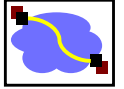




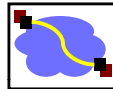
15-441 Computer Networking P2P

Why P2P?



- Scaling: system scales with number of clients, by definition
- Eliminate centralization:
 - Eliminate single point of failure
 - Eliminate single point of control
- Use “spare” capacity
- Self-managing

P2P example: file-sharing



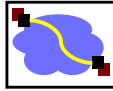
- Lots of different P2P uses, but file sharing is common and useful example
- Issues:
 - **Search:** Find a file
 - **Fetch:** Downloading the file
 - **Publish:** inserting a new file
 - **Join:** Maintaining network (e.g., dealing with faults, new members, etc.)

Approaches to P2P



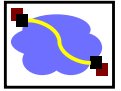
- Centralized
- Flooding
- Supernodes
- Routing
 - Structured
 - Un-structured

Centralized



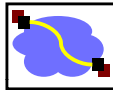
- (How is this P2P?)
- Search: ask central server
- Fetch: download from a peer
- Publish: report to central server
- Join: contact central server

Flooding



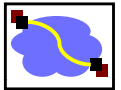
- Search: ask everyone you know, then they ask, ...
 - How to limit search message traffic?
- Publish: implicit
- Join: let a peer know you exist
- Supernodes help with this, but in the end still has same properties

Structured routing: e.g, Chord



- We saw how this might work with Chord
- Search: use hash to find node.
Makes none-exact searches hard
- Fetch: from single peer or ?
- Publish: implicit in joining
- Join: needs one peer in system

Unstructured routing: E.g., Freenet



- Freenet is an example of an unstructured overlay using key-based routing.
- Goals
 - **Censorship-resistance**
 - **Anonymity**: for producers and consumers of data
 - Nodes don't even know what they are storing
 - **Survivability**: no central servers, etc.
 - **Scalability**

Big Idea: Keys as First-Class Objects

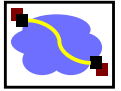


Keys name both the objects being looked up and the content itself

- **Content Hash Key**
 - SHA-1 hash of the file/filename that is being stored
 - Produces 160-bit digest
 - Hash property: should be difficult to find a different file/filename that hashes to same value
- **File key**
 - Key is based on human-readable description of the file (SHA-1 hash)
 - used to search for content
 - Problem: same name might be selected for different files
- **Encryption of files**
 - Documents can be encrypted: suggested key is unhashed human-readable description of file

9

Publishing and Querying in Freenet



- Process for both operations is the same!
- Keys passed through a chain of proxy requests
 - Nodes make local decisions about routing queries
 - Queries have **hops-to-live** and a **unique 64-bit ID**
- Two cases
 - **Node has local copy of file**
 - File returned along reverse path
 - Nodes along reverse path cache file
 - **Node does not have local copy**
 - Forward to neighbor "thought" to know of a key close to the requested key
 - First look up nearest key in routing table and forward request to that neighbor, if that fails, try second nearest key, etc.

10

Routing Queries in Freenet

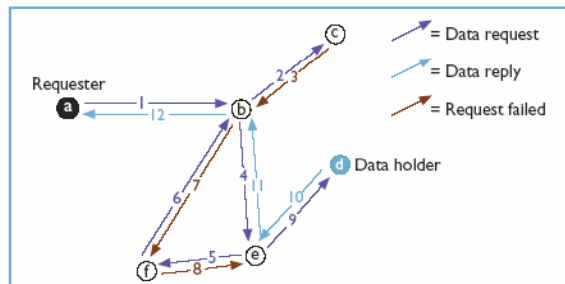
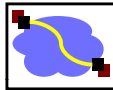


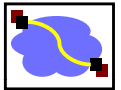
Figure 1. Typical request sequence. The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

After success, node **a** creates a link in its routing table for the key to node **d**.

Note: alternatively, any node on path from **d** to **a**,

11

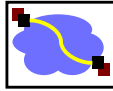
Freenet Design



- Strengths
 - Decentralized
 - Anonymous
 - Scalable
- Weaknesses
 - **Problem:** how to find the names of keys in the first place?
 - No guarantee search will succeed
 - No file lifetime guarantees
 - No efficient keyword search
 - No defense against DoS attacks

12

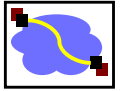
Freenet Security Mechanisms



- Encryption of messages
 - Prevents eavesdropping
- Hops-to-live
 - prevents determining originator of query
 - prevents determining where the file is stored
- Hashing
 - checks data integrity
 - prevents intentional data corruption

13

P2P: summary



- Many different types: centralized, flooding, routing
- Issues:
 - Failure mode: single point of failure? Search success?
 - Flooding is onerous
 - Network Topology different than overlay topology
 - Nodes aren't all the same
 - Search can be hard

15-441 S'10

Lecture 22: P2P

14