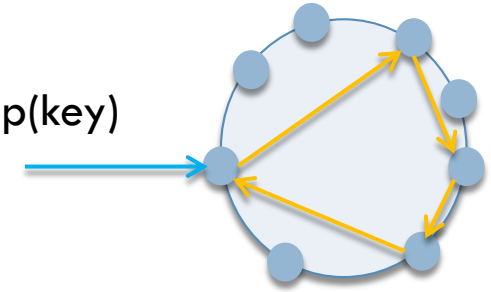
Big Data Research Lab
Computer Science Department
Colorado State University

D1HT: A Distributed One Hop Hash Table

Feb 27, 2015

What is Distributed Hash Table (DHT)?

- Storing and retrieved the data of form (key, values) pairs



- Find the responsible peer
- Store/retrieve data

- The key converted into numeric ID in the ring's range
- Each peer is responsible for IDs (ID>predID && and ID<=peerID)
- Lookup needs at most log(n) peers to traverse
- Peers allowed to join and leave
- Data replicated on k successor peers

Multiple-hops vs single-hop DHT

- In multiple hops DHT
  - Each peer has to keep a few routing information
  - The maintenance traffic is reduced
  - Lookup() will be forwarded to at most log(n) peers
- One hop DHT
  - Each peer has to keep a routing information of all peers in the system
  - The maintenance traffic is maximized
  - Lookup() operation requires only one peer to visit

The Purposed Solution

D1HT is one hop DHT with reasonable maintenance overhead

- Performs the majority of lookups with single hop
- Requires low bandwidth
- Provides balanced maintenance traffic
- The routing table of each peer
  - Includes the addresses of all peers in the system
  - Must be up to date to perform any lookup() in only one hop

Event Detection and Report Algorithm (EDRA) is purposed to overcome some challenges

- All peers in the system must be notified about any joining or leaving peer
  - In reasonable time
  - With reasonable bandwidth consumption
  - Without causing hot spots

How does EDRA work?

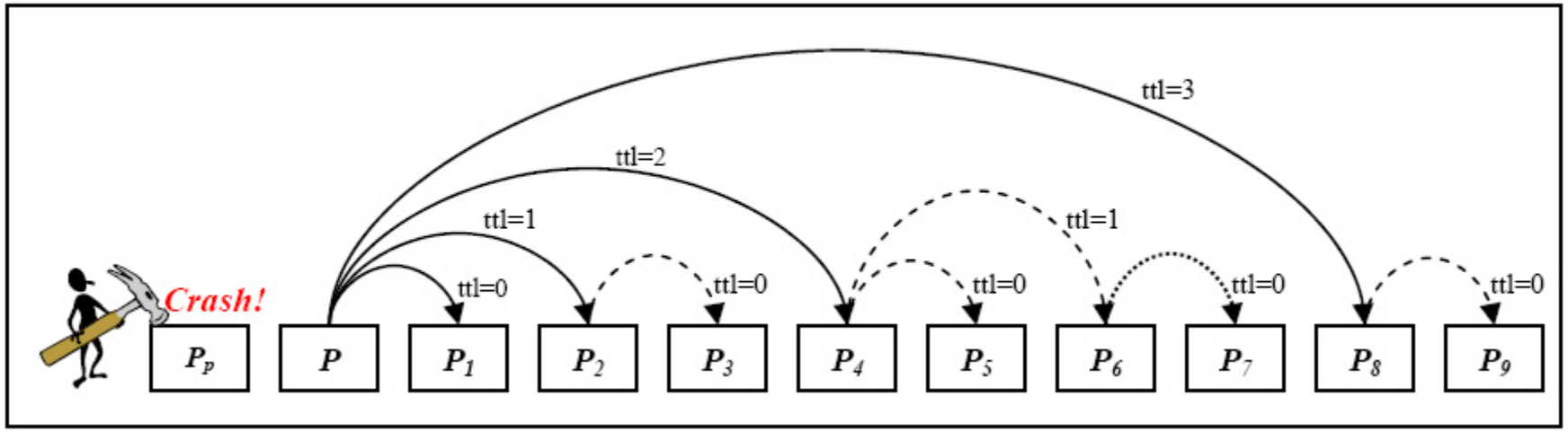
- Any peer will assume that its predecessor has left when it does not receive messages for a number of predefined seconds
- The detecting peer p will report an event to log(n) peers
  - The report message will send to the peers, succ(p, 2^l)
  - Where TTL=l included in each message: l = [0,log(n)-1)

Each peer will notified just once

- The peers pi that receive the notification message, msg(TTL)
  - Will not forward the message if TTL=0
  - Will forward the message to peers, succ(pi, 2^l)
    - Where l = 0,1,.. LLT-1
- To avoid redundancy the messages addressed to peers with id < pID will be ignored
- All peers will be notified in O(log(n))

