Unstructured Routing : Gnutella and Freenet

Presented By
Matthew, Nicolai, Paul

Presentation Overview

- Gnutella
  1. What Gnutella is
  2. How it works
  3. Its positives and negatives

- Freenet
  1. Motivation and Philosophy
  2. Architecture and use
  3. Performance, Strengths and Weaknesses

What is Gnutella?

- Gnutella is a protocol for distributed search
- Each node in a Gnutella network acts as both a client and server
- Peer to Peer, decentralized model for file sharing
- Any type of file can be shared
- Nodes are called “Servents”

What do Servents do?

- Servents "know" about other Servents
- Act as interfaces through which users can issue queries and view search results
- Communicate with other Servents by sending "descriptors"

Descriptors

- Each descriptor consists of a header and a body.
- The header includes (among other things)
  - A descriptor ID number
  - A Time-To-Live number
- The body includes:
  - Port information
  - IP addresses
  - Query information
  - Etc… depending on the descriptor

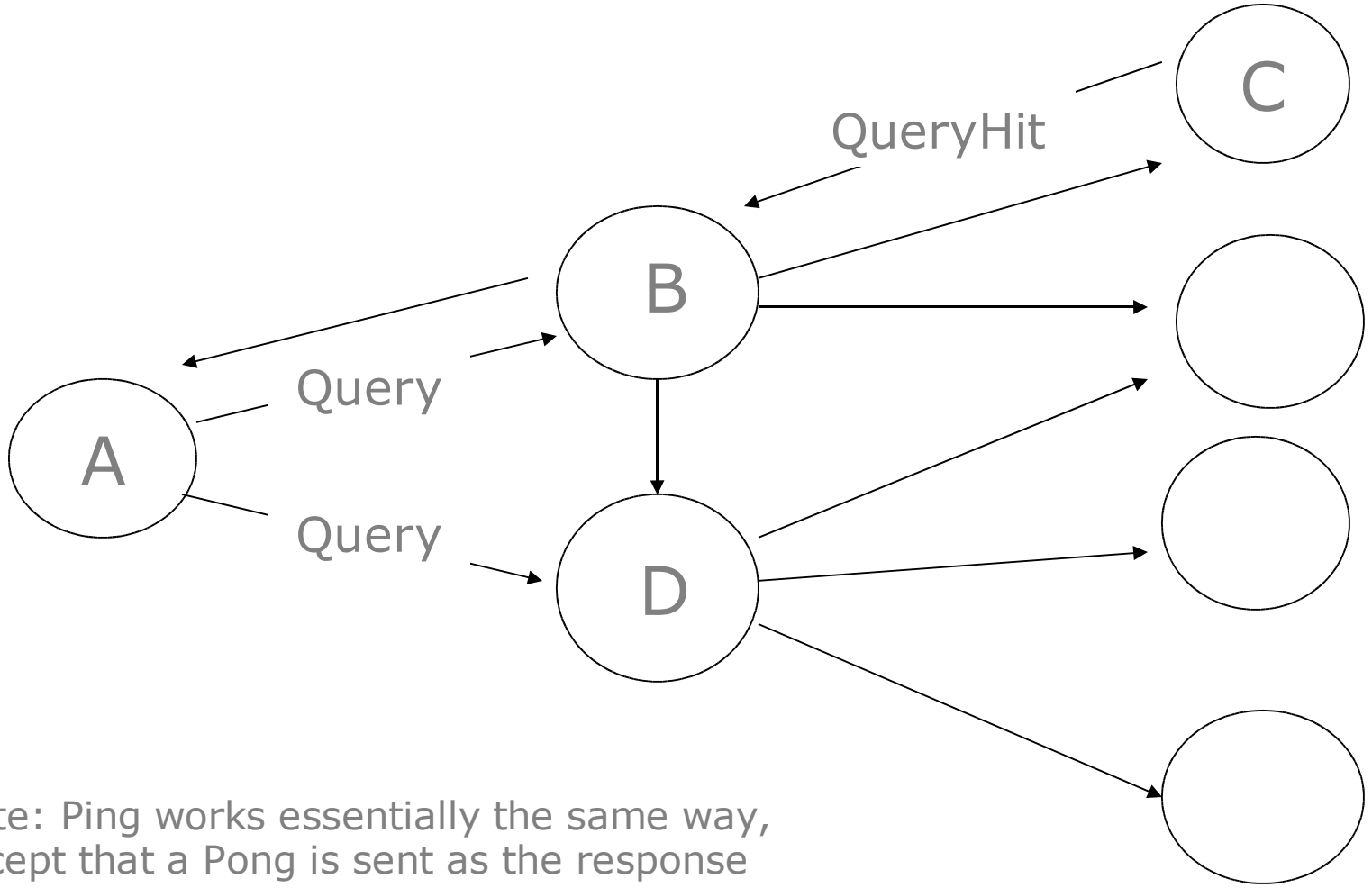
Gnutella Descriptors

- **Ping**: Used to discover hosts on the network.
- **Pong**: Response to a Ping
- **Query**: Search the network for data
- **QueryHit**: Response to a Query. Provides information used to download the file
- **Push**: Special descriptor used for sharing with a firewalled servent

Routing

- Node forwards Ping and Query descriptors to all nodes connected to it
- Except:
  - If descriptor's TTL is decremented to 0
  - Descriptor has already been received before
- Loop detection is done by storing Descriptor ID's
- Pong and QueryHit descriptors retrace the exact path of their respective Ping and Query descriptors

Routing2



Note: Ping works essentially the same way, except that a Pong is sent as the response

