CS-438 Decentralized Systems Engineering

Fall 2022

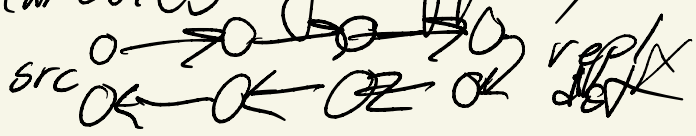
Week 7

Structured vs Unstructured search

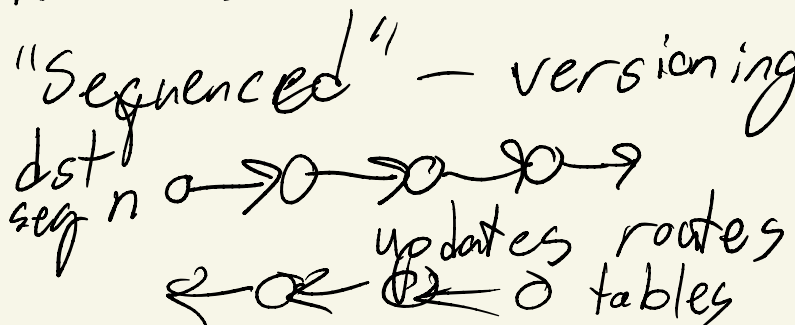
- Unstructured: "stateless" (mostly)
  Gnutella, Friendster, ... - flooding search
  + automatically adapts to network dynamics
  + general - arbitrary query predicate - $O(n)$
- Semi-structured: Bubblestorm - $O\sqrt{n}$
  + "mostly" stateless + general - "only" $O(\sqrt{n})$ efficient
- Structured search: more state
  - harder to handle dynamics / churn
  - less general
  + more efficient: $O\sqrt{\log n}$

Comparison: ad-hoc routing protocols

- AODV - Ad-hoc On-demand Distance Vector
  - Flooding search for a node
  - Searches gossipped, → return paths



- DSDV - Destination Sequenced Distance Vector
  - Build/maintain routing tables O(n)
  - "Sequenced" - versioning



Efficient structured search — distributed hash tables (DHTs)

Hash tables — depend on:
- random-access memory (RAM)
- "good" hash function
- not too full

Distributed:
- Hash function: avoiding collisions — more important
  - less time-sensitive
  - cryptographic hashes — common
  - well distributed / balanced
- Missing: RAM

Chord DHT

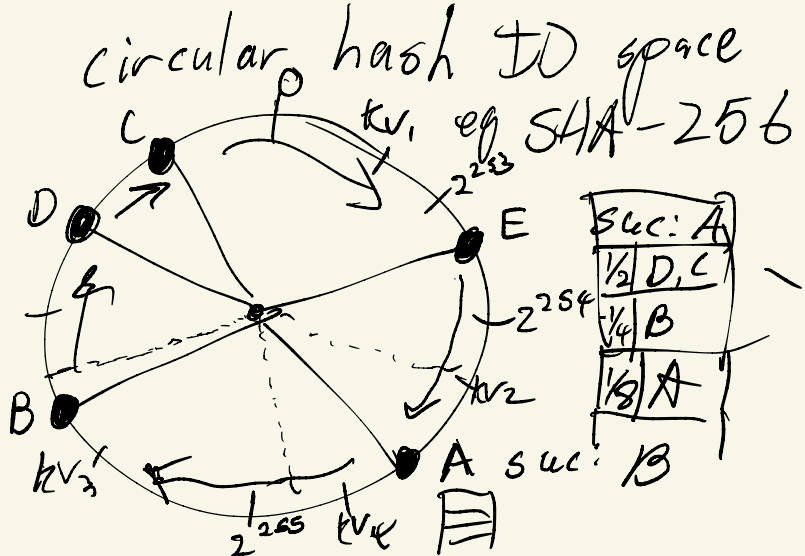
Goal:
$O(\log n)$
storage per node
comm/comp per op

- Hash into collection of RAMs
- Each node has hash-ID
  - input? name, IP, time, nard #, pub key
- Objects: key/value pairs
- API:
  - put(K, V)
  - get(K) → V/error

K → same hash

$2^{255}+2^{254}$

n nodes total
m k/v pairs
load: $\sim \frac{m}{n}$ per node
w/ redundancy
load: $\sim \frac{rm}{n}$



Challenge: reliability

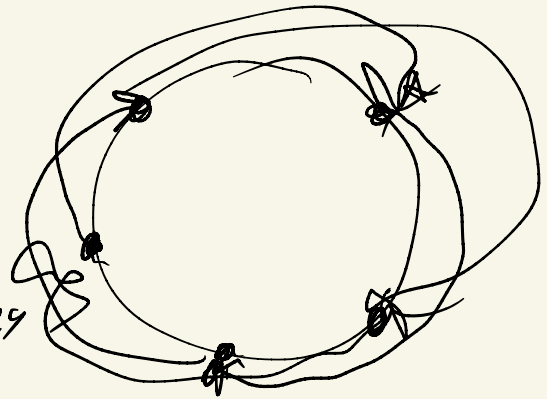
- solution: redundancy - factor r
  "owner" + (r-1) successors store copies

Challenge: maintaining structure

log n finger table buckets per node

Challenges:

- churn
- repairing bad structures



Challenge:
malicious
security