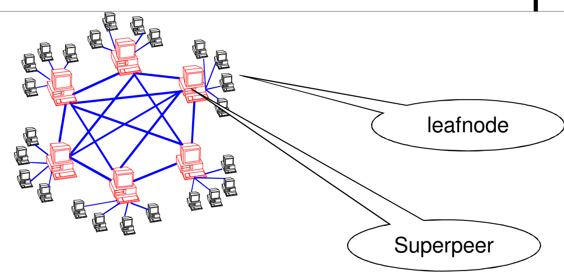
Definition of Hybrid P2P

FOUNDATION OF PEER-TO-PEER SYSTEMS

- Main characteristic, compared to pure P2P: Introduction of another dynamic hierarchical layer
- Hub based network
- Reduces the signaling load without reducing the reliability
- Election process to select an assign Superpeers
- Superpeers: high degree (degree>>20, depending on network size)
- Leafnodes: connected to one or more Superpeers (degree<7)



Model of Hybrid P2P Networks

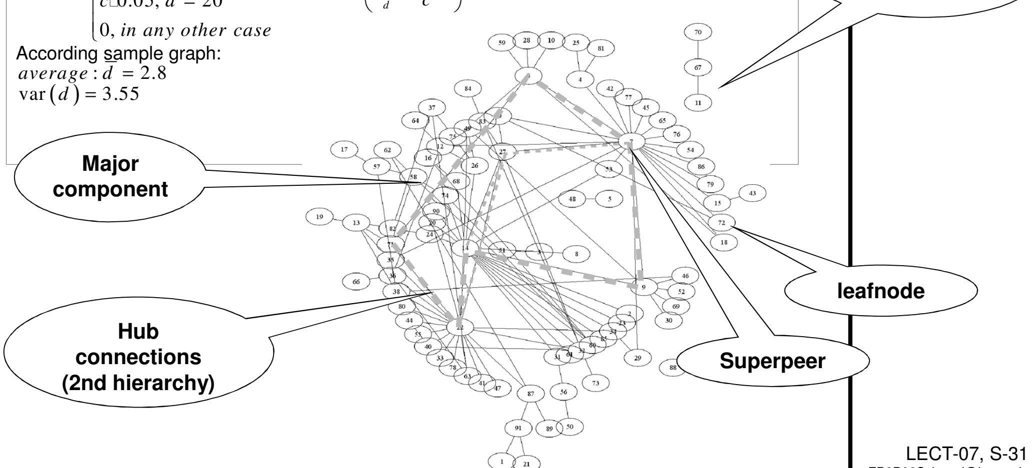
Degree distribution:

$$p(d) = \begin{cases} c \cdot d^{-1.4}, 1 < d \leq 7 \\ c \cdot 1^{-1.4} - 0.05, d = 1 \\ c \cdot 0.05, d = 20 \\ 0, \textit{in any other case} \end{cases}, \textit{with } c = \left(\sum_d \frac{p(d)}{c} \right)^{-1}$$

According sample graph:

$average: d = 2.8$

$\mathrm{var}(d) = 3.55$

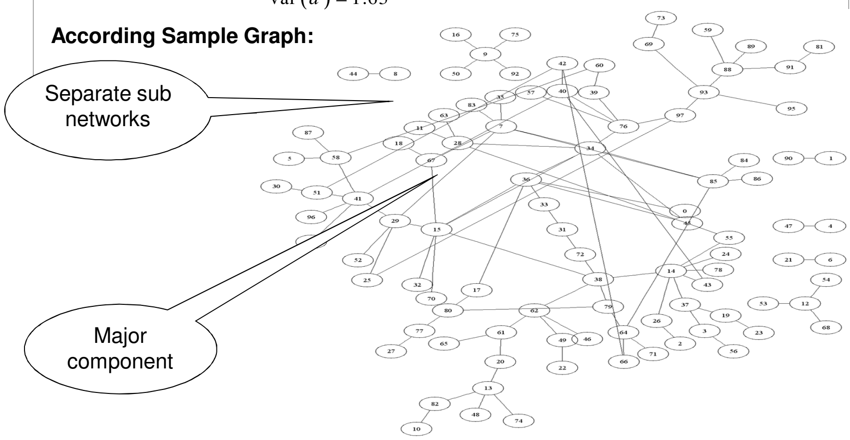


Flashback: Degree Distribution in Pure P2P Networks

Degree distribution:
$$p(d) = \begin{cases} c \cdot d^{-1.4}, & 0 < d \leq 7 \\ 0, & in\ any\ other\ case \end{cases}, \quad with\ c = \left(\sum_{d} \frac{p(d)}{c} \right)^{-1}$$