Joining a Gnutella Network

- Servent connects to the network using TCP/IP connection to another servent.
- Could connect to a friend or acquaintance, or from a “Host-Cache”.
- Send a **Ping** descriptor to the network
- Hopefully, a number of **Pongs** are received

Querying

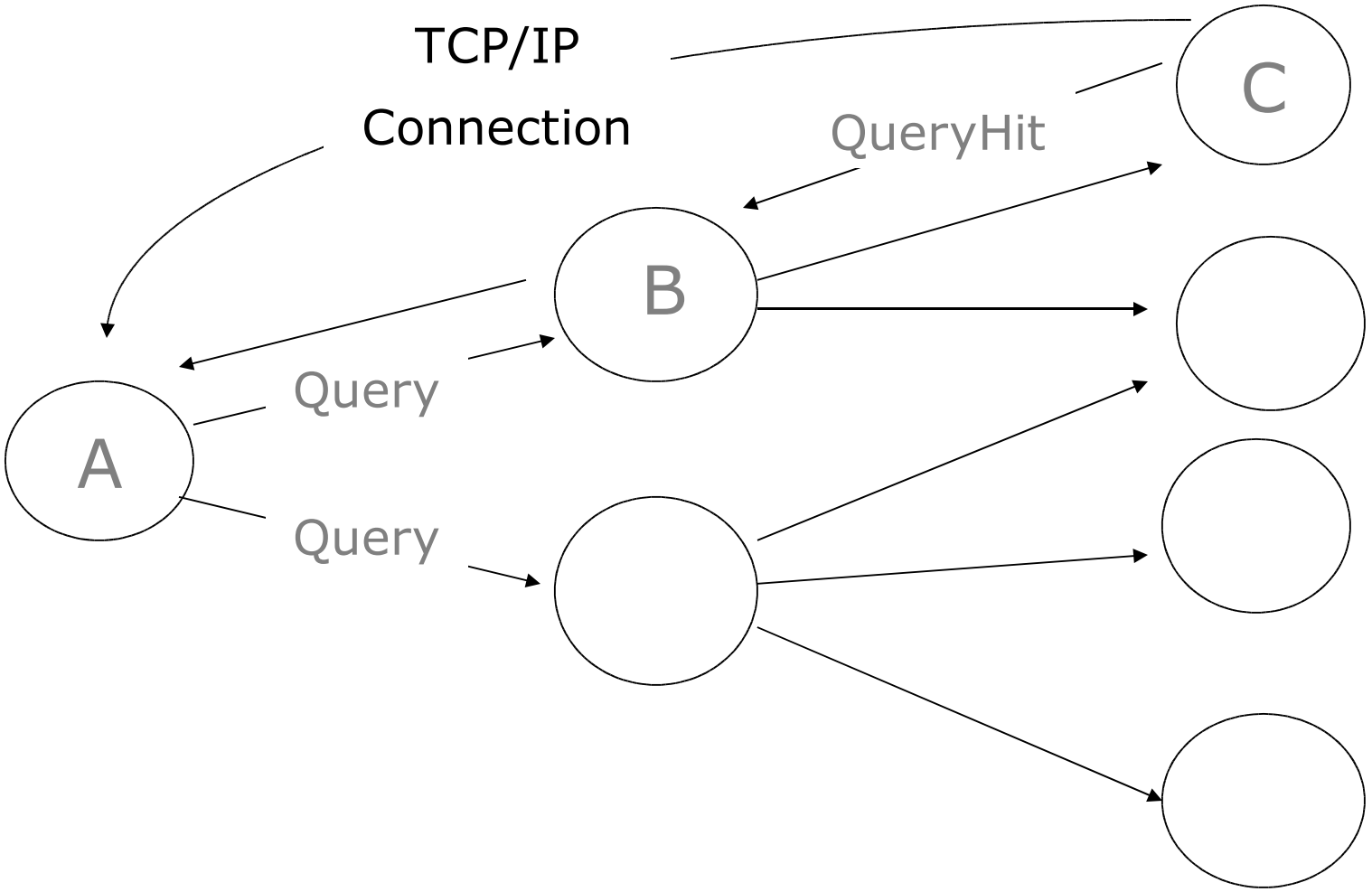
- Servent sends **Query** descriptor to nodes it is connected to.
- Queried Servents check to see if they have the file.
  - If query match is found, a **QueryHit** is sent back to querying node

Downloading a File

- File data is never transferred over the Gnutella network.
- Data transferred by direct connection
- Once a servent receives a QueryHit descriptor, it may initiate the direct download of one of the files described by the descriptor's Result Set.
- The file download protocol is HTTP. Example:

```
GET /get/<File Index>/<File Name>/ HTTP/1.0\r\n
Connection: Keep-Alive\r\n
Range: bytes=0-\r\n
User-Agent: Gnutella\r\n3
```

Direct File Download



Overall:

- Simple Protocol
- Not a lot of overhead for routing
- Robustness?
  - No central point of failure
  - However: A file is only available as long as the file-provider is online.
- Vulnerable to denial-of-service attacks

Overall 2:

- Scales poorly: Querying and Pinging generate a lot of unnecessary traffic
- Example:
  - If TTL = 10 and each site contacts six other sites
  - Up to 10^6 (approximately 1 million) messages could be generated.
  - On a slow day, a GnutellaNet would have to move *2.4 gigabytes per second* in order to support numbers of users comparable to Napster. On a heavy day, *8 gigabytes per second* (Ritter article)
- Heavy messaging can result in poor performance

Final thoughts about Gnutella

- Gnutella developers acknowledge the problems with Gnutella
- Gnutella2 (Mike's protocol) is now released, but it is substantially different from original Gnutella
- Gnutella2 is not compatible with original
- Some say Gnutella2 is attempt to hijack Gnutella

Freenet

Freenet

- What is Freenet ?
  - A Decentralized Distributed File Storage System
- How does it work ?
  - Files stored and replicated across a distributed network environment, with a peer-to-peer query and data access system. No centralized system management.

