Conținutul trebuie localizat
 - Strategii de căutare prin “*forta bruta*”: inundarea rețelei – cereri propagate via BFS/DFS
 - Strategii mai sofisticate: drumuri aleatorii, probabilistice etc
 - **Slab structurata** (*loosely structured*): deși localizarea conținutului nu e complet specificată, aceasta este afectată de dirijare
 - Categorie aflată între rețele structurate și cele nestructurate

Infrastruc.(localizare & dirijare)

- **Aspecte privind structura rețelei:**
 - **Structurata:** topologia este controlata, iar fisierele (sau pointerii la ele) sunt plasate in locatii precise
 - Se realizeaza o asociere (*mapping*) intre continut (identificatorul de fisier) si locatie (adresa nodului)
 - In genul unei tabele de rutare distribuita
 - Cautarile exacte (*exact-match queries*) pot fi realizate in mod scalabil
 - Structura folosita la dirijarea eficienta a mesajelor este dificil de mentinut in cazul unor noduri tranziente, cu rata mare de atasare si deconectare de la retea

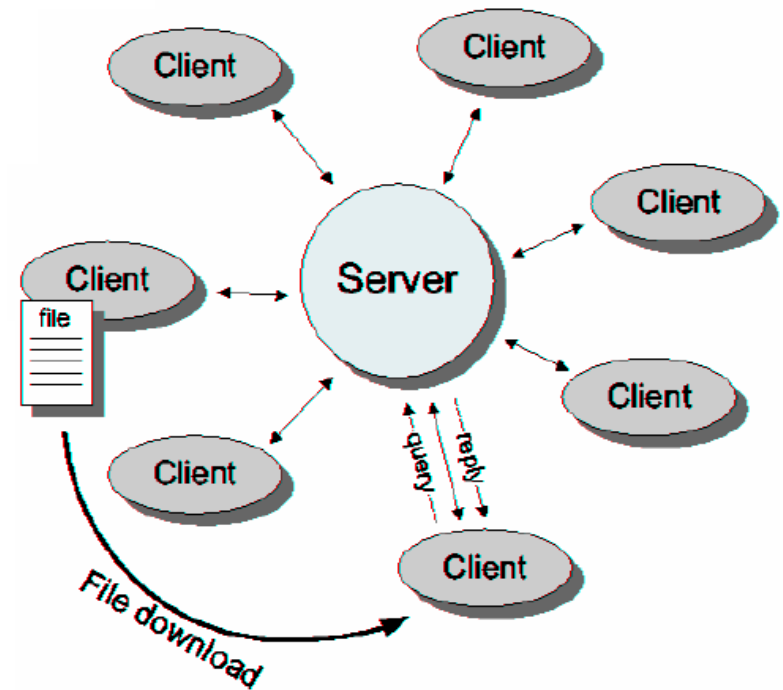
Infrastruc.(localizare & dirijare)

	Centralizare		
	Hibridă	Partială	Absentă
Nestructurată	Napster Publius	KaZaA Morpheus Gnutella Edutella	Gnutella FreeHaven
Infrastruc. structurată			Chord, CAN Tapestry, Pastry
Sisteme structurate			OceanStore Scan, PAST Kademlia Tarzan

Arhitecturi nestructurate

Descentralizate hibride

- Fiecare calculator client stocheaza continut (fisiere) partajat(e)
- Serverul central mentine o tabela cu conexiunile utilizatorilor inregistrati (IP, latime de banda,...) + o tabela cu lista fisierelor fiecarui utilizator & meta-date
- Exemple: **Napster, Publius**



