Introduction to Peer-to-Peer Systems

吳俊興
國立高雄大學 資訊工程學系
Outline

• Introduction
  – Why P2P

• Basic P2P Technologies – Search/Lookup
  – Centralized
  – Unstructured
  – Structured

• Comments on P2P
Peer-to-peer Model

“Peer-to-Peer (P2P) is a way of structuring distributed applications such that the individual nodes have symmetric roles. Rather than being divided into clients and servers each with quite distinct roles, in P2P applications a node may act as both a client and a server.”

Excerpt from the Charter of Peer-to-Peer Research Group, IETF/IRTF, June 24, 2003
http://www.irtf.org/charters/p2prg.html

Peers play similar roles
No distinction of responsibilities
Client-server Model

Clients and servers each with distinct roles

The server and the network become the bottlenecks and points of failure

- DDoS
- Flash Crowd
Content Distribution Networks

Hosting + Hierarchical Proxies + DNS Request Routing
(Akamai, CacheFlow, etc.)

Overlay Network

Content Router or Peer Node

Name: lb1.www.ms.akadns.net
Addresses: 207.46.20.60, 207.46.18.30, 207.46.19.30, 207.46.19.60, 207.46.20.30

Name: www.yahoo.akadns.net
Addresses: 66.94.230.33, 66.94.230.34, 66.94.230.35, 66.94.230.39, 66.94.230.40, …
Aliases: www.yahoo.com

Name: e96.g.akamaiedge.net
Address: 202.177.217.122

Name: a1289.g.akamai.net
Addresses: 203.133.9.9, 203.133.9.11
Content Distribution Networks (Cont.)

ICPs need contents, servers, equipments, IT experts, etc.

Now, ICPs need only contents and trusted CNPs.

“Serverless ICP” (Fabless IC Design House)

“Content Foundry” (Semiconductor Foundry)

I.C. = Internet Contents
P2P Example - Napster

Any two P2P nodes can exchange contents directly
+ Running P2P software is much easier than maintaining servers
+ Reduce the load of server, even without servers
Paradigm Shift of Computing System Models

1980~
Terminal-Mainframe (Super-computing)

1990~
Client-Server (Micro-computing /Personal Computer)

2000~
Peer-to-Peer (Macro-computing)

RS-232
VT100/DOS

Dialup/10M Ethernet
Windows 31/95

ADSL/100M+ Ethernet
Linux/Windows XP
Generations

• Early services
  – DNS, Netnews/Usenet
  – Xerox Grapevine name/mail service
  – Lamport’s part-time parliament algorithm
  – Bayou replicated storage system
  – classless inter-domain IP routing algorithm

• 1st generation
  – Napster

• 2nd generation
  – Freenet, Gnutella, Kazaa, BitTorrent

• 3rd generation – P2P middleware
  – Pastry, Tapestry, CAN, Chord, Kademlia
Example of P2P File Swapping: KaZaA

<table>
<thead>
<tr>
<th>As of July 2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users Online</td>
</tr>
<tr>
<td>Total Number of MP3 Files</td>
</tr>
<tr>
<td>Total Size of Movies Files</td>
</tr>
<tr>
<td>Total Number of Files Shared</td>
</tr>
<tr>
<td>Total Size of Shared Files</td>
</tr>
</tbody>
</table>

- 77% of surveyed companies had at least one installation of file-swapping software (AssetMetrix, July 16, 2003)
- Today more KaZaA traffic than Web traffic!
Skype – P2P VoIP

Background
- Created by Niklas Zennstrom and Janus Friis, founders of KaZaA
- Beta launched August 29, 2003
- As of 16 Nov 2004: 34,642,017 downloads and 2,551,163,700 Minutes served (1,771,641 days; 4853 years)
- Uses 3-16 KB/s while calling and 0-0.5 KB/s while idle

Features
- High-quality Internet conference calls
- Firewall and NAT traversal
- Global decentralized user directory
- Intelligent routing for encrypted calls through effective path
- Encrypting all calls and instant messages end-to-end
P2P Applications

- P2P File Swapping (Sharing)
  - Napster, FreeNet, Gnutella, KaZaA, eDonkey/eMule, EZPeer, Kuro, BT
- P2P Communication
  - NetNews (NNTP), Instant Messaging (IM), Skype (VoIP)
- P2P Lookup Services and Their Applications
  (Distributed Hash Tables and Global Repositories)
  - IRIS, Chord/CFS, Tapestry/OceanStore, Pastry/PAST, CAN
- P2P Overlay Networking
  - (Inter-domain Routing – BGP), RON, PDF, Detour
- P2P Multimedia Streaming (Application-layer Multicast)
  - CoopNet, Zigzag, Narada, P2Cast, Joost, PPStream
- Proxies and Content Distribution Networks
  - Squid, Akamai, DigitIsland
- Overlay Testbed
  - PlanetLab, NetBed/EmuLab
- Other Areas
  - P2P Gaming, Grid Computing
Basic P2P Issues