Example

- n = 11, log(n)=4



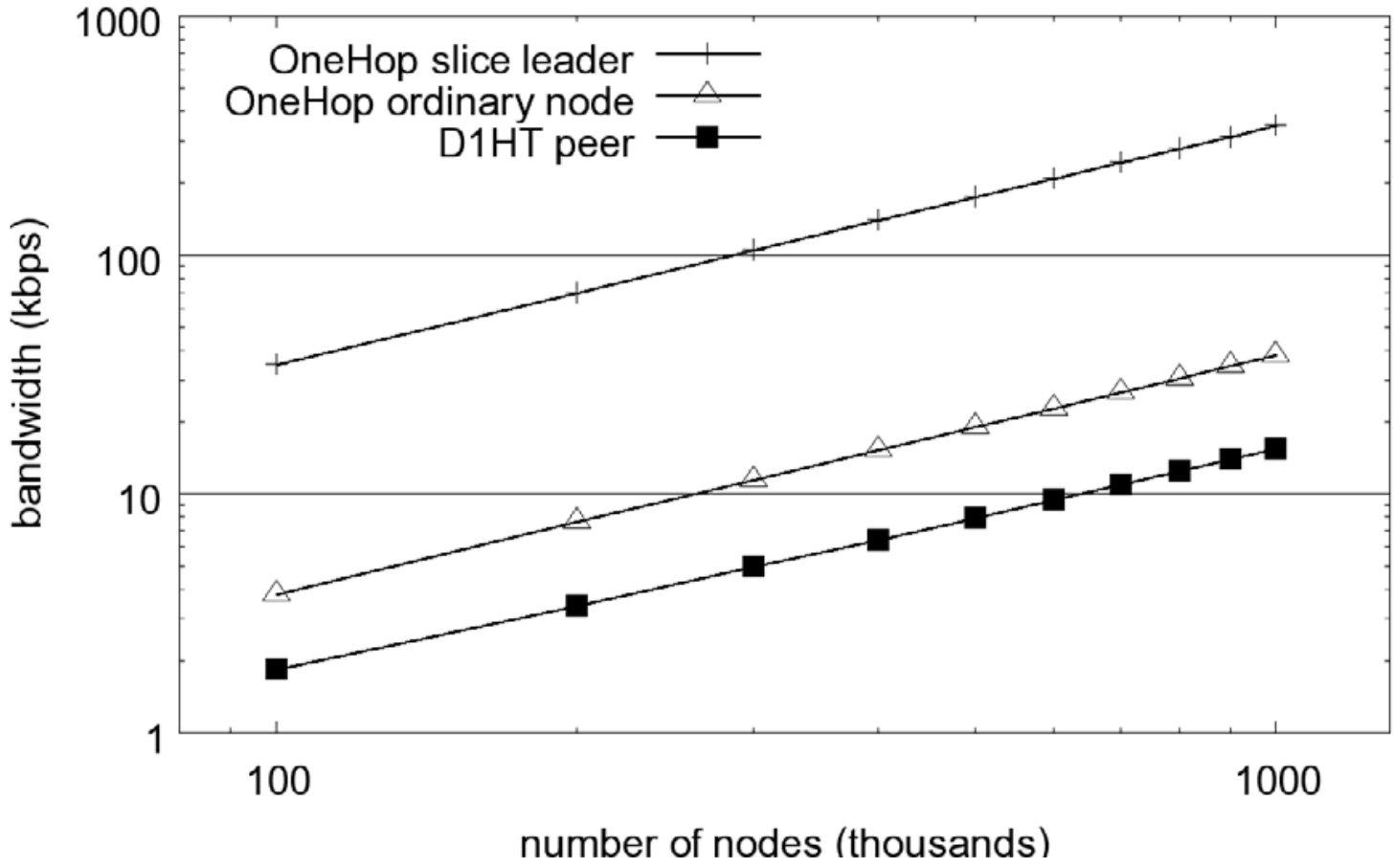
- All peers will be acknowledged in O(log(n))
- No peer will be acknowledged more than once

How load balancing is achieved?

- EDRA makes efficient use of the bandwidth
  - Good balancing in terms of incoming messages
- In terms of outgoing messages
  - For one event, the maximum load will be on the successor of the failed peer
  - For large number of events in a second, the load will depend on how the peers are distributed along the ring
    - Since the peers are uniformly distributed using hashing function the load will be also uniformly distributed

Comparing outgoing bandwidth requirements with similar approach

Outgoing bandwidth demands for oneHop and D1HT



Questions?