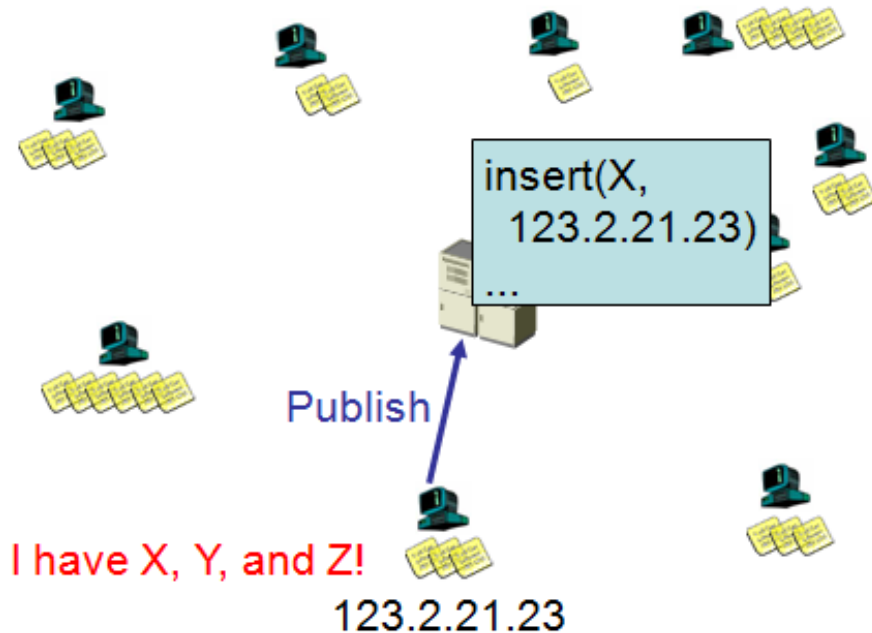
Arhitecturi nestructurate

Napster

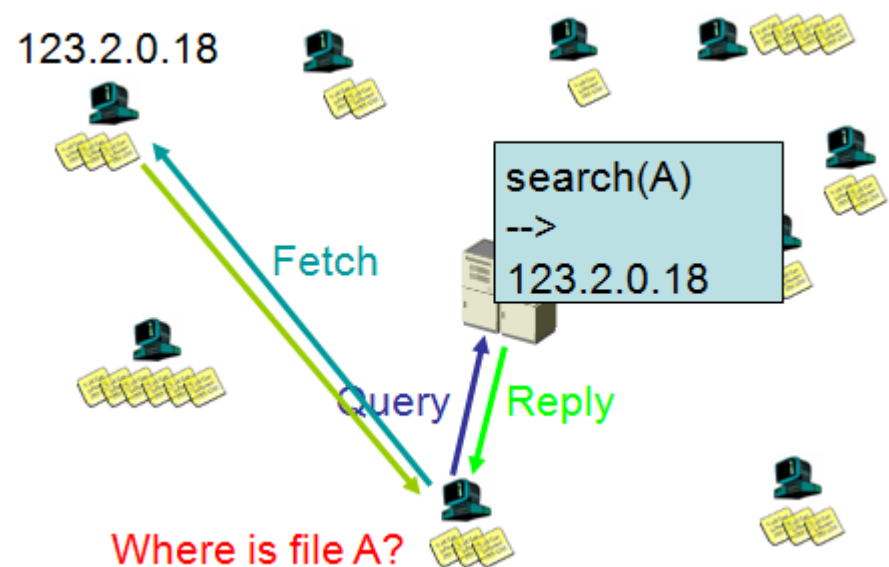
- 1999: Sean Fanning lanseaza Napster
- A atins cota de 1.5 milioane de utilizatori simultan
- Baze de date centralizata - operatii:
 - **Join**: clientul contacteaza serverul central (via TCP)
 - **Publish**: raportarea unei liste de fisiere serverului central
 - **Search**: interogarea serverului => se intoarce cineva care stocheaza fisierul cerut
 - **Fetch**: ia fisierul direct de la *peer* (cel cu cea mai buna rata de transfer)
- Iulie 2001: Napster a fost inchis

Arhitecturi nestructurate

Napster: Publish



Napster: Search



Discutii:

- Serverul face toate procesarile
- Avem "single point of failure"
- Probleme de scalabilitate, unele sisteme nu permit adaugarea altor servere (lista serverelor disponibile este statica)

Arhitecturi nestructurate

Descentralizate pure

- Se construiește o rețea acoperitoare (*overlay*) cu propriile mecanisme de rutare prin IP
- Nu există o coordonare centrală
- Utilizatorii se conectează via o aplicație care are rol dublu – *servent*
- Comunicarea între *serventi* se bazează pe un protocol la nivel de aplicație, cu 4 tipuri de mesaje:
 - Ping – cere ca un nod să se anunțe
 - Pong – replică la mesajul *ping* (IP, port, numărul & mărimea fișierelor)
 - Query – cerere de căutare (șir de căutare + viteză minimă de transfer)
 - Query hints – răspuns (IP, port, viteză, dim. fis., index fis.)

Arhitecturi nestructurate

Descentralizate pure

- Cautarea se realizeaza prin inundare (flooding)
 - Daca nu ai fisierul dorit, intreaba pe n vecini
 - Daca nici ei nu au fisierul, vor intreba pe vecinii lor in maxim m hop-uri
 - Pe calea de intoarcere se vor intoarce raspunsurile (nu continutul fisierelor)
- Fiecare mesaj are un TTL atasat
- Exemplu: **Gnutella**

