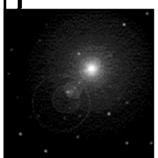


A Course on Foundations of
Peer-to-Peer Systems & Applications

**CS 6/75995**
**Foundation of Peer-to-Peer**

**Applications & Systems**

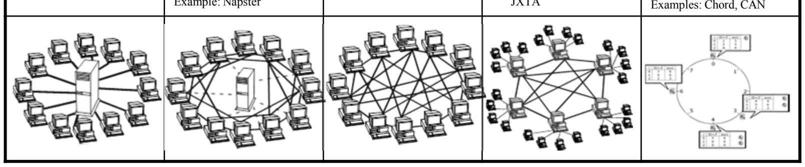
# Kent State University

Dept. of Computer Science
www.cs.kent.edu/~javed/class-P2P08/
LECT- 7

# Vibrant Systems:
# Gnutella Family

## Classification
[*Eberspacher & Schollmeier 2005]

| ***Client-Server*** | ***Peer-to-Peer*** | | | |
|---|---|---|---|---|
| 1. Server is the central entity and only provider of service and content. → Network managed by the Server<br>2. Server as the higher performance system.<br>3. Clients as the lower performance system<br><br>Example: WWW | 1. Resources are shared between the peers<br>2. Resources can be accessed directly from other peers<br>3. Peer is provider and requestor (Servent concept) | | | |
| | ***Unstructured P2P*** | | | ***Structured P2P*** |
| | *Centralized P2P* | *Pure P2P* | *Hybrid P2P* | *DHT-Based* |
| | 1. All features of Peer-to-Peer included<br>2. Central entity is necessary to provide the service<br>3. Central entity is some kind of index/group database<br>Example: Napster | 1. All features of Peer-to-Peer included<br>2. Any terminal entity can be removed without loss of functionality<br>3. → No central entities<br>Examples: Gnutella 0.4, Freenet | 1. All features of Peer-to-Peer included<br>2. Any terminal entity can be removed without loss of functionality<br>3. → dynamic central entities<br>Example: Gnutella 0.6, JXTA | 1. All features of Peer-to-Peer included<br>2. Any terminal entity can be removed without loss of functionality<br>3. → No central entities<br>4. Connections in the overlay are "fixed"<br>Examples: Chord, CAN |

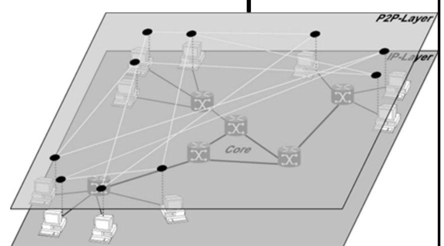


**1st Gen.** **2nd Gen.**

FOUNDATION OF PEER-TO-PEER SYSTEMS

# Overview

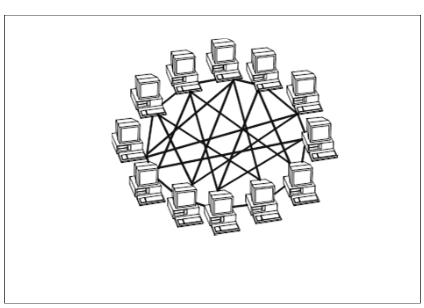
1. General Characteristics of Early Peer-to-Peer Systems
2. Centralized Peer-to-Peer Networks
    1. Basic Characteristics
    2. Signaling Characteristics
    3. Discussion
3. Pure Peer-to-Peer Networks
    1. Basic Characteristics
    2. Signaling Characteristics
    3. Discussion
4. Hybrid Peer-to-Peer Networks
    1. Basic Characteristics
    2. Signaling Characteristics
    3. Discussion






# Definition of Pure P2P

- Any terminal entity can be removed without loss of functionality
- No central entities employed in the overlay
- Peers establish connections between each other randomly
    - To route request and response messages
    - To insert request messages into the overlay