Freenet

- Motivation – What does it provide ?
  - Anonymity for both producers and consumers of information
  - Deniability for storers of information
  - Resistance to attempts by third parties to deny access to information
  - Efficient dynamic storage and routing of information
  - Decentralization of all network functions
    - From *"Freenet: A Distributed anonymous Information Storage and Retrieval System"*, Ian Clarke et. al.

Freenet

- Architecture
  - Key generation
  - Distributed information storage
  - Query procedure
  - Data retrieval
  - Data removal

Freenet

- Architecture (2)
  - Location independence
  - Transparent lazy replication
  - File encryption
  - Dynamic network expansion/contraction

Freenet

- Routing

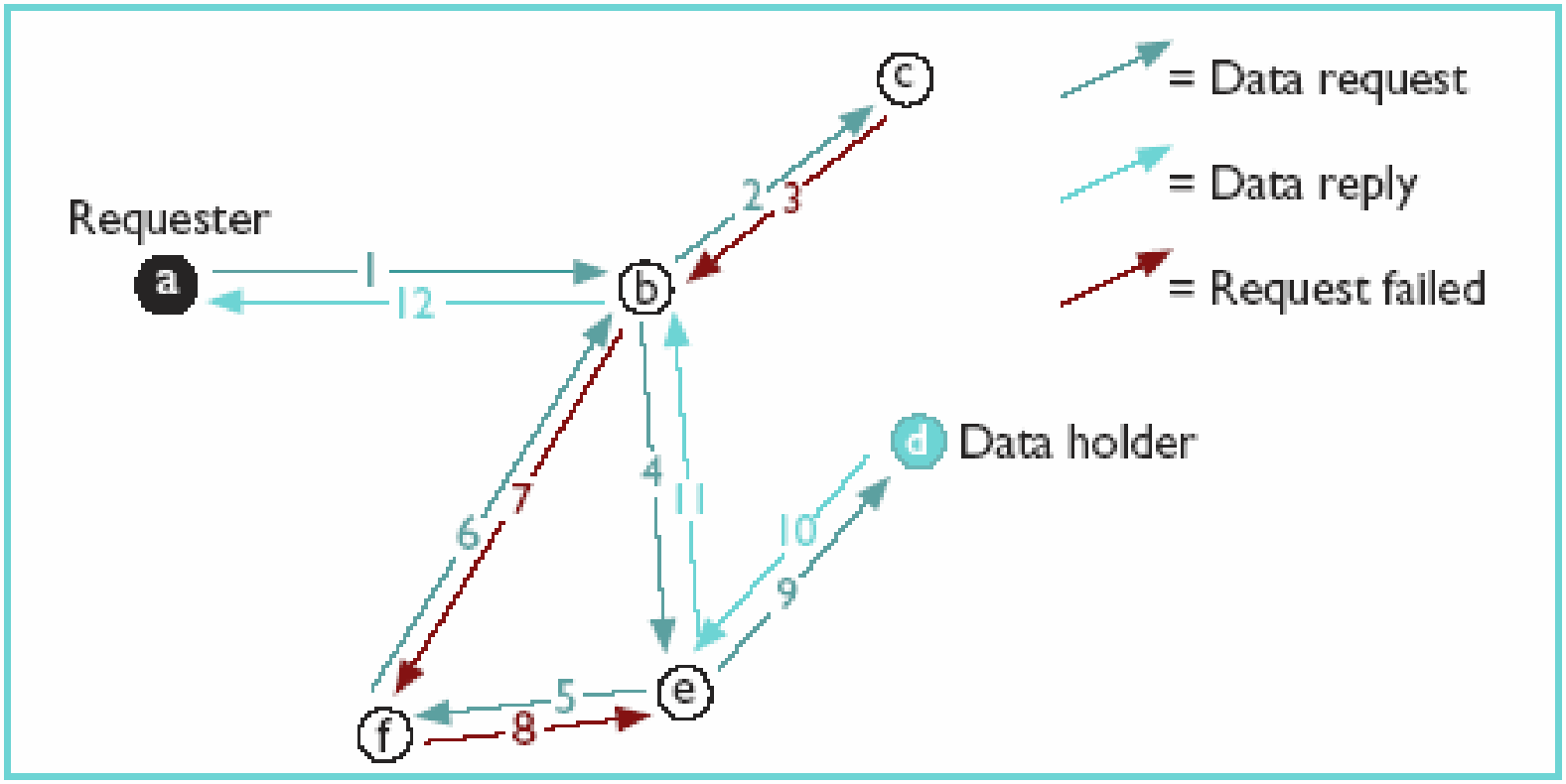


Figure 1. Typical request sequence. The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

Freenet

- Lookup / Insert

1. Hash key for data (160-bit SHA-1)
2. Find node with closest match
3. Forward query to this node
4. Return data, replicating along the way
5. For insert, push data onto node

