University of Würzburg
Distributed Systems
Prof. Dr. P. Tran-Gia

Dr. K. Tutschku

Management of Peer-to-Peer Networks

2. Würzburger Workshop „IP Netzmanagement, IP Netzplanung und Optimierung“

Würzburg, 24. Juli 2001

Overview

- What is Peer-to-Peer Networking?
- Why P2P Networking?
- Selected P2P Architectures
- Management of P2P Networks
  - Security Management
  - Resource Management / Performance Management
- Conclusion

Introduction

gnutella

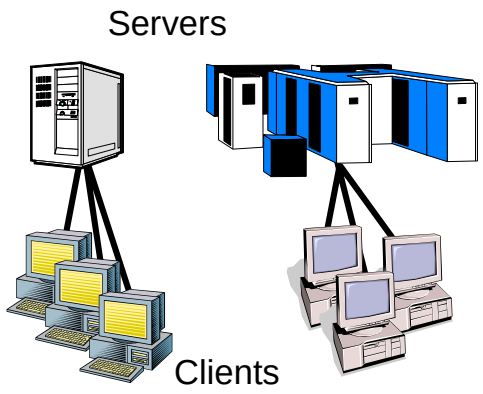
groove™

Project JXTA

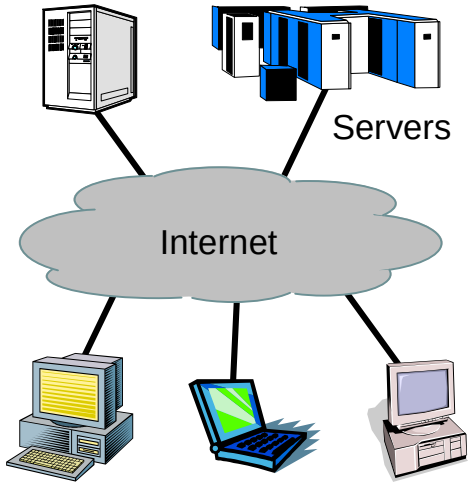
- Is P2P networking a hype?
  - O'Reilly P2P conference, March 2001 / panel at InfoCom 2001
  - Napster: 40 million user deployments in two years
  - strong support: Intel, Deutsche Bank

➔ Yes, but why? What is the impact on networks? How should it be handled ?

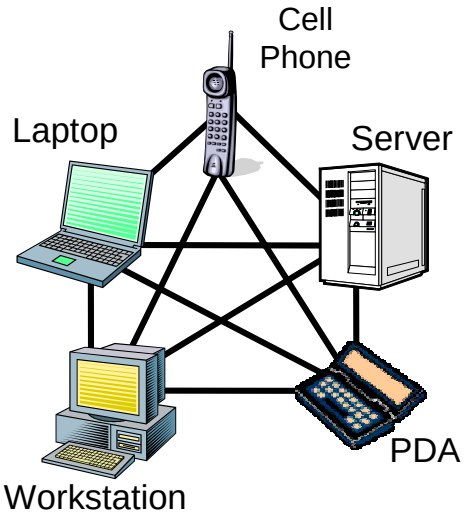
Evolution of Network Computing



Client/Server stacks



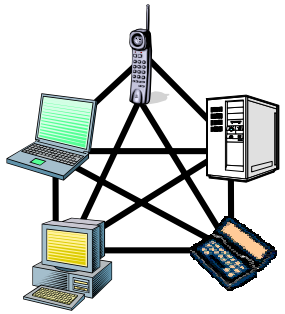
Web-based



Peer-to-Peer

- Up to 1994:
  - permanent IP addresses, static DNS mapping
  - always connected
  - only client/server
  - limited, specialized, centralized applications
- 1994 - present:
  - WWW
  - dynamic IP addresses / NAT / roaming users
  - heterogeneous
  - “linked” servers
  - asymectric server-based services
- Now:
  - collaboration and personalized app
  - powerful edge devices
  - instant network-ing

What is Peer-to-Peer (P2P) Networking



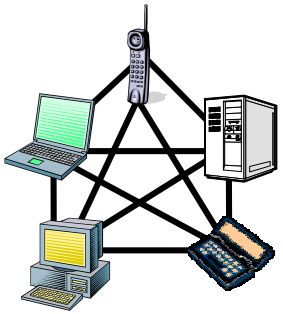
- *Simple definition:*
  - everything except the client/server model
- *Traditional definition:*
  - type of network of workstation with equivalent capabilities/responsibilities; different from client/server architectures, in which some computers are dedicated to serving the others

➔ *is this correct? e.g.: isn't the Internet already P2P?*

- **Litmus tests (Clay Shirkey's)**
  1) does it treat variable connectivity as the norm?
  2) does it give the nodes at the edges of the network significant autonomy?