• How to connect each other?
  – Topology

• How to find out resources?
  – Search / Lookup

• P2P Classification
  – Centralized
  – Unstructured: flooding (DFS, BFS)
  – Structured: tree, hypercube
Example of Centralized P2P Systems: Napster

- Announced in January 1999 by Shawn Fanning for sharing MP3 files and pulled plug in July 2001
- Centralized server for search, direct file transfer among peer nodes
- Proprietary client-server protocol and client-client protocol
- Relying on the user to choose a ‘best’ source
- Disruptive, proof of concepts
- IPR and firewall issues
Example of Decentralized P2P Systems: Gnutella

- Open source
- 3/14/2000: Released by NullSoft/AOL, almost immediately withdrawn, and became open source
- Message flooding: serverless, decentralized search by message broadcast, direct file transfer using HTTP
- Limited-scope query

Servent (=Server + Client)
Example of Unstructured P2P Systems: Freenet

- Ian Clarke, Scotland, 2000
- Distributed depth-first search, Exhaustive search
- File hash key, lexicographically closest match
- \textit{Store-and-forward} file transfer
- Anonymity
- Open source
Example of Hybrid P2P Systems: FastTrack / KaZaA

- Proprietary software developed by FastTrack in Amsterdam and licensed to many companies
- Summer 2001, Sharman networks, founded in Vanuatu, acquires FastTrack

- Hierarchical supernodes (Ultra-peers)
- Dedicated authentication server and supernode list server
- From user’s perspective, it’s like Google
- Encrypted files and control data transported using HTTP
- Parallel download
- Automatically switch to new server
Example of Structured P2P Systems: Chord

- Frans Kaashoek, et. al., MIT, 2001
- IRIS: Infrastructure for Resilient Internet Systems, 2003
- Distributed Hashing Table
- Scalable lookup service
- Hyper-cubic structure
Skype – P2P VoIP

Background
- Created by Niklas Zennstrom and Janus Friis, founders of KaZaA
- Beta launched August 29, 2003
- As of 16 Nov 2004: 34,642,017 downloads and 2,551,163,700 Minutes served (1,771,641 days; 4853 years)
- Uses 3-16 KB/s while calling and 0-0.5 KB/s while idle

Features
- High-quality Internet conference calls
- Firewall and NAT traversal
- Global decentralized user directory
- Intelligent routing for encrypted calls through effective path
- Encrypting all calls and instant messages end-to-end
BitTorrent: Swarming and Tit-for-Tat

• Swarming: splitting and pipelining
  – Search: Out-of-band. E.g., use Google to find the .torrent metafile for the file you want
    • file information: length, name, hashing information
    • URL of tracker
  – Join: contact centralized “tracker” server, get a random list of peers
  – Publish: Run a tracker server
  – Fetch: Download chunks of the file from your peers. Upload chunks you have to them.

• Tit-for-tat: “I’ll share with you if you share with me”
  – Be optimistic: occasionally let freeloaders download
Outline

• Road to P2P Systems
  – Why P2P
  – Definition of P2P

• Basic P2P Technologies – Search/Lookup
  – Centralized
  – Unstructured
  – Structured

• Comments on P2P
Characteristics of P2P Systems

• Heterogeneous
  – Ubiquitous devices: PDAs, laptops, desktops, servers, etc
  – Multi-roles: clients, servers, and relays

• Autonomous
  – Join or leave at any time: network churn
  – Share or not share: incentive
  – Select or be selected: peer selection

• Self-organizing
  – Centralized or ultra-peer infrastructure
  – Decentralized: structured or non-structured
Strengths of P2P Systems

• Solve distributed denial-of-service (DDoS) problems
• Support serverless content providers
• Achieve high-performance systems through cooperation among participants
  – content distribution and file swapping
  – application-layer multicasting
  – overlay networking
  – VOIP communication
• Resilient - ease management overheads of large-scale distributed systems
Challenges to P2P Systems

In an ideal P2P system
- Each node will contribute as he can and consume as he need
- It will cooperate to perform much more stably, securely and efficiently than any single one computer

In a real open P2P system
- There are many nodes that consume much more resources than they contribute ⇒ *Fairness*
- Some nodes or groups of nodes who may attack or cheat the other normal nodes of the system ⇒ *Trustworthy*

These may probably make the system malfunctioned, unfair or inefficient
Attacks on P2P Systems

- **Sybil attacks** – any node could manufacture any number of identities
- **Malicious peers or frauds**
  - Announce false capacities or query results
  - Accept payments or obtain contents but fail to complete the transaction
- **Spurious files or messages**
  - Malicious members could put up dummy/polluted files giving them popular names
- **Collusion**
  - Collect credits from colluded dummy peers
Summary

• Introduction
  – P2P definition: client + server / relay
  – Why P2P

• Basic P2P Technologies – Search/Lookup
  – Centralized
  – Unstructured: flooding
  – Structured

• Comments on P2P
  – Characteristics
  – Strengths
  – Challenges
  – Attacks