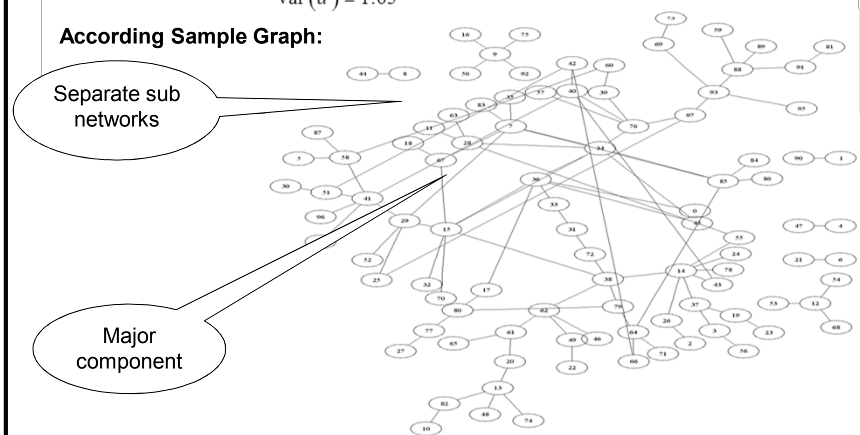
# Model of Pure P2P Networks

**Degree distribution:**

$$p(d) = \begin{cases} c \cdot d^{-1.4}, \; 0 < d \le 7 \\ 0, \; in\ any\ other\ case \end{cases}, \; with\ c = \left( \sum_{d} \frac{p(d)}{c} \right)^{-1}$$

$$average: \bar{d} = 2.2$$
$$\mathrm{var}(d) = 1.63$$

**According Sample Graph:**

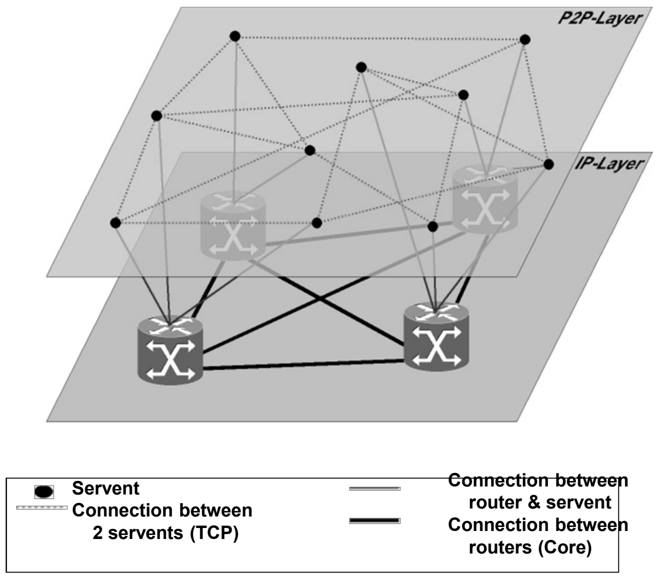





# Basic Characteristics of Pure P2P

- Bootstrapping:
  - Via bootstrap-server (host list from a web server)
  - Via peer-cache (from previous sessions)
  - Via well-known host
  - No registration
- Routing:
  - Completely decentralized
  - Reactive protocol: routes to content providers are only established on demand, no content announcements
  - Requests: flooding (limited by TTL and GUID)
  - Responses: routed (Backward routing with help of GUID)
- Signaling connections (stable, as long as neighbors do not change):
  - Based on TCP
  - Keep-alive
  - Content search
- Content transfer connections (temporary):
  - Based on HTTP
  - Out of band transmission

# Topology of Pure P2P




FOUNDATION OF PEER-TO-PEER SYSTEMS



# Example:
# Gnutella 0.4

# Gnutella Social History

- Program for sharing files over the Internet
- Focus: decentralized method of searching for files
- A "disruptive" application/technology?

- A decade of existence on March 14, 2010 In late 2007, it was the most popular file sharing network on the Internet with an estimated market share of more than 40%.