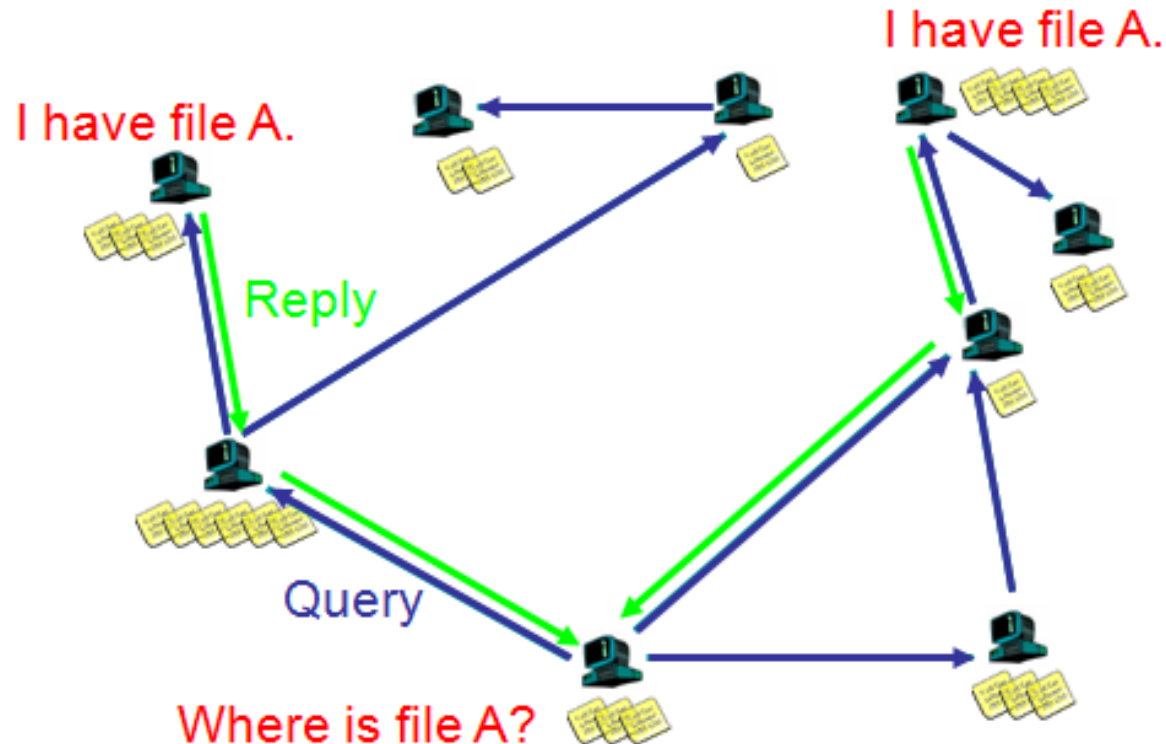
Arhitecturi nestructurate

Gnutella

- 2000: L. Frankel si T. Pepper(Nullsoft) lanseaza Gnutella
- Apar clienti: Bearshare, Morpheus, LimeWire
- *Query Flooding*:
 - **Join**: la intrare in sistem, clientul contacteaza cateva noduri care devin “vecinii” sai
 - **Publish**: nu este necesar
 - **Search**: se intreaba vecinii, care isi intreaba vecinii lor, ...
 - Exista un TTL ce limiteaza propagarea
 - **Fetch**: preia fisierul direct de la *peer*

Arhitecturi nestructurate

Gnutella



Aspecte:

- Timpul de cautare este... $O(?)$
- Nodurile pleaca adesea => retea instabila

Arhitecturi nestructurate

Partial centralizate

- Folosesc conceptul de **supernod**: are activitati de servire a unei sub-retele P2P (indexare, *caching*)
- Nodurile sunt alese automat ca fiind supernoduri daca au suficienta latime de banda si putere computationala
- Toate cererile sunt trimise initial la supernoduri
- Avantaje: timpul descoperirii resurselor e mai redus + eterogenitatea este exploatata
- Exemplu: **KaZaA**