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

According Sample Graph:



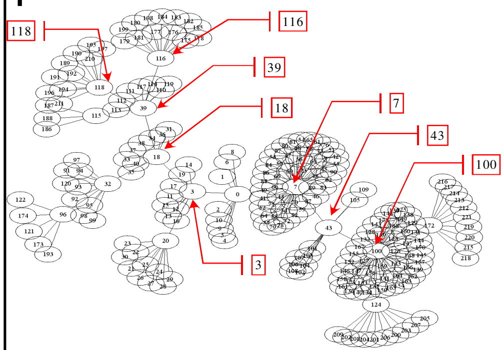
FOUNDATION OF PEER-TO-PEER SYSTEMS

Basic Characteristics of Hybrid P2P

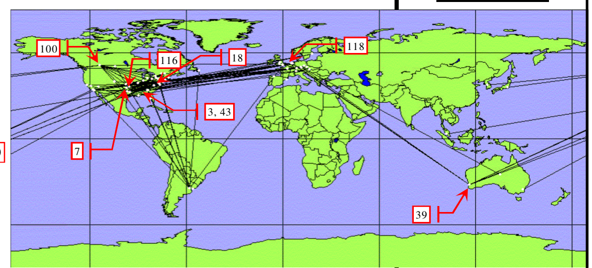
- Bootstrapping:
  - Via bootstrap-server (host list from a web server)
  - Via peer-cache (from previous sessions)
  - Via well-known host
  - Registration of each leafnode at the Superpeer it connects to, i.e. it announces its shared files to the Superpeer
- Routing:
  - Partly decentralized
    - Leafnodes send request to a Superpeer
    - Superpeer distributes this request in the Superpeer layer
    - If a Superpeer has information about a matching file shared by one of its leafnodes, it sends this information back to the requesting leafnode (backward routing)
  - Hybrid protocol (reactive and proactive): routes to content providers are only established on demand; content announcements from leafnodes to their Superpeers
  - Requests: flooding (limited by TTL and GUID) in the Superpeer layer
  - 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 (directly between leafnodes)

FOUNDATION OF PEER-TO-PEER SYSTEMS

Topology of Hybrid P2P



Abstract network structure of a part of the Gnutella network (222 nodes Geographical view given by Figure on the right, measured on 01.08.2002



Geographical view of a part of the Gnutella network (222 nodes); The numbers depict the node numbers from the abstract view (Figure on the left, measured on 01.08.2002)

- Virtual network not matched to physical network. See path from node 118 to node 18.
- Superpeer (hub) structure clearly visible in abstract view

LECT-07, S-34
FP2P08S, javed@kent.edu
Javed I. Khan@2008

Example: Gnutella 0.6

Example: Gnutella 0.6

- Program for sharing files over the Internet
- Focus:
  - decentralized method of searching for files
  - Higher signaling efficiency than Pure P2P
  - Same reliability (no single point of failure)
- Basis of most file-sharing applications (not BitTorrent)
- Brief History:
  - **Spring 2001**: resulted from Gnutella 0.4 by 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,…)

FOUNDATION OF PEER-TO-PEER SYSTEMS

Gnutella 0.6 Network Organization

New connection/network setup

- Upon connection to the network via a Superpeer, each node is a leafnode
- It announces its shared content to the Superpeer it connected to
- Superpeer thus updates its routing tables
- Election mechanism decides which node becomes a Superpeer or a leafnode (depending on capabilities (storage, processing power) network connection, the uptime of a node,…), if
  - Too many nodes are connected to one Superpeer
  - A Superpeer leaves the network
  - To less nodes are connected to a Superpeer

FOUNDATION OF PEER-TO-PEER SYSTEMS

Concept: Ultra Peers

- It is a scheme to have a hierarchical Gnutella network by categorizing the nodes on the network as leaves and ultrapeers. An ultrapeer acts as a proxy to the Gnutella network for the leaves connected to it.