- Brief History:
  - **March 2000**: open source release by by Justin Frankel and Tom Pepper of Nullsoft, a division of AOL, and almost immediately withdrawn because of legal concern.
  - This did not stop Gnutella; after a few days, the protocol had been reverse engineered, and compatible free and open-source clones began to appear.
  - Still the third most popular filesharing system (after Bitorrent & FastTrack), 2008.
  - **Spring 2001**: further developments to improve scalability → Gnutella 0.6 (Hybrid P2P)
  - Since then:
    - available in a lot of implementations (Limewire, bearshare,…)
    - Developed further on (privacy, scalability, performance,…)





# Gnutella Social History (cont..)

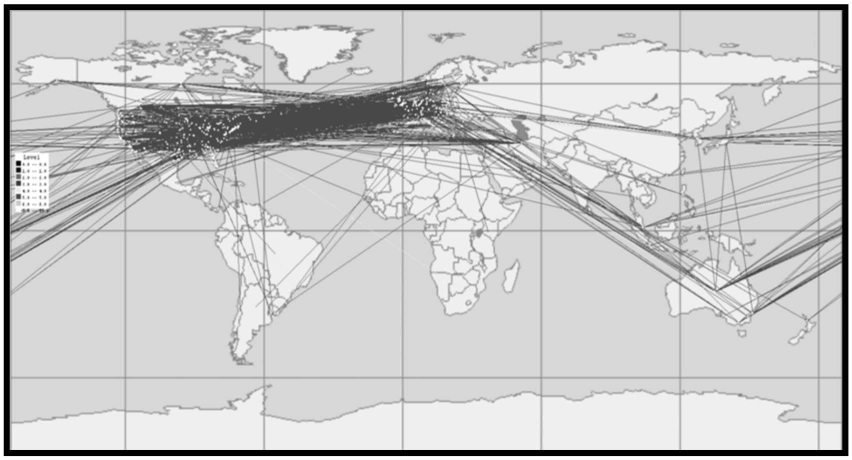
- On October 26, 2010, Gnutella servent LimeWire was ordered shut down . This event reduced the size of the network. The shut down did not affect FrostWire, a 2004 fork of LimeWire excluded blocking code and adware.

- Since LimeWire is free software, nothing prevents people from making additional forks of LimeWire as long as it does not use LimeWire trademarks.

- On November 9, 2010, LimeWire was resurrected by a secret team of developers and named LimeWire Pirate Edition.[10] It was based on LimeWire 5.6 BETA. This version had its server dependencies removed and all the PRO features enabled for free.

FOUNDATION OF PEER-TO-PEER SYSTEMS

# The Gnutella Network



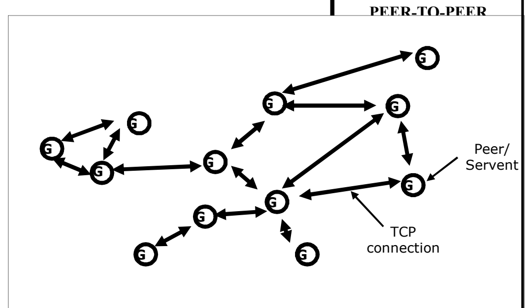
Measurements taken at the LKN in May 2002

FOUNDATION OF PEER-TO-PEER SYSTEMS



# Gnutella: How Does It Work

- Application-level, peer-to-peer protocol over point-to-point TCP Partcipants:
  - Gnutella peers/servents
  - Router Service
    - Flood incoming requests (regard TTL!)
      - Keep alive
      - content
    - Route responses for other peers (regard GUID of message)
      - Keep alive (PING/PONG)
      - Content (QUERY/QUERYHIT)
    - Data-requests
    - Download-requests
  - Lookup Service
    - Initialize Data requests
    - Initialize keep alive requests
  - "Server"-Service
    - Serve Data-requests (HTTP)
- Five steps:
  - Connect to at least one active peer (address received from bootstrap)
  - Explore your neighborhood (PING/PONG)
  - Submit Query with a list of keywords to your neighbors (they forward it)
  - Select "best" of correct answers (which we receive after a while)
  - Connect to providing host/peer




FOUNDATION OF PEER-TO-PEER

## Node Joining

- In order to join the system a new node/servent initially connects to one of several known hosts that are almost always available (e.g., gnutellahosts.com).

- Once attached to the network (e.g., having one or more open connections with nodes already in the network), nodes send messages to interact with each
- other.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-15
FP2P13F, javed@kent.edu
Javed I. Khan@2008

## PING & PONG

- *Used to Establish Group Membership.*

- A node joining the network initiates a broadcasted PING message to announce its presence.

