COMP7370 Advanced Computer and Network Security

Freenet: A Distributed Anonymous Information Storage and Retrieval System (3)

Topics:

1. Retrieve Data
2. Store files
3. Manage files
4. Security issues

- Retrieve data
  - Need binary file keys to retrieve files
  - Send binary file keys as requests
  - Each node check its local store
    - Return if found, return the file
    - Otherwise, What?
      a) Look keys in its routing table;
      b) Forward the request to other nodes
      c) Found: pass data back to upstream requestor
      d) caching: update routing table (file key and source)

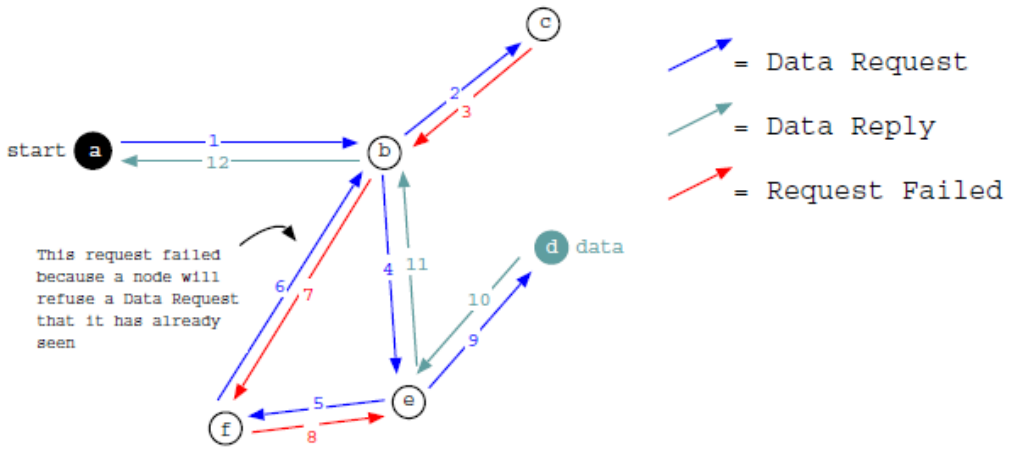

Fig. 1. A typical request sequence.

- Store file - input: f and hops_to_live
  - Compute binary file key, i.e., bfk(f)
  - Check existing keys in own store
  - If pre-existing key found, re-compute binary key
  - Look up the routing table, find the nearest key bfk' to bfk
  - Send f and its bfk to node(bfk')
  - Do above until hops_to_live limit is reached

- Manage data
  - Each node: configure datastore size
  - LRU cache
  - Routing table: LRU.
  - No guarantees of file lifetimes: outdated files can fade away

- Performance
  - Why simulations?
  - How to design experiments?
    - Performance
    - Scalability
    - Fault tolerance

- Security
  - Anonymity of requestors and inserters of files
  - Protection against malicious modifications. How?
  - A malicious node can not tell who are senders. Why?
  - Can we hide keys? No. need keys in routing tables
    - Use pre-routing of messages.