- This has an effect of making the Gnutella network scale, by reducing the number of nodes on the network involved in message handling and routing, as well as reducing the actual traffic among them.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Ultrapeer Election

- Since Gnutella is a decentralized system, ultrapeers are **elected** without the use of a central server. It is up to each node to determine if it is to become an ultrapeer or a shielded leaf node.

- Some Basic Requirements:
  - Not firewalled.
  - Sufficient downstream and upstream bandwidth.
  - Sufficient uptime
  - Sufficient RAM and CPU speed.

- If the above criterion are met, a node is said to be ultrapeer capable. When either an ultrapeer capable node will actually become an ultrapeer depends on if there is need for more ultrapeers on the network, and on how well the above criterion are met.

Ultrapeer Messages

- Ultrapeer capabilities and information are exchanged during the handshaking sequence when trying to establishing a new Gnutella connection. The following new headers are used in handshake:

- X-Ultrapeer: "True"
  - signals that node is an ultrapeer, "False" signals that the node wants to be a shielded leaf node.
- X-Ultrapeer-Needed:
  - Used to balance the number of ultrapeers.
- X-Try-Ultrapeers:
  - contains only addresses of ultrapeers.
- X-Query-Routing:
  - Signals support for the Query Routing Protocol

FOUNDATION OF PEER-TO-PEER SYSTEMS

Handshake Messages

- A leaf is trying to connect to a Ultrapeer.

Leaf	Ultrapeer
GNUTELLA CONNECT/0.6 User-Agent: LimeWire/1.0 X-Ultrapeer: False X-Query-Routing: 0.1	
	GNUTELLA/0.6 200 OK User-Agent: LimeWire/1.0 X-Ultrapeer: False X-Ultrapeer-Needed: False X-Query-Routing: 0.1 X-Try: 24.37.144:6346, 193.205.63.22:6346 X-Try-Ultrapeers: 23.35.1.7:6346, 18.207.63.25:6347
GNUTELLA/0.6 200 OK	

- The leaf is now a shielded node of the ultrapeer. The leaf should drop any non ultrapeer connections and send a QRP routing table (assuming QRP is used).

FOUNDATION OF PEER-TO-PEER SYSTEMS

Example Handshake Messages

- A leaf is trying to connect to another leaf.

New Leaf	Existing Leaf
GNUTELLA CONNECT/0.6 X-Ultrapeer: False	
	GNUTELLA/0.6 503 I am a leaf X-Ultrapeer: False X-Try: 24.37.144:6346 X-Try-Ultrapeers: 23.35.1.7:6346
	[DROP CONNECTION]

- If a shielded leaf node receives a connection request, it will refuse to accept the connection by returning a 503 error code together with X-Try and X-Try-Ultrapeer headers to redirect the remote host to other addresses.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Example Handshake Messages

- A leaf is trying to connect to another leaf.

New Leaf	Existing Leaf
GNUTELLA CONNECT/0.4 X-Ultrapeer: False	
	GNUTELLA/0.6 200 OK X-Ultrapeer: False
GNUTELLA/0.4 200 OK	

- Sometimes nodes will be ultrapeer-incapable but unable to find an ultrapeer. In this case, they behave exactly like old, unrouted Gnutella 0.4 connections.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Example Handshake Messages

- When two ultrapeers meet, both set X-Ultrapeer: true.

Ultrapeer A	Ultrapeer B
GNUTELLA CONNECT/0.6 X-Ultrapeer: True	
	GNUTELLA/0.6 200 OK X-Ultrapeer: True
GNUTELLA/0.6 200 OK	

- If both have leaf nodes, they will remain ultrapeers after the interaction. No QRP route table is sent between ultrapeers.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-44
FP2P08S, javed@kent.edu
Javed I. Khan@2008

Example Handshake Messages

- Sometimes there will be too many ultrapeer-capable nodes on the network. Consider the case of an ultrapeer A connecting to an ultrapeer B.