- When a node receives a PING message it forwards it to its neighbors and initiates a back-propagated PONG message.

- The PONG message contains information about the node such as its IP address and the number and size of shared files.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-16
FP2P13F, javed@kent.edu
Javed I. Khan@2008

# QUERY & QUERY HIT

- *Used For Searching*.
- QUERY messages contain a user specified search string that each receiving node matches against locally stored file names. QUERY messages are broadcasted.
- QUERY RESPONSES are back-propagated replies to QUERY messages and include information necessary to download a file.

FOUNDATION OF PEER-TO-PEER SYSTEMS



# Gnutella Message Structure

## General Header Structure:

**MESSAGEHEADER**: 23Byte

| **GnodeID** | **Function** | **TTL** | **Hops** | **Payload Length** |
|---|---|---|---|---|
| 16 Bytes | 1 Byte | 1 Byte | 1 Byte | 4 Bytes |

Describes the message type (e.g. login, search,…)

Describes parameters of the message (e.g. IDs, keywords,…)

- **GnodeID**: unique 128bit Id of any Hosts
- **TTL**(Time-To-Live): number of servents, a message may pass before it is killed
- **Hops**: number of servents a message already passed

FOUNDATION OF PEER-TO-PEER SYSTEMS

# Gnutella Messages

**PING** (Function:0x00) *No Payload*

**PONG** (Function:0x01)

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

**QUERY** (Function:0x80)

| **Minimum Speed** | **Search Criteria** |
|---|---|
| 2 Bytes | n Bytes |

**QUERY HIT** (Function:0x81)

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

Result Set:

| **File Index** | **File Name** |
|---|---|
| 4 Bytes | n Bytes |

FOUNDATION OF PEER-TO-PEER SYSTEMS



# PONG MESSAGE

| **Bytes** | **Field name** | **Description** |
|---|---|---|
| 0-1 | Port Number | The port number on which the responding host can accept incoming connections. |
| 2-5 | IP Address | The IP address of the responding host. Note: This field is in big-endian format. |
| 6-9 | Number of shared files | The number of files that the servent with the given IP address and port is sharing on the network. |
| 10-13 | Number of kilobytes shared | The number of kilobytes of data that the servent with the given IP address and port is sharing on the network. |
| 14- | GGEP block | OPTIONAL extension (see GGEP). |

FOUNDATION OF PEER-TO-PEER SYSTEMS

# QUERY MESSAGE

| Bytes | Field name | Description |
|---|---|---|
| 0-1 | Minimum Speed (Flags) | The minimum speed (in kb/second) of servents that should respond to this message. A servent receiving a Query message with a Minimum Speed field of n kb/s SHOULD only respond with a Query Hit if it is able to communicate at a speed >= n kb/s. |
| 2- | Search Criteria | This field is terminated by a NUL (0x00). See section 2.2.7.3 for rules and information on how to Interpret the Search Criteria |
| Rest | Extensions Block | OPTIONAL. The rest of the query message is used for extensions to the original query format. The allowed extension types are GGEP, HUGE and XML (see Section 2.3 and Appendixes 1 and 2).<br>If two or more of these extension types exist together, they are separated by a 0x1C (file separator) byte. Since GGEP blocks can contain 0x1C bytes, the GGEP block, if present, MUST be located after any HUGE and XML blocks.<br>The type of each block can be determined by looking for the prefixes "urn:" for a HUGE block, "<" or "{" for XML and 0xC3 for GGEP.<br>The extension block SHOULD NOT be followed by a null (0x00) byte, but some servents wrongly do that. |