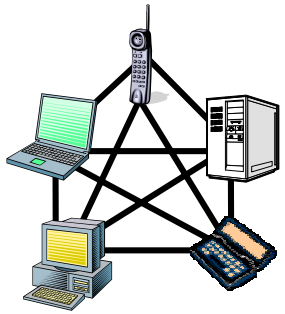
  ➔ if answer to both is yes then *application is P2P* else it's *not*

Peer-to-Peer Architecture Today



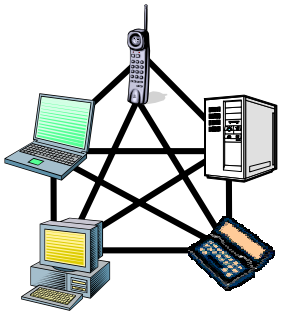
- *Characteristics:*
  - available at the edges of the Internet
  - have significant or total autonomy from central servers
  - nodes can act as both client and server; “SERVer + cliENT = SERVENT” concept
  - overall system is easy to use and well-integrated; may include tools to create content
  - take advantage of resources (storage, cycles, content, human presence)
  - able to operate in an unstable environment with unpredictable addresses; loosely connected
  - operates outside the DNS system
  - self-organization facility for group membership and resource/performance control

Why Peer-to-Peer Networking?



- P2P enables new services at the edge of the network
- P2P group collaboration superior for business processes:
  - grow organically, non-uniform, and high dynamic
  - largely manual, ad-hoc, iterative, and document-intensive
  - distributed, not centralized; no single person/organi-zation/application understands/controls the entire process from end-to-end
- P2P cost effectiveness:
  - reduces centralized management resources
  - optimizes computing, storage and communication resources
  - rapid deployment
- P2P applications/protocols tailored for user's needs

Areas of Application



- Distributed Collaboration/Communication:
  - P2P groupware, P2P content generation
  - P2P instant messaging system
  - online games
- Distributed Storage:
  - P2P file sharing, online backups
- Distributed Computing
  - P2P CPU cycle sharing
  - distributed simulation
- Distributed Search Engines, Intelligent agents

Selected P2P Applications - SETI@home

- Purpose:
  - massively parallel analysis of radio signals received from the Arecibo telescope
- Uses idle CPU cycles on ordinary PCs
- Central SETI@home server distributes data
- Analyzer: SETI@home screen saver

- Classification (according to K. Kant):
  - (scattered, organized, isolated, non_RT)