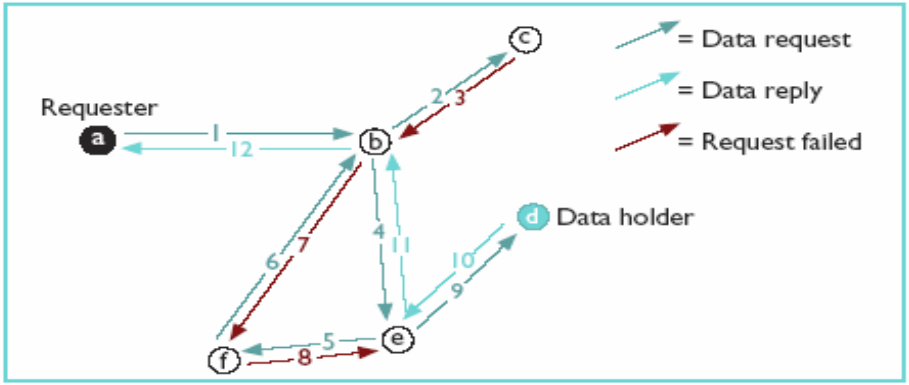
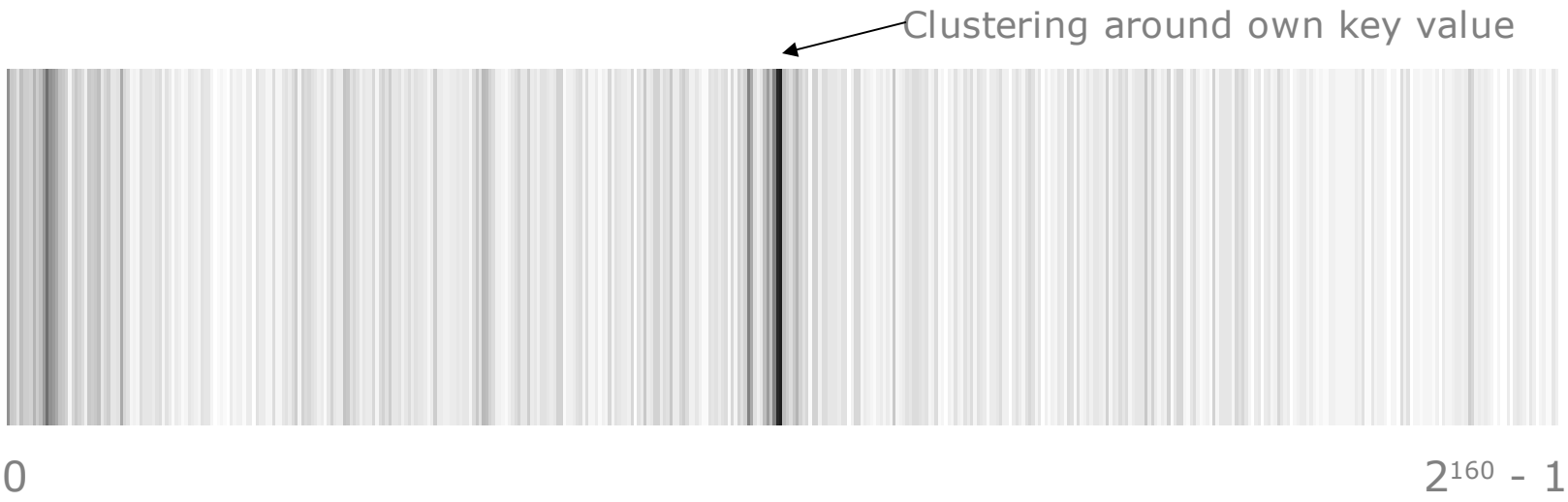


Figure 1.Typical request sequence.The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

Freenet

- Keys and Data distribution
  - 160-bit keyspace
  - Data clustered according to key values
  - Nodes attract requests for data with keys similar to theirs

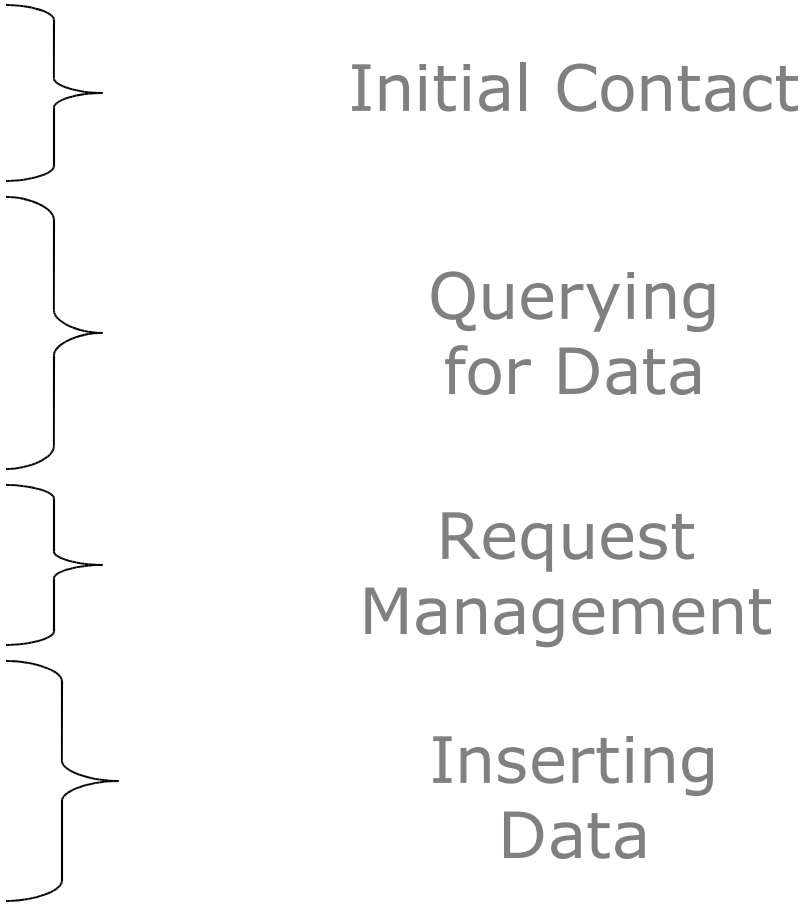


Freenet

- Data Store
  - Each node has an inventory of locally stored data, their hash keys and their most recent access/modification times
  - Each node has limited storage capacity
    - Potential overflow of data handled by removing least-recently used (LRU) files
    - NO file lifetime guarantees
  - Data passing through a node is stored locally, creating a dynamic cache