Ultrapeer A	Ultrapeer B
GNUTELLA CONNECT/0.6 X-Ultrapeer: True	
	GNUTELLA/0.6 200 OK X-Ultrapeer: True X-Ultrapeer-Needed: False
GNUTELLA/0.6 200 OK X-Ultrapeer: False	

- If B doesn't have enough leaves, it may direct A to become a leaf node. If A has no leaf connections, it stops fetching new connections, drops any Gnutella 0.4 connections, and sends a QRP table to B. Then B will shield A from all traffic. If A has leaf connections, it ignores the guidance, as in the above case.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-45
FP2P08S, javed@kent.edu
Javed I. Khan@2008

Query Routing Protocol (QRP)

- The Query Routing Protocol (QRP for short) is an essential part of the Ultrapeer specification: it governs how the Ultrapeer will filter queries and only forward those to the leaf nodes most likely to have a match.
- This is done without even knowing the resource names, by looking the query words through a big hash table, that is sent by the leaf node to its Ultrapeer.
- The aim of the QRP is to avoid forwarding a query that cannot match, it is not to forward only those queries that will match.

FOUNDATION OF PEER-TO-PEER SYSTEMS

QRP Leaf Node Role

- Break all the resource names into individual words. A word is made of a consecutive sequence of letters and digits.
- Hash each word with a well-known hash function and insert a "present" flag in the corresponding hash table slot. Note that this hash table is a big array, and we don't store the key, only the fact that a key ended up filling some slot. All words are lower-cased and all accents are removed from them, i.e. "déjà" is transformed into "deja", so that only ASCII characters remain. Only those words that are made of at least 3 letters are retained.
- All words are re-hashed with their trailing 1, 2, or 3 letters removed, provided the word length after such trimming is at least 3 letter long. This is a simple attempt to remove plural from words. Optionally, nodes can chop off more letters from the end, provided that each hashed word is at least 3 character long.
-
- The "boolean vector" built at later stage is optionally compressed, broken up in small messages, and sent mixed with regular Gnet traffic to the ultrapeer.

FOUNDATION OF PEER-TO-PEER SYSTEMS

QRP Ultrapeer Role

- Until the whole "boolean vector" is received from a leaf node, all queries are forwarded to that node.
- When the "boolean vector" is fully received, it is going to be used as the Query Routing table for that leaf node: queries are broken into individual words, all accentuated letters are removed.
- For each leaf node with a Query Routing table:
  - Each word is then hashed and looked up in the Query Routing table.
  - Depending on the query matching rules either ALL the words will be required to be found in the Query Routing, or only some of them, to declare a Query Routing Hit.
  - Only those queries that were declared a Hit at the previous stage will be forwarded to a given leaf node.

FOUNDATION OF PEER-TO-PEER SYSTEMS

QRP Messages

- ROUTE_TABLE_UPDATE (0x30), Reset variant (0x0): to clear the routing table and to set a new routing table for one leafnode

Variant	Table_Length	Infinity

ield Name	Bytes	Meaning
VARIANT	1	The message variant. Always 0x0 for RESET.
TABLE_LENGTH	4	The length of the sender's route table, i.e., the number of entries. *(Earlier versions of this document incorrectly stated the meaning of this value.)* For hashing reasons, this *must be a power of 2.*
INFINITY	1	The route table value for infinity, i.e., the maximum distance to any file in the table+1.

FOUNDATION OF PEER-TO-PEER SYSTEMS

QRP Messages

- ROUTE_TABLE_UPDATE (0x30), Patch variant(0x1): to update and set a new routing table with a certain number of entries (e.g. new shared files)

0	1	2	3	4	5 n+4
Variant	Seq_No	Seq_Size	Compressor	Entry_Bits	DATA

Field Name	Bytes	Meaning
VARIANT	1	The message variant. Always 0x1 for PATCH.
SEQ_NO	1	The position of this message in the update sequence.
SEQ_SIZE	1	The total number of messages in this update sequence.
COMPRESSOR	1	The algorithm to use when decompressing data. Currently defined values: 0x0 no compression, 0x1 ZLIB compression
ENTRY_BITS	1	The number of bits per uncompressed patch entry, including the sign bit. Must be 4 or 8.
DATA	to end	The compressed table patch.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-50
FP2P08S, javed@kent.edu
Javed I. Khan@2008