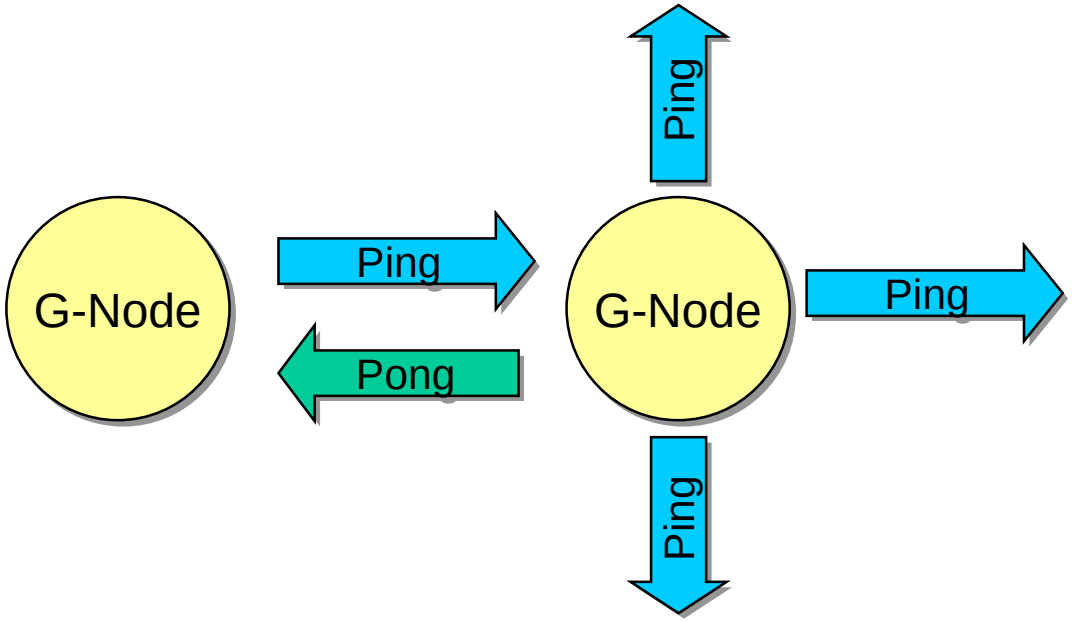
- Similar architectures:
  - Napster

Selected P2P Applications - Gnutella (1)

- Purpose:
  - distributed and anonymous file sharing
- Exploits unused storage on edge nodes
- Servents operate completely without central control
- Characteristics:
  - numeric IP addresses; message broadcasting for node discovering and search requests
  - connecting: join the “several known hosts”
  - data transfer: *store and forward* using HTTP
- Classification:
  - (scattered, scattered, isolated, non_RT)

Selected P2P Applications - Gnutella (2)

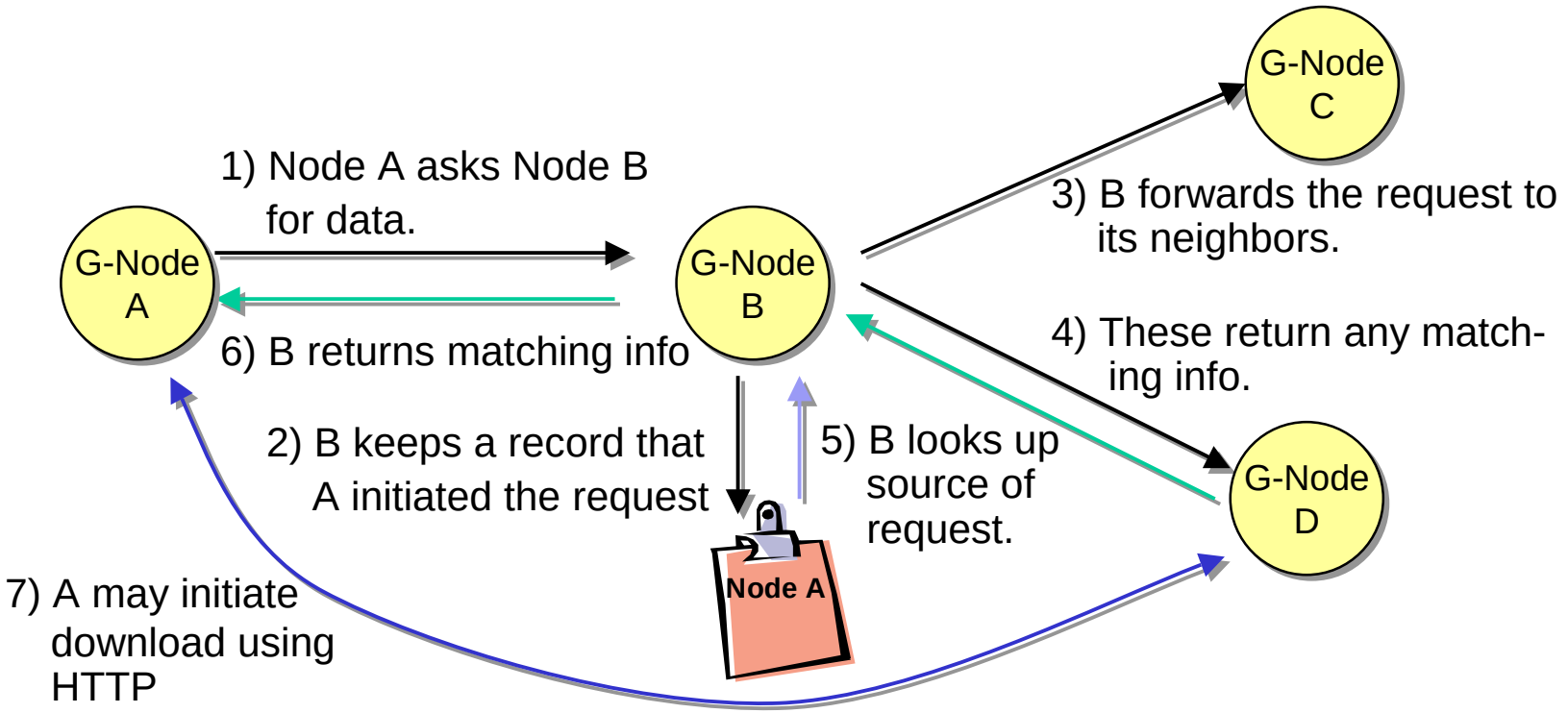
- Call-and-Response protocol mechanism: ***node discovery***