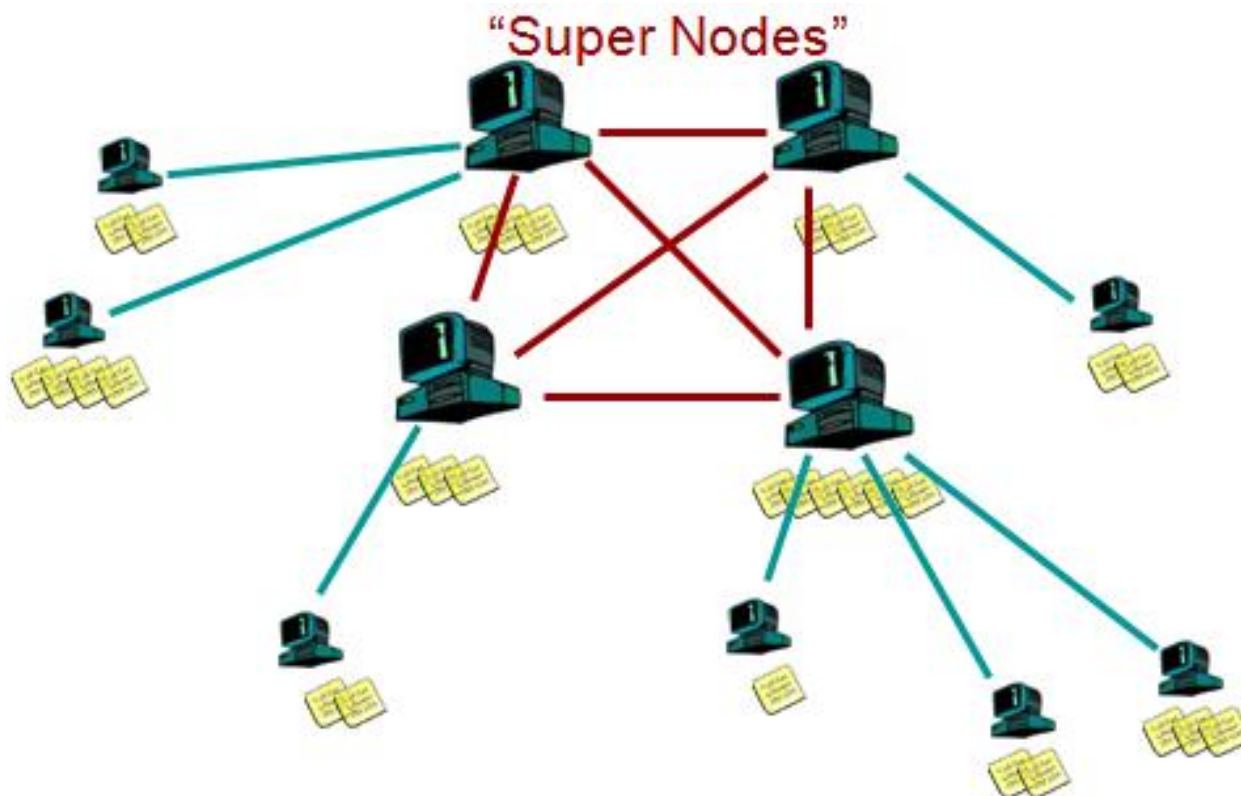
Arhitecturi nestructurate

KaZaA

- 2001: Se lanseaza KaZaA
- Apar clienti: Morpheus, giFT
- Se utilizeaza un mecanism de tip “*smart*” *query flooding*:
 - **Join**: la intrare in sistem, clientul contacteaza un “*supernode*” (poate deveni si el *supernod* la un moment dat)
 - **Publish**: trimite lista de fisiere *supernodului*
 - **Search**: trimite interogarea *supernodului*, si *supernodurile* se interogheaza intre ele
 - **Fetch**: ia fisierul direct de la *peer(s)*; poate prelua fisierul simultan de la mai multe *peer-uri*