Freenet

- Protocol
  - Request.Handshake
  - Reply.Handshake
  - Request.Data
  - Send.Data
  - Reply.NotFound
  - Reply.Restart
  - Request.Continue
  - Request.Insert
  - Reply.Insert
  - Send.Insert



Freenet

- Protocol (2)
  - All messages contain
    - Transaction ID – 64-bit randomly generated
    - Hops-to-live limit
  - Request messages also contain
    - Search key or
    - Proposed key

Freenet

- Performance
  - Network convergence
    - Evolution of path length stability
  - Scalability
    - Network adaptability to increasing number of nodes and increasing traffic
  - Fault-tolerance
    - System resistance to node / network failure
  - Small-world scenario
    - Preferential attachment in the network permits efficient short paths between arbitrary points

Freenet

- Network convergence

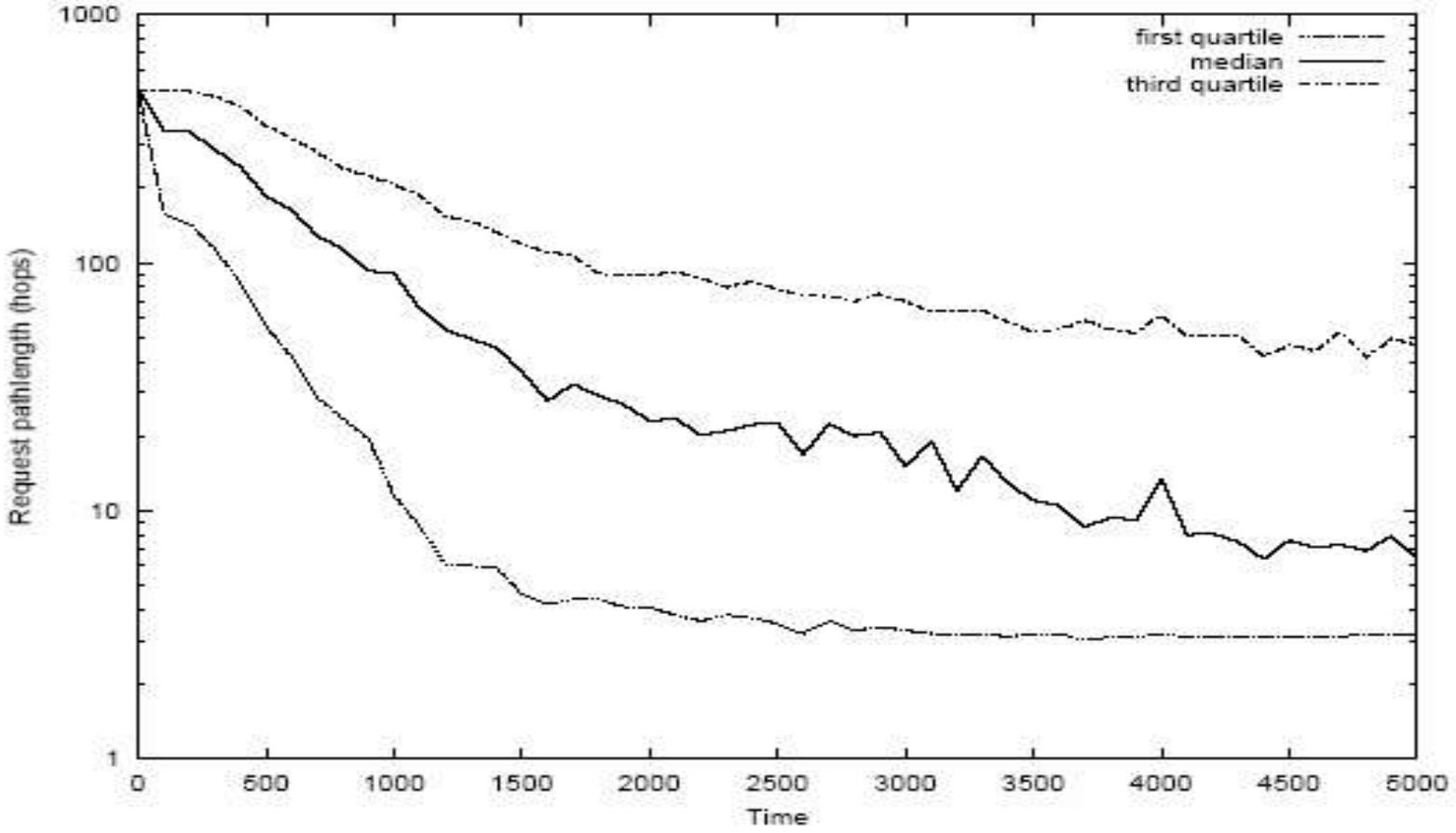


Fig. 2. Time evolution of the request pathlength.