# QUERY Flag/ Speed Flag (Actually G.06)

| Bit | Flag | Description |
|---|---|---|
| 15 | MinSpeed/Flags Indicator | MUST be set to 1 to indicate that the flags below are used instead of encoding the Minimum Speed. |
| 14 | Firewalled Indicator | The host who sent the query is unable to accept incoming connections. This flag can be used by the remote servent to avoid returning Query Hits if it is itself firewalled, as the requesting servent will not be able to download any files. |
| 13 | XML Metadata | Set this bit to 1 if you want the sharing servent to send XML Metadata in the Query Hit. This flag has been assigned to spare bandwidth, returning metadata in queryHits only if the requester asks for it. If this bit is not set, the sharing host MUST NOT send XML metadata in return Query Hit messages. |
| 12 | Leaf Guided Dynamic Query | When the bit is set to 1, this means that the query is sent by a leaf which wants to control the dynamic query mechanism. This is part of the Leaf guidance of dynamic queries proposal. This information is only used by the ultrapeers shielding this leave if they implement leaf guidance of dynamic queries. If this bit is set in a Query from a Leaf it indicates that the Leaf will respond to Vendor Messages from its Ultrapeer about the status of the search results for the Query. |
| 11 | GGEP "H" Allowed | If this bit is set to 1, then the sender is able to parse the GGEP "H" extension which is a replacement for the legacy HUGE GEM extension. This is meant to start replacing the GEM mechanism with GGEP extensions, as GEM extensions are now deprecated. |
| 10 | OOB Query | This flag is used to recognize a Query which was sent using the Out Of Band Query extension. |
| 9 | ? | Reserved for a future use. Must be set to 0. |
| 0-8 | Maximum Query Hits | Set when a maximum number of Query Hits is expected, 0 if no maximum. This does not mean that no more Query Hits may be returned, but that the query should be propagated in a way that will cause the specified number of hits. |

# Gnutella Routing

- **Basic Routing Principle: "Enhanced" Flooding**

- Save Origin of received **PINGs** and **QUERIEs**
- Decrease **TTL** by 1
- If **TTL** equals 0, kill the message

- **Flooding**: Received **PINGS** and **QUERIES** must be forwarded to all connected Gnodes
- **PINGS** or **QUERYS** with the same FUNCTION ID and GNODE ID as previous messages are destroyed (avoid loops)

- **PONG** and **QUERY HIT are forwarded to the** origin of the corresponding PING or QUERY

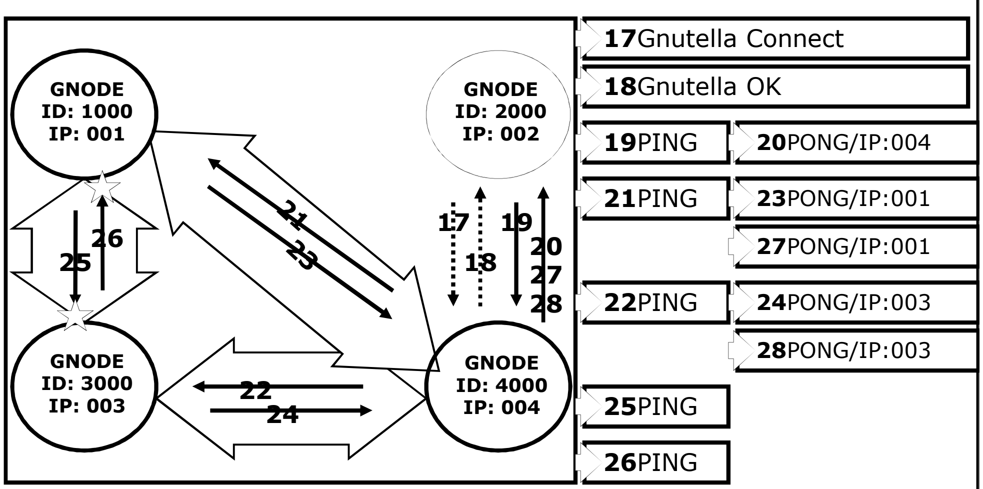
FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-23