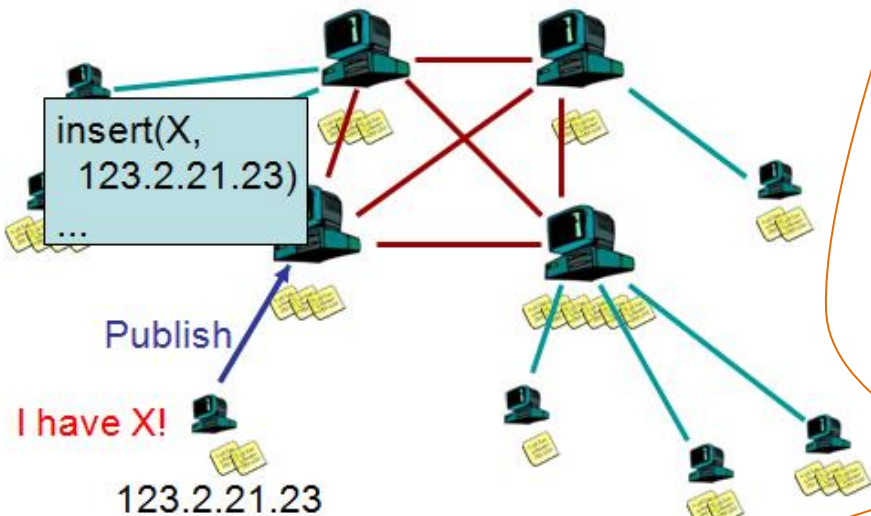
Arhitecturi nestructurate

KaZaA: Designul rețelei

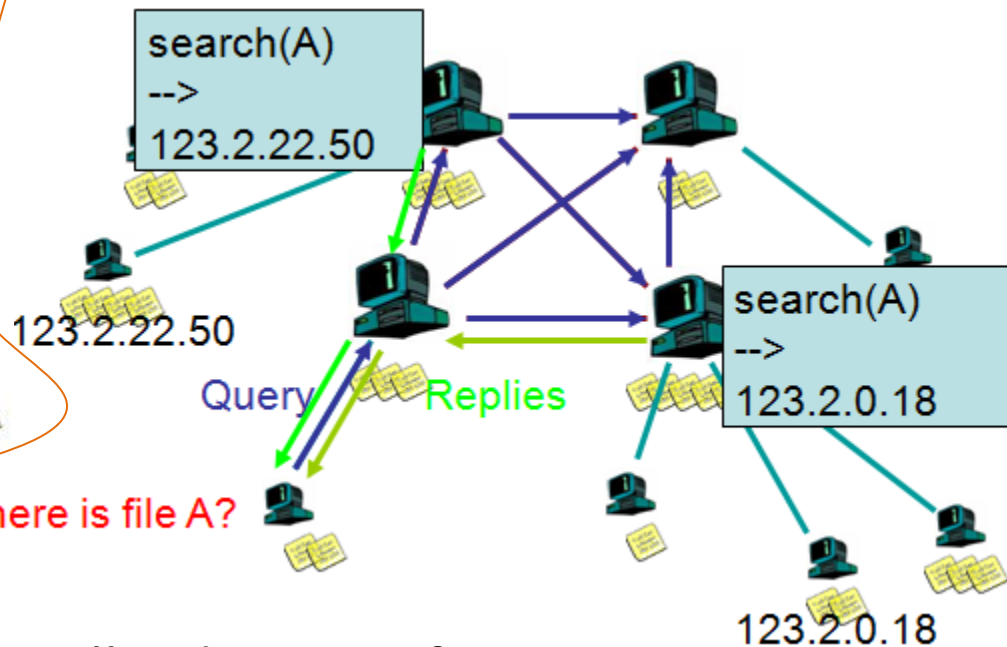


Arhitecturi nestructurate

KaZaA: Inserarea de Fisiere



KaZaA: Cautarea de Fisiere



Discutii:

- Comportament similar cu Gnutella, dar mai eficient
- Nu este nici o garantie asupra timpului de cautare sau a domeniului de cautare

Arhitecturi nestructurate

Partial centralizate

- Software-ul KaZaA este proprietar
- Datele de control P2P sunt criptate
- Mesajele folosesc HTTP
- Un nod e fie un supernod, fie asignat unui supernod
- Un supernod are 100-150 noduri-copil
- O retea poate avea ~30000 supernoduri
- Fiecare supernod are conexiuni TCP cu 30-50 supernoduri
- Pentru fiecare fisier se mentin meta-date (nume, dimensiune, *content hash*, descriptor de fisier)
- *Content hash*-ul este folosit pentru cautarea altei copii a unui fisier partial transferat
- Varianta fara *spyware* si *pop-up*-uri: **KaZaA-lite**