- PING/PONG messages with TTL to limit broadcasting range.
- short time memory of messages already seen; prevents re-broadcasting. GUIDs to distinguish messages

Selected P2P Applications - Gnutella (3)

- Call-and-Response protocol mechanism: ***search query*** / ***download***

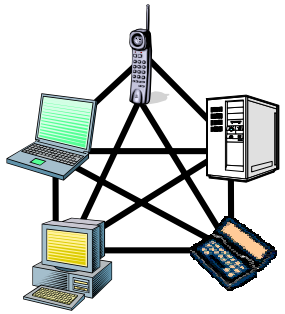


- search: Query/Query-Response
- download: GET/PUSH.

Selected P2P Applications - Gnutella (4)

- Limitations/Problems:
  - unstable/loose connectivity of the Servents
    →performance management difficult
  - scalability: e.g. TTL=10, every node broadcasts to six others
    → 6^{10} msg; problem in huge networks
  - denial-of-service attacks
  - low TTL, low search horizon
  - HTTP-protocol: lots of information available for adversary
  - authentication is application specific

Challenges / Shortcomings in P2P architectures



- Security and privacy issues
  - authentication, authorization, anonymity of users
  - self-organization
- Robustness
  - availability of resources
  - hard to predict the consequences of failures
  - administrative actions
- Performance
  - bandwidth consumption
  - scalability, self-organization
  - end-to-end quality-of-service
  - overload / network planning

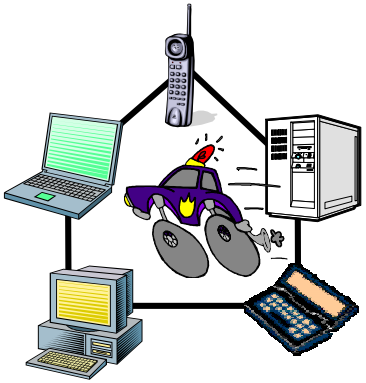
➔ smart security and resource/performance management required

P2P Security Management

- Goals:
  - no user can impersonate another user's identity
  - strong security always in force:
    - can not accidentally/intentionally be turned off;
    - consistent information throughout the net
  - no eavesdropping
  - group collaboration example:
    - all information is confidential; readabel/writeable only group members; messages can be securely recovered

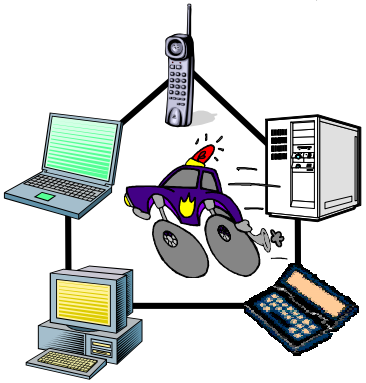
- Approaches:
  - Groove: shared spaces - incremental changes transmitted to all devices
  - Napster/Morpheus: central user registration and logon
  - Mojo Nation: RSA encryption