FP2P13F, javed@kent.edu
Javed I. Khan@2008

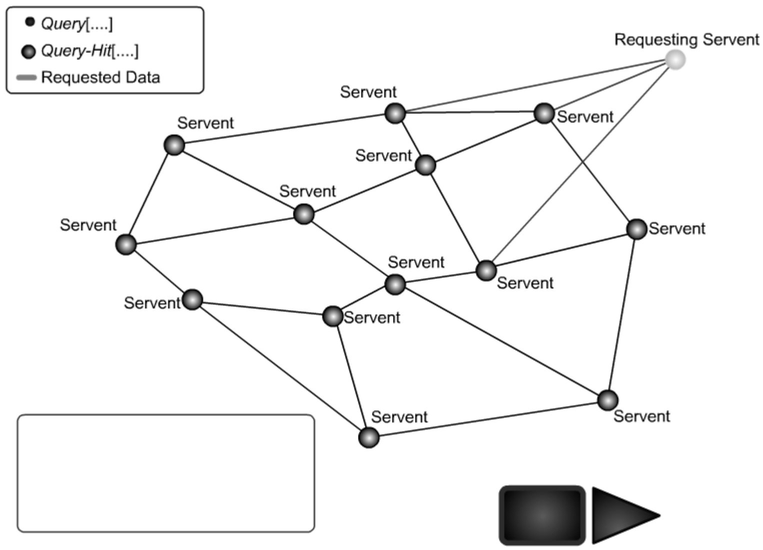
# Gnutella Connection Setup

Gnode 2000 establishes a connection to 4000




Javed I. Khan@2008

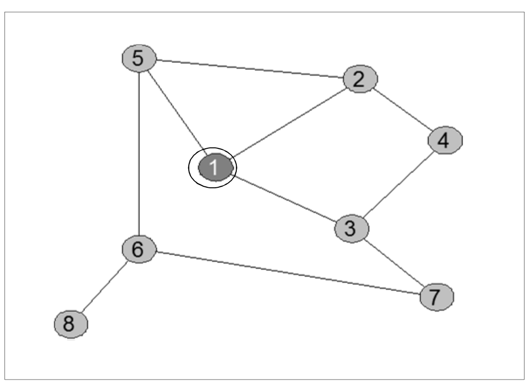
# Gnutella Wrap-Up




FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-07, S-25
FP2P13F, javed@kent.edu
Javed I. Khan@2008

# Messaging Overhead

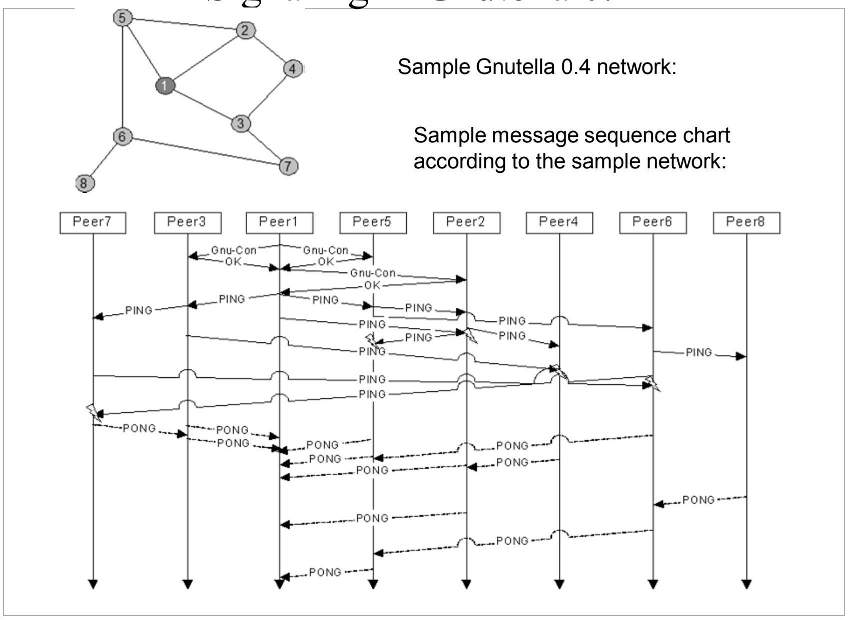



- PING 23 bytes
- PONG 37 bytes
- GnuCon+OK 34 bytes
- Huge overhead.
- Non Optimum Physical/Overlay mapping

FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-07, S-26
FP2P13F, javed@kent.edu
Javed I. Khan@2008

# Signaling in Gnutella 0.4




FOUNDATION OF PEER-TO-PEER SYSTEMS



# Firewall (PUSH)

- The server behind a firewall may not know initially that someone is trying to contact it. However, the transfer can be initiated by the server. But how does the server know that it has been selected by the query originator and it is expected to build this TCP connection?
- The PUSH message is used by the downloader when a chosen server (the node which has the file) is behind a firewall.
- The PUSH message has four fields in its payload- the port and IP address or the downloader, the GUID of the chosen server and the file index of the file found in the QUERY HIT message.
- When a Gnutella node is behind a firewall, then it too generates the same QUERY-HIT responses that it would have generated had it not been so.