Arhitecturi nestructurate

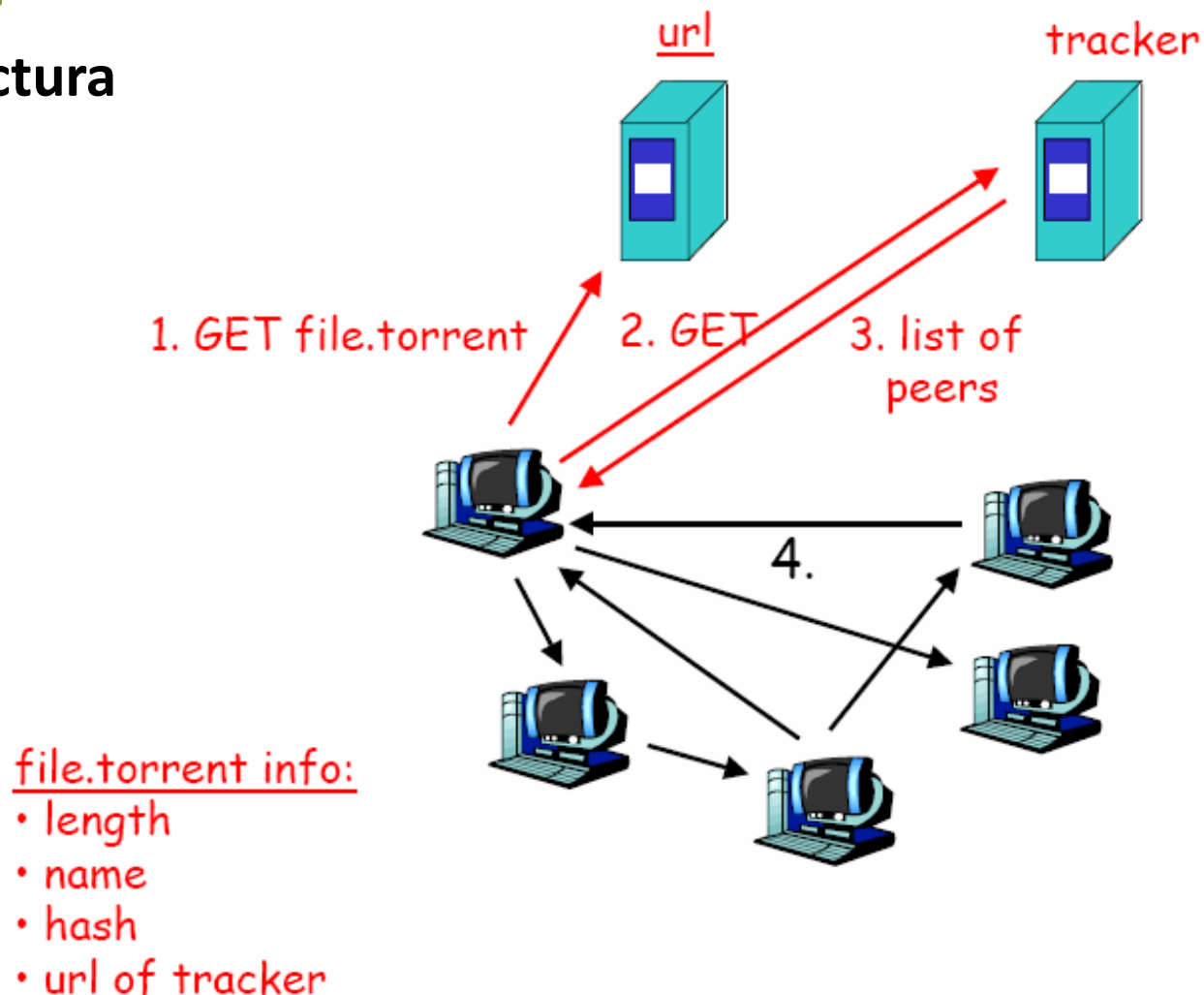
Partial centralizate

- Daca un fisier este gasit pe mai multe noduri, transferul poate fi realizat in paralel
 - Copiile identice se identifica via *content hash*
- Diferite portiuni din fisier sunt transferate de pe noduri diferite
- Pentru transferurile intrerupte, se face o recuperare automata (*automatic recovery*)
- Exemplu: **BitTorrent**
 - In 2002, B. Cohen a lansat BitTorrent
 - Si-a propus concentrarea pe problema legata de obtinerea eficienta a resurselor (*efficient fetching*) si nu pe cautare (*searching*)
 - Sustinatori inca de la aparitie
 - Blizzard Entertainment folosea BitTorrent pentru distributia versiunilor beta a noilor jocuri

Arhitecturi nestructurate

Partial centralizate

BitTorrent - arhitectura



Arhitecturi nestructurate

Partial centralizate

BitTorrent

- Se bazeaza pe mecanismul de *swarming*:
 - **Join**: contacteaza un server centralizat (*tracker*), si obtine o lista de *peer*-uri
 - **Publish**: ruleaza un server *tracker*
 - **Search**: e.g. foloseste Google pentru a gasi un *tracker* pentru fisierul dorit
 - **Fetch**: *Downlodeaza* bucati de fisiere de la *peer*-uri; incarca bucatile de fisier pe care le ai
- Obs. Diferenta fata de Napster
 - Downloadarea de bucati (*chunk*) de fisiere
 - Utilizarea strategiei “tit-for-tat”: A *downlodeaza* de la alte noduri, atunci A trebuie sa permita si *downloadarea* de la el (*free-rider problem*)_

Arhitecturi nestructurate

Probleme

- **Noduri ale caror adrese IP sunt disponibile via NAT**
 - Nu pot fi servere TCP pentru rețeaua P2P
 - Soluție parțială: *reverse call*
 - A vrea să transfere de la B, iar B folosește NAT
 - A și B stabilesc conexiuni TCP cu serverul C (IP rutabil)
 - A poate cere lui B, via C, să realizeze o conexiune TCP de la B la A
 - A poate trimite o cerere lui B, iar B îi oferă răspunsul
 - Dacă A și B utilizează NAT?
- **Flash crowd: o creștere neașteptată de cereri pentru o anumită resursă**
 - Pentru conținutul dorit nu există suficiente copii încărcate
 - Cât timp ia unui utilizator să localizeze fișierul?
 - Câte mesaje va primi un nod datorită căutărilor realizate de alte noduri?
 - Se poate folosi un protocol de căutare generic, bazat pe TTL

Arhitecturi structurate

- Reprezinta solutia **academica** pentru P2P
- Scop:
 - Cautare cu succes
 - Timp de cautare in limite cunoscute
 - Scalabilitate demonstrata
- Abordare: **DHT (Distributed Hash Table)**
 - Se stocheaza perechi (key, value)
 - Key – nume de fisiere
 - Value – continut de fisier sau pointer la o locatie
 - Fiecare *peer* stocheaza o multime de (key, value)
 - *Operatii: gaseste nodul responsabil cu un Key*
 - Mapare *key – node*
 - Rutarea eficienta a cererilor de *insert/lookup/delete* asociate cu acest nod
 - Se permite o mare fluctuatie a nodurilor