Freenet

- Scalability

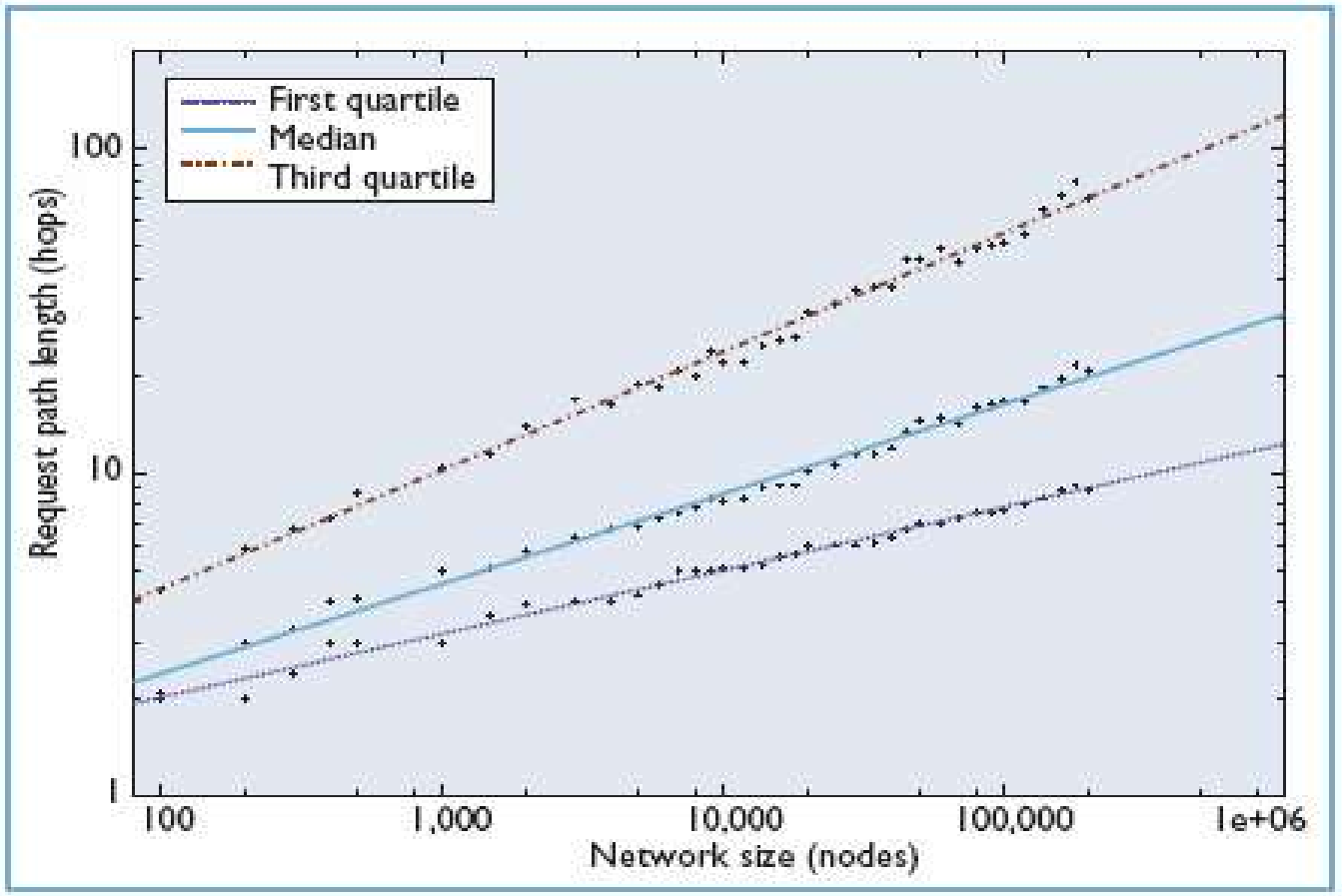


Figure 3. Request path length versus network size. The median path length in the network scales as $N^{0.28}$.