FOUNDATION OF PEER-TO-PEER SYSTEMS

## Climbing Firewall (contd..)

- To get around this problem, the client uses the message PUSH. When the downloading client is unable to contact the server it wants to download from, then it sends the special PUSH request.
- It is forwarded across the network exactly like the PING.
- Each node that receives this PUSH checks the GUID indicated in payload to verify it is meant for itself. If it finds that the message is meant for itself, it builds a TCP connection to the client that initiated the push and proceeds to upload the requested file.





## Direct & Browsing Ping

- Gnutella has two special provisions called "direct" and "browsing" PINGs. A "direct" PING message has TTL=1, and Hop Count=0. A "browsing" PING message has TTL=2, and Hop Count=0.
- Each servant should respond (at least once for each connected remote servant) with a valid PONG answer about itself in response to an incoming "direct" PING request signaled by TTL=1 and Hop Count=0. Naturally, it will not be propagated further. It is used to query ones' immediate neighbor.
- In reply to an incoming "browsing" PING request (with TTL=2 and Hop Count=0) a neighbor SHOULD return PONGs from the list of their currently connected (or recently cached) accessible neighbor servants. In a way it is a browsing of a neighbor's routing table.

FOUNDATION OF PEER-TO-PEER SYSTEMS

# Pong Cache

## Pong Cache

Nodes may implement Pong Cache to reduce network traffic.

- For each connection an array (about 10) of Pong Messages are stored. When a pong comes in, it overwrites the oldest stored pong for the connection. (Note there many be many more nodes along a connection. Only 10 recent ones are kept.)
- The information that must be stored for each pong is: * IP Address * Port number * Number of files shared * Number of kilobytes shared * GGEP extension block (if present) * Hops value, and the time it received the PONG.
- When a Ping message, called P, is received over connection C, and it has been at least one second since last time a ping was received over C, the servant will return several Pongs (10 for example) from all of its stored pongs.

## Pong Cache

- The Pongs will be picked from all connections except from C, since it would be useless sending pongs back where they came from.
- Not all cached Pongs are not sent in response to every Ping. A good idea is to pick pongs from different connections and with varying stored Hops values.
- A replying servant should also return a Pong with information about itself, if it can accept incoming connections.
- The outgoing pong will have the same message GUID as P, not the message GUID it had when the pong was received.
- The Hops is set to the stored hops value + 1, and TTL so that TTL+Hops=7.





## Keeping Cache Updated

- If the TTL in stored Pong is less than P's Hops value, the current stored Pong will not be sent. This also means that Pongs whose Hops value already is 7 will not be propagated any further.
- To keep the cache fresh, a ping (TTL=7, Hops=0) is sent over all connections at small interval (like every 3 seconds).
- This look like very often, but remember that the neighbour servents will just respond with pongs from its own cache. The short time ensures that pongs are always fresh.
- To neighbour hosts who has not indicated that they support pong caching (using the Pong-Caching handshaking header), one ping per minute might be a better number.

FOUNDATION OF PEER-TO-PEER SYSTEMS

# Performance
# Of Gnutella 4.0

Ripeanu, Foster & Iamnitchi Study, 2000-2001

## Network Growth

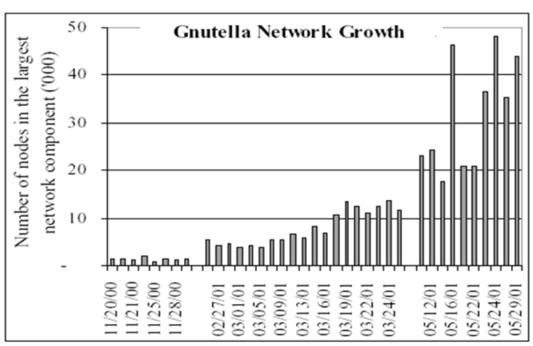



**Figure 1**: Gnutella network growth. The plot presents the number of nodes in the largest connected component in the network. Data collected during Nov. 2000, Feb./March 2001 and May 2001. We found a significantly larger network around Memorial Day (May 24-28) and Thanksgiving 2000, when apparently more people hunt for shared music online.