Arhitecturi structurate

- Aspect de interes: **localizarea conținutului**
- Idee: Responsabilitatea este distribuita mai multor noduri ale rețelei de acoperire, într-un mod adaptiv
- Fiecarei resurse i se asociază o cheie unică via o funcție *hash*: $h(\text{"Curs Retele"}) \rightarrow 7929$; Intervalul de valori ale funcției *hash* se distribuie în rețeaua P2P
- Fiecare nod trebuie să "cunoască" locația macar a unei singure copii a fiecărei resurse pentru care funcția sa *hash* ia valori în intervalul lui
- Nodurile pot menține în cache-ul propriu o copie a fiecărei resurse pe care trebuie să o "cunoască"

Arhitecturi structurate

- Aspect de interes: **dirijarea**
- Pentru fiecare resursa, un nod ce “cunoaste” resursa trebuie sa fie accesat pe calea cea mai “scurta”
- Abordarile de sisteme P2P structurate difera prin strategia de dirijare
- Nodurile din sistem formeaza o structura de date distribuita care poate fi: inel, arbore, hypercub, skip list, etc.
- Se ofera un API pentru tabelele distribuite de hash-uri (DHT – Distributed Hash Table)
 - Dand o cheie k , API-ul va returna adresa IP a nodului responsabil pentru valoarea cheii k

Arhitecturi structurale

- **Implementari**

- Chord [MIT]
- Pastry [Microsoft Research UK, Rice University]
- Tapestry [UC Berkeley]
- Content Addressable Network (CAN) [UC Berkeley]
- SkipNet [Microsoft Research US, Univ. of Washington]
- Kademlia [New York University]
- Viceroy [Israel, UC Berkeley]
- P-Grid [EPFL Switzerland]

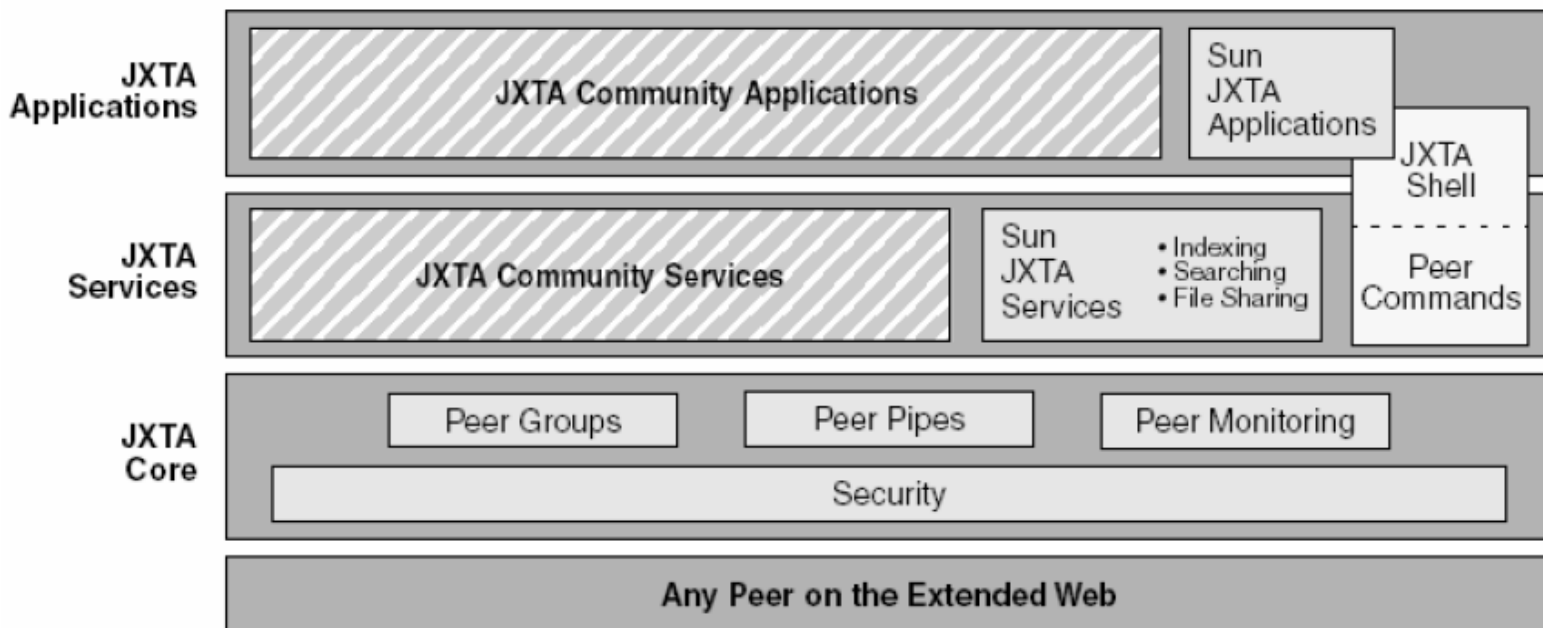
Arhitecturi structurate

- **Slab structurate**

- Nodurile pot estima ce noduri stocheaza resursele cautate
 - Se evita broadcasturile oarbe
 - Se foloseste o propagare in lant (*chain mode propagation*): fiecare nod ia decizii locale privitoare la care va fi nodul urmator interogat
- Cautarea unui fisier presupune utilizarea unei chei si a unui mecanism de *timeout (hops-to-live)*
- Exemplu: **Freenet**