P2P Resource/Performance Management



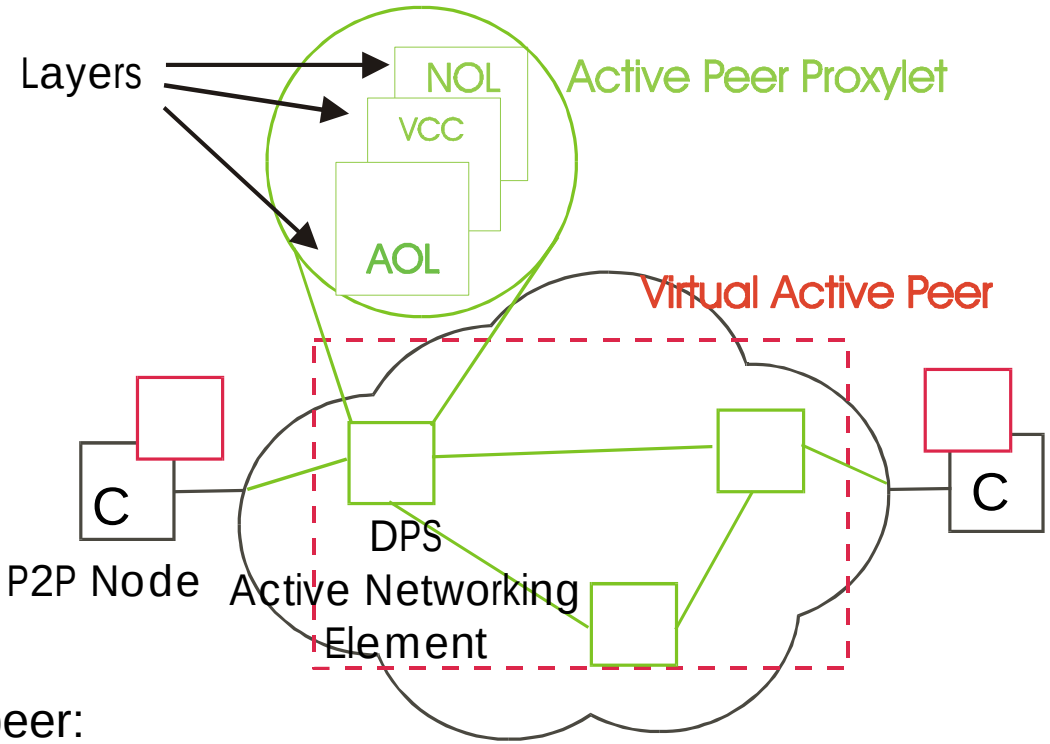
- Goals:
  - maximizes a peer's utility to the overall system while minimizing its potential threat
  - increase stability
  - introduce administrative rules
- Problems to tackle:
  - simple overuse (e.g. freeloaders)
  - intentional attacks
  - resource allocation
  - reduction of synchronization traffic
  - aggregation and self-organization
  - latency/bandwidth/packet loss

P2P Resource/Performance Management



- Approaches:
  - enhance P2P protocols:
    - smart topology construction (FLAPPS - Michel et al., 2001)
    - sophisticated group multicasting (Lee et al., 2001)
  - accountability:
    - used in Free Haven project / Mojo Nation
    - restricting access: “micro payments”
    - selecting favored users: reputation system
  - superpeers:
    - Morpheus/KaZaA, Clip2’s Gnutella Reflector
    - *Virtual Active Peers* (deMeer/Tutschku, 2001)
      - based on Application-Level Active Networks

“Virtual Active Peer” Architecture



- Superpeer:
  - layering provides control capability
  - permits topology management
- Application-level Routing
  - optimizes for different metrics (e.g. privacy, policies, latency)
  - provides smart multicast, caching and replication capabilities

Conclusion & Outlook

- P2P networking is a promising paradigm for services operating at the edges of the network
- decentralized P2P applications offer big cost/time savings
- P2P networks currently scale in small to mid-size networks
- many open issues in P2P security, resource and performance management:
  - e.g.: self-organization, metrics, reliability, scalability, multi-media
- Outlook:
  - HailStorm: Microsoft's set of peer-to-peer technologies;
  - part of *.NET;* uses XML, SOAP, and proprietary authentication