- Gnutella's failure to scale has been predicted during this time. But it grew 25 times in six months period! (they studied 400,000+ live nodes)

FOUNDATION OF PEER-TO-PEER SYSTEMS

## Traffic Type

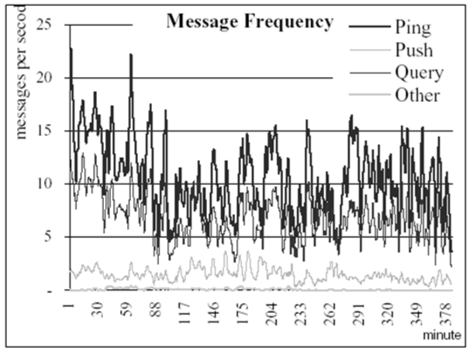



**Figure 2:** Generated traffic (messages/sec) in Nov. 2000 classified by message type over a 376 minute period. Note that overhead traffic (PING messages, that serve only to maintain network connectivity) formed more than 50% of the traffic. The only 'true' user traffic is QUERY messages. Overhead traffic has decreased by May 2001 to less than 10% of all generated traffic.

FOUNDATION OF PEER-TO-PEER SYSTEMS



## Scalability

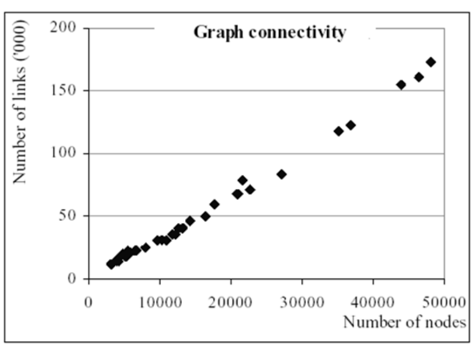



**Figure 4**: Average node connectivity. Each point represents one Gnutella network crawl. Note that, as the network grows, the average number of connections per node remains constant (average node connectivity is 3.4 connections per node).

- Among 95% of the nodes are in largest connected component. About 40% of the nodes leave the network in less than 4 hours, while only 25% of the nodes are for more than 24 hours.

FOUNDATION OF PEER-TO-PEER SYSTEMS

## Connectivity Distribution

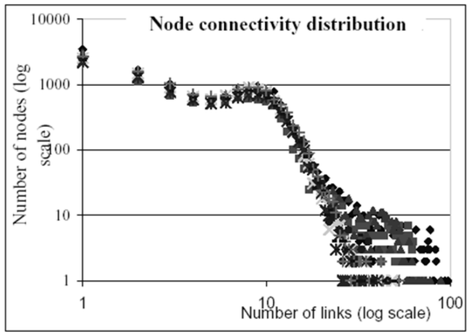



**Figure 6:** Connectivity distributions during March 2001. Each series of points represents one Gnutella network topology discovered during March 2001. Note the log scale on both axes. Networks crawled during May/June 2001 show a similar pattern.

- **Although data are noisy (due to the small size of the networks), we can easily recognize the signature of a power-law distribution: the connectivity distribution appears as a line on a log-log plot. [6,4] confirm that early Gnutella networks were power-law.**

FOUNDATION OF PEER-TO-PEER SYSTEMS



## Discussion

- Disadvantages
  - High signaling traffic, because of decentralization
  - Modem nodes may become bottlenecks
  - Overlay topology not optimal, as
    - no complete view available,
    - no coordinator
  - If not adapted to physical structure delay and total network load increases
    - Zigzag routes
    - loops
- Advantages
  - No single point of failure
  - Can be adapted to physical network
  - Can provide anonymity
  - Can be adapted to special interest groups
- Application areas
  - File-sharing
  - Context based routing (see chapter about mobility)

FOUNDATION OF PEER-TO-PEER SYSTEMS