Freenet

- Fault-tolerance

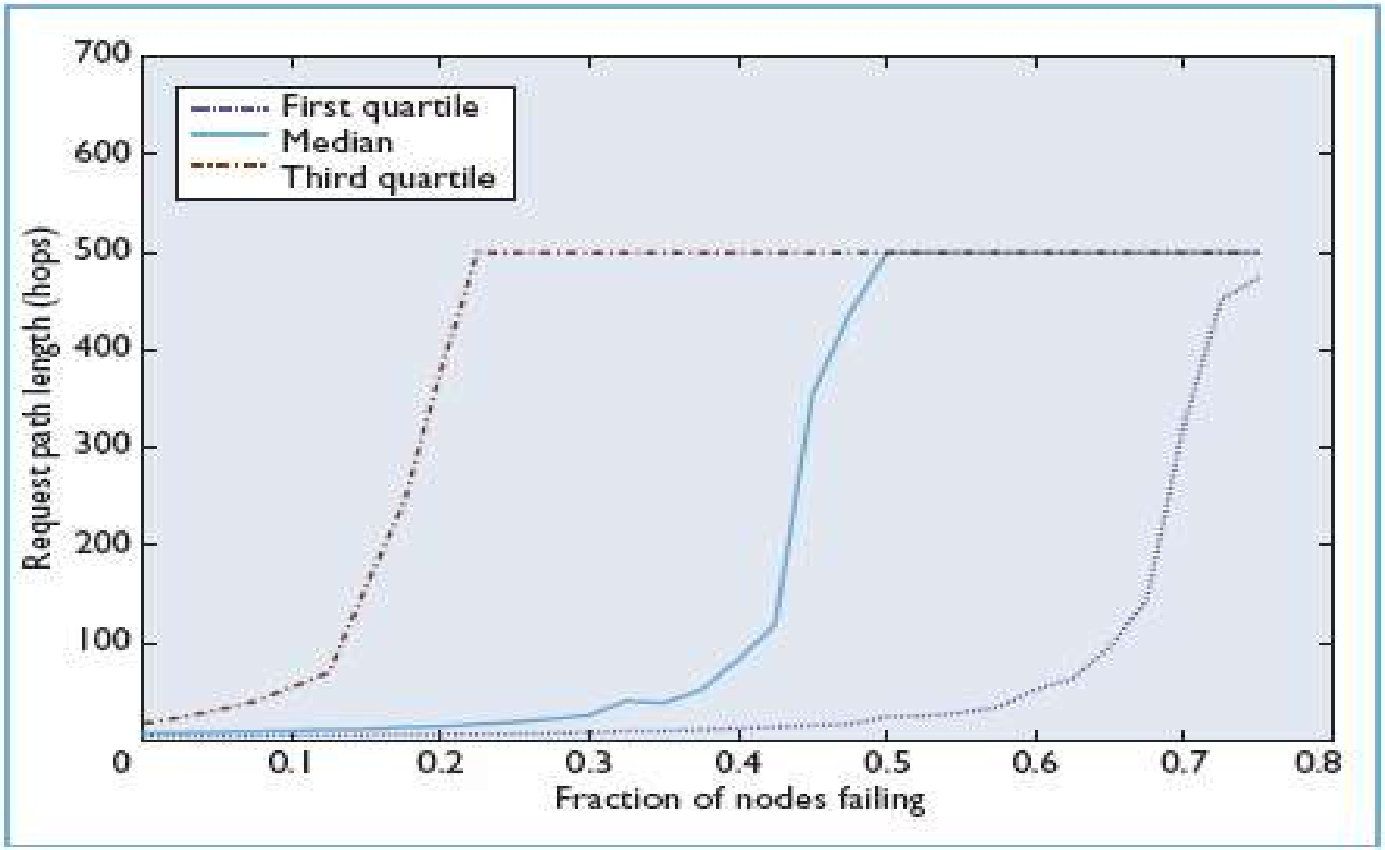


Figure 4. Request path length under random failure.

Freenet

- Fault-tolerance (2)

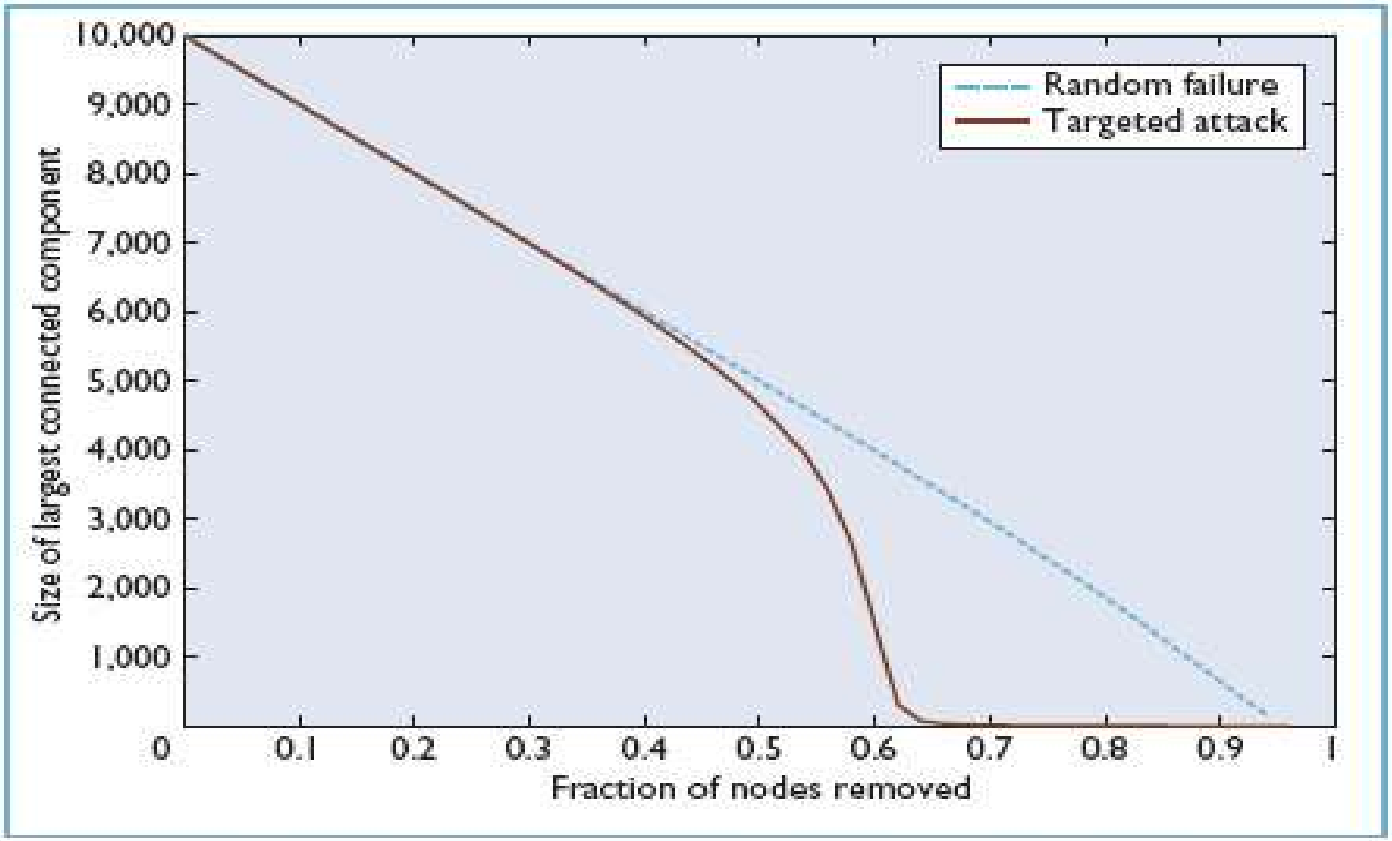


Figure 5. Connectivity under random failure and targeted attack. The network falls apart quickly when the well-connected nodes are targeted first.

