Freenet: A Distributed Anonymous Information Storage and Retrieval System

Presentation by Theodore Mao
<tmao@uclink.berkeley.edu>
CS294-4: Peer-to-peer Systems
August 27, 2003

Topics

- Overview
- Architecture
  - GUID Keys
  - Routing
  - Network Evolution
  - Storage
  - Performance
- Planned Improvements
- Related Work
- Conclusion/Questions

Overview (1/2)

- What is Freenet?
  - Freenet is a P2P application designed to ensure true freedom of communication over the Internet. It allows anybody to publish and read information with complete anonymity.
- Who is behind Freenet?
  - Originally, Ian Clarke while a student at the University of Edinburgh, Scotland.
  - Still supervised by Ian Clarke, though many other people contribute to the project.
- How recent is Freenet?
  - Original paper appeared in 1999.
  - According to CiteSeer, it has been cited 195 times.

Overview (2/2)

- Purpose:
  - Prevent information censorship
  - Maintain personal privacy
- Goals:
  - Privacy for information producers, consumers, and holders
  - Resistance to information censorship
  - High availability and reliability through decentralization
  - Efficient, scalable, and adaptive storage and routing

Architecture

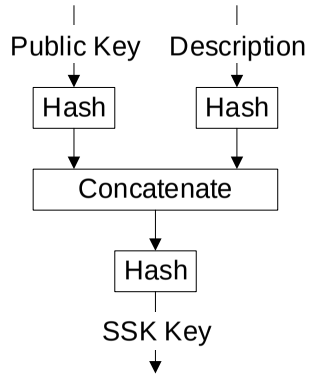
- Peer-to-peer network
- Participants share bandwidth and storage space
- Each file in network given a globally-unique identifier (GUID)
- Queries routed through steepest-ascent hill-climbing search

GUID Keys

- Calculated with an SHA-1 hash
- Two main types of keys
  - Content-hash keys
    - Used primarily for data storage
    - Generated by hashing the content
  - Signed-subspace keys (SSK)
    - Intended for higher-level human use
    - Generated with a public key and (usually) text description, signed with private key
    - Can be used as a sort of private namespace
    - Description e.g. politics/us/pentagon-papers

SSK Generation and Query Example

- Generate SSK:
  - Need: public/private keys, chosen text description
  - Sign file with private key
- Query for SSK:
  - Need: public key, text description
  - Verify file signature with public key



Routing (1/2)

- Every node maintains a routing table that lists the addresses of other nodes and the GUID keys it thinks they hold.
- Steepest-ascent hill-climbing search
- TTL ensures that queries are not propagated infinitely
- Nodes will occasionally alter queries to hide originator

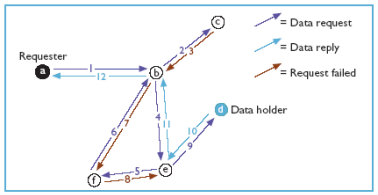


Figure 1.Typical request sequence.The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

Routing (2/2)

- Requesting Files:
  - Nodes forward requests to the neighbor node with the closest key to the one requested
  - Copies of the requested file may be cached along the request path for scalability and robustness
- Inserting Files:
  - If the same GUID already exists, reject insert – also propagate previous file along request path
  - Previous-file propagation prevents attempts to supplant file already in network.

Network Evolution

- Adding nodes:
  - Announce public key and physical address (e.g. IP) to an existing node
  - Announcement is recursively forwarded to random nodes
  - Nodes in the chain then collectively assign the new node a random GUID
- Route training:
  - As more requests are processed, nodes should specialize in handling a few parts of the key space

Storage

- LRU file elimination when out of disk space
- Possibly encrypted data (by content publisher), so that data holders can claim to be ignorant of the content they store (plausible deniability)

Performance

- Some real-world and simulated data available, but generally hard to test
  - Hard to tell the size of the network
  - Nodes are all anonymous

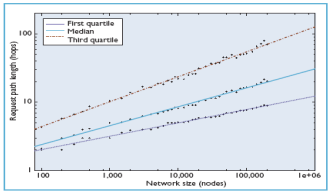


Figure 3. Request path length versus network size. The median path length in the network scales as $N^{0.28}$.

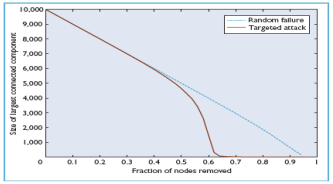


Figure 5. Connectivity under random failure and targeted attack. The network falls apart quickly when the well-connected nodes are targeted first.

Planned Improvements

- Next-Generation Routing (NGR)
  - Make Freenet nodes much smarter about deciding where to route information
  - Collect statistical information for each node in its routing table, e.g. response times, successful responses, etc.
  - Use this information to improve routing decisions

Related Work

- File-sharing: Gnutella, FastTrack, Overnet
- Consumer Anonymity: Anonymizer, SafeWeb/Triangle Boy
- Producer Anonymity: Rewebber, TAZ, Publius
- Shared-storage: OceanStore, Cooperative File System, PAST

Conclusion (1/3)

- Primary Points
  - Prevention of censorship and protection of privacy is an important and active field of research.
  - Freenet is a (successful?) implementation of a system that resists information censorship
  - Freenet is an ongoing project that still has plenty of flaws
  - There may be a tradeoff between network efficiency and anonymity, robustness.

Conclusion (2/3)

- What's wrong with Freenet?
  - Not well tested in the wild – scalability, resilience. Insertion flooding is one way to take out the network.
  - Anonymity guarantees not that strong – "Most non-trivial attacks would probably be successful in identifying someone making requests on Freenet."
  - No search mechanism – a standard search would allow attacks to take out specific content holders
  - Suffers from problems of establishing initial network connection.

Conclusion (3/3)

- More information at http://freenetproject.org/
- Questions? Comments?