Aplicatii | Exemplu

- JXTA – www.jxta.org
 - Mediu de dezvoltare a sistemelor & aplicatiilor P2P
 - Bazat pe java, disponibil in regim open source



- p2psim – simulator pentru protocoalele p2p
 - <http://pdos.csail.mit.edu/p2psim/>

Consumer Internet Traffic 2005–2011							
	2005	2006	2007	2008	2009	2010	2011
By Sub-Segment (terabytes per month)							
Web, e-mail, file transfer	362,084	505,996	692,812	948,425	1,233,172	1,603,615	2,756,415
P2P	1,060,226	1,329,770	1,772,403	2,379,025	3,111,891	4,040,403	5,269,360
Gaming	66,844	91,943	133,367	188,680	250,574	318,212	386,832
Video Communications	11,629	15,575	24,932	36,638	47,173	66,101	92,453
VoIP	10,965	23,035	39,339	57,653	75,575	92,815	110,456
Internet Video to PC	53,074	174,427	484,027	838,154	1,232,461	1,726,114	2,331,908
Internet Video to TV	0	12,727	110,692	353,095	620,197	936,580	1,342,482
By Geography (TB per month)							
North America	534,236	618,765	917,365	1,287,026	1,698,700	2,242,841	2,861,772
Western Europe	334,600	505,329	814,015	1,281,041	1,856,310	2,515,070	3,458,721
Asia Pacific	565,782	819,072	1,201,277	1,742,834	2,315,755	3,049,294	4,663,774
Japan	60,080	98,747	147,733	223,120	319,788	436,057	556,631
Latin America	19,917	33,755	57,083	90,765	130,466	189,992	268,559
Central Eastern Europe	40,773	59,097	86,196	122,272	165,387	222,895	294,901
Middle East and Africa	9,435	18,708	33,904	54,613	84,637	127,689	185,549

Statistici

Legenda:

Web, E-mail, and File Transfer – includes Web, e-mail, instant messaging, newsgroups, and file transfer (excluding P2P and commercial file transfer such as iTunes)

P2P – includes peer-to-peer traffic from all recognized P2P systems such as BitTorrent, eDonkey, etc.

Gaming – includes casual online gaming, networked console gaming, and multiplayer virtual world gaming

Video Communications – includes PC-based video calling, Webcam viewing, and Web-based video monitoring

VoIP – includes traffic from retail VoIP services and PC-based VoIP, but excludes wholesale VoIP transport

Internet Video to PC – free or pay TV or VoD viewed on a PC, excludes P2P video file downloads

Internet Video to TV – free or pay TV or VoD delivered via Internet but viewed on a TV screen using a STB or media gateway

Global Consumer Peer-to-Peer Traffic 2005–2011

Consumer Peer-to-Peer Traffic 2005–2011							
	2005	2006	2007	2008	2009	2010	2011
By Geography (TB per month)							
North America	381,746	378,538	462,356	560,817	673,083	852,483	1,080,979
Western Europe	223,519	304,988	411,057	540,032	757,818	991,817	1,330,885
Asia Pacific	391,235	550,664	762,276	1,074,759	1,401,028	1,811,094	2,327,648
Japan	28,621	42,883	58,463	87,446	117,967	154,868	206,803
Latin America	8,732	14,358	23,247	37,284	53,587	80,043	117,731
Central Eastern Europe	22,075	31,009	43,117	59,928	79,589	106,543	141,282
Middle East and Africa	4,297	7,329	11,886	18,759	28,819	43,553	64,033
Total (TB per month)							
Peer-to-Peer Traffic	1,060,226	1,329,770	1,772,403	2,379,025	3,111,891	4,040,403	5,269,360

Rezumat

- **Paradigma peer-to-peer(P2P)**
 - Preliminarii
 - Definitii
 - Caracterizare
 - Tipuri de aplicatii
 - Infrastructuri
 - Aplicatii

Bibliografie

- P2P Networking and Applications, John F. Buford, Heather Yu, Eng Keong Lua ,2009, Elsevier
- <http://mtc.sri.com/Conficker/P2P/>
- http://www.cisco.com/en/US/docs/cable/serv_exch/serv_control/broadband_app/protocol_ref_guide/01_p2p.pdf
- <http://pdos.csail.mit.edu/p2psim/>
- Statistici: http://www.sandvine.com/news/pr_detail.asp?ID=312
- Statistici: http://www.hbtf.org/files/cisco_IPforecast.pdf



Intrebari?

Intrebari?