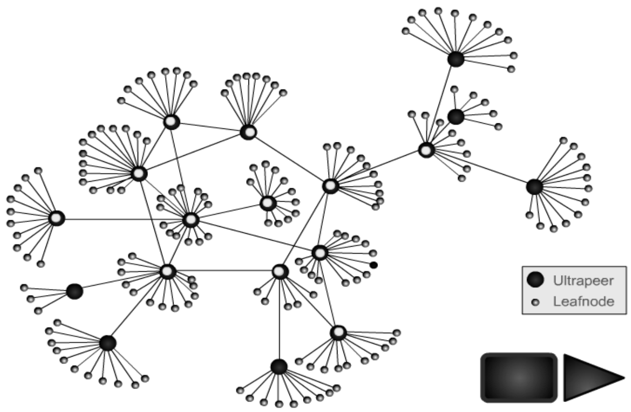
Gnutella 0.6 Routing

- Content requests:
  - Leafnode sends request to Superpeer
  - Superpeer looks up in its routing tables whether content is offered by one of its leafnode. In this case the request is forwarded to this node.
  - Additionally the Superpeer increases the hopcounter and forwards this request to the Superpeers it is connected to.
  - To enable backward routing, the peer has to store the GUID of the message connected to the information from which peer it received the request in the previous hop
  - If a Superpeer receives such a request from another Superpeer, this request is handled the same way, as if it would have received it from one of its leafnodes
  - After the hopcounter of the request reaches the TTL-value it is not forwarded any further (prevent circles)
- Content responses:
  - If a leafnode receives a request, it double-checks whether it shares the file (should be the case, as long as the routing tables of the Superpeer are correct)
  - In case of success, the leafnode sends a content reply back to the requesting peer, by sending it back to that node (Superpeer) it received the message from (backward routing)
  - Hop by hop the message can thus be routed back to the requesting node
- Content exchange:
  - Directly between the leafnodes, via HTTP connections

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-07, S-51
FP2P08S, javed@kent.edu
Javed I. Khan@2008

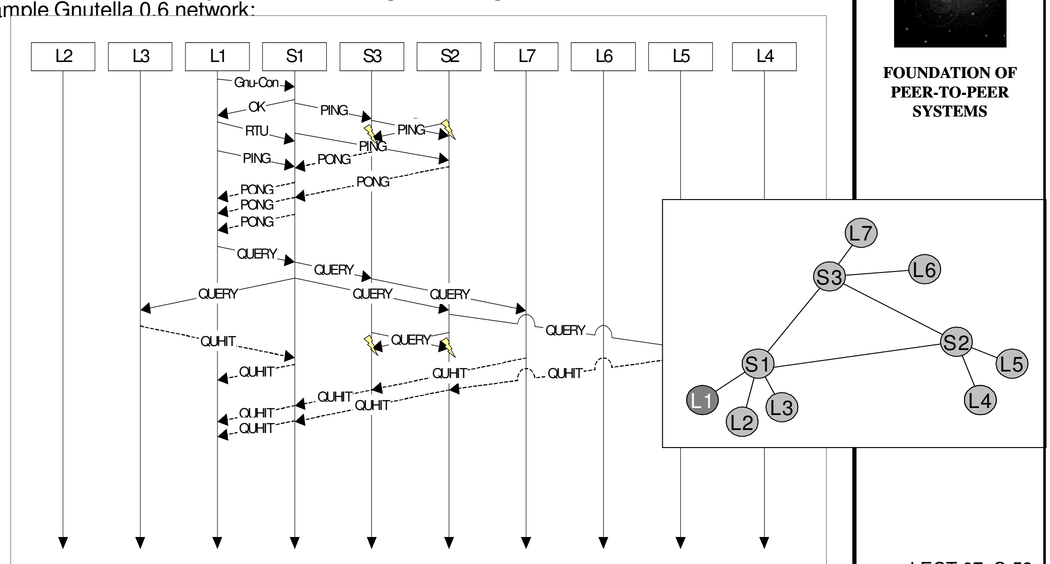
Gnutella 0.6: How Does It Work



FOUNDATION OF PEER-TO-PEER SYSTEMS

Summary of the Signaling in Gnutella 0.6

Sample Gnutella 0.6 network:



FOUNDATION OF PEER-TO-PEER SYSTEMS

Advanced Topic: Performance

Ripeanu, Foster & Iamnitchi Study, 2000-2001

Network Growth

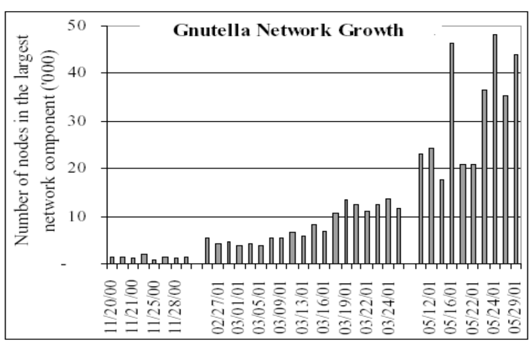


Figure 1: Gnutella network growth. The plot presents the number of nodes in the largest connected component in the network. Data collected during Nov. 2000, Feb./March 2001 and May 2001. We found a significantly larger network around Memorial Day (May 24-28) and Thanksgiving 2000, when apparently more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

LECT-07, S-55
FP2P08S, javed@kent.edu
Javed I. Khan@2008

Traffic Type

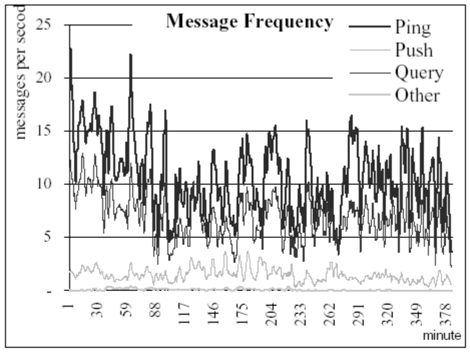


Figure 2: Generated traffic (messages/sec) in Nov. 2000 classified by message type over a 376 minute period. Note that overhead traffic (PING messages, that serve only to maintain network connectivity) formed more than 50% of the traffic. The only 'true' user traffic is QUERY messages. Overhead traffic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

FOUNDATION OF PEER-TO-PEER SYSTEMS

Scalability

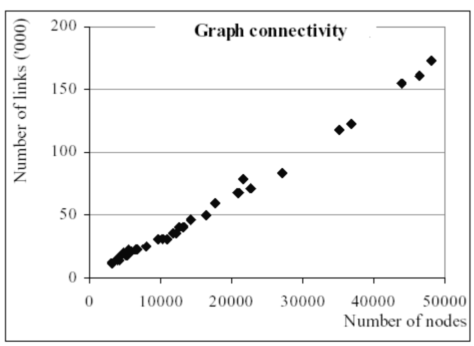


Figure 4: Average node connectivity. Each point represents one Gnutella network crawl. Note that, as the network grows, the average number of connections per node remains constant (average node connectivity is 3.4 connections per node).

- 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

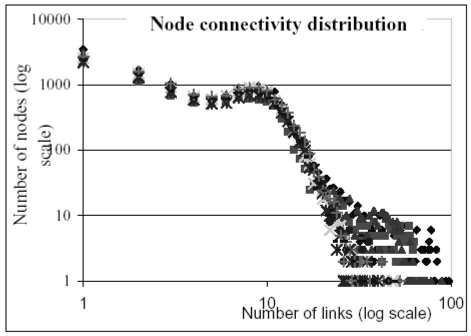


Figure 6: Connectivity distributions during March 2001. Each series of points represents one Gnutella network topology discovered during March 2001. Note the log scale on both axes. Networks crawled during May/June 2001 show a similar pattern.

- **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.**

Further P2P systems based on hybrid P2P

- Edonkey
- Kazaa/FastTrack
- Emule
- OpenNap
- …

Discussion

- Disadvantages
  - Still High signaling traffic, because of decentralization
  - No definitive statement possible if content is not available or not found [dealing with incomplete information, seti, Asrar's work- javed]
  - 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
  - Can not be adapted to physical network completely because of hub structure
  - Asymmetric load (Superpeers have to bear a significantly higher load)
- Advantages
  - No single point of failure
  - Can provide anonymity
  - Can be adapted to special interest groups
- Application areas [p2p techniques are becoming a layer than application-javed]
  - File-sharing
  - Context based routing (see chapter about mobility)

Summary

- P2P technologies offer an innovative overlay infrastructure for decentralized and distributed systems
- Due to the distributed nature, the signaling load is very high, but it can be reduced with introduction of hierarchies, compression and geo-sensitive protocols
- Advantages:
  - Simple basic principle
  - Enhanced reliability
  - Redundancy (high replication rate)
  - Unsusceptible against Denial of Service attacks (DOS)
  - No single point of failure
  - No central instances/administration
  - Direct and instantaneous communication possible
  - Large variety of applications possible