Freenet

- Small-world scenario

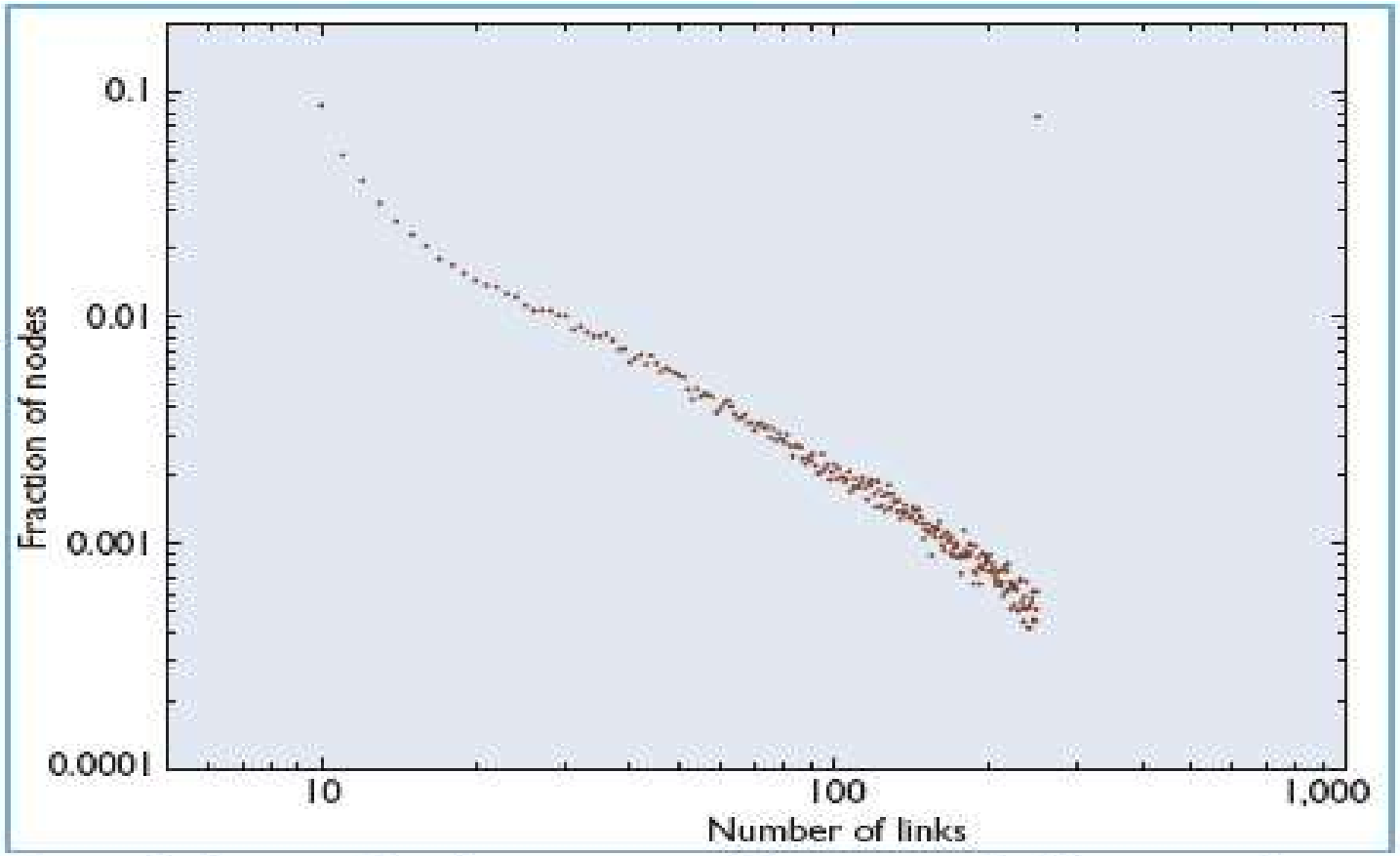


Figure Degree distribution among Freenet nodes. The network shows a close fit to a power-law distribution.

Freenet

- Security
  - Nodes are unable to determine origin of messages
  - Messages between nodes encrypted against local eavesdropping
  - Data source information periodically removed from data transfer
  - Hops-to-live trick
  - Hashing used to check data integrity and safeguard against intentional data corruption

Freenet

- Design weaknesses
  - No file lifetime guarantees
  - No efficient keyword search
  - Currently, no defense against DoS attacks
  - Bandwidth limitations not considered

Freenet

- Design strengths
  - Decentralized - no single point of failure
  - Scales well
  - Dynamic routing adapts well to changing network topology
  - High resilience to attacks

Freenet

- Next Generation Routing protocol
  - Nodes become smarter about deciding where to route information
    - Bandwidth considered when routing
    - Statistical information gathered about response times, successful requests and connection times
    - This information used to estimate nodes most likely to retrieve data quickest

Gnutella vs. Freenet

- Common features
  - Decentralization
  - Out-of-network initial connection
  - Peer-based query system

Gnutella vs. Freenet

- Differences
  - Flood-based routing vs. Dynamic decision-based routing
  - Out-of-band vs. In-band data transfer
  - No memory of past network traffic (stateless) vs. Routing tables
  - Read-only (File sharing) vs. Read/Write (File storage)
  - Static file locations vs. Dynamic file removal and replication
  - Openness vs. Anonymity
  - Low security vs. High security

End of Presentation

Nutella and Questions !!!!