

Geographic Hash Tables (Ratnasamy, Karp, Yin, Yu, Estrin, Govindan, Shenker) [Ratnasamy02b]

CSci551: Computer Networks
SP2006 Thursday Section
John Heidemann

15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

1

Key ideas

- distributed hash service: key->location
- hashing based on physical location
 - store data at location closest to hash location
- replication
 - structured replication: for hot keys, map to multiple locations
 - perimeter refresh protocol: store data not just at correct location, but also *near* it
 - allows failures of individual nodes
 - also comes for free in the routing protocol
- locality
 - potentially we could get locality (get data near you)
 - but not really since it's hash

15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

5

Basic Idea

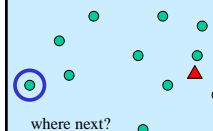
- hash data to a physical location
 - spreads load
 - avoids sending all data to user (otherwise user becomes a hotspot)
- use GPSR (Greedy Perimeter Stateless Routing) to get there
- shift in thinking for sensor nets from search to storage

15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

6

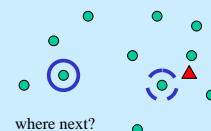
GPSR

- Greedy
 - most of the time
 - moves towards target
- Perimeter
 - when encounter holes
 - always required at end



15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

8



Geographic Hash Table

- store data at the *home* node
 - hash key into target location
 - home node is defined as the node closest to the target location
 - get there with GPSR
- place copies of data at *home perimeter*
 - why? **fault tolerance if home node fails**
 - how? **store data in last walk of perimeter**

15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

11

Perimeter and Structured Replication

- perimeter replication* for reliability
- structured replication*
 - why? for load balancing
 - how:
 - map data to *multiple* locations (on a quad tree)
 - store data at one location; go to all locations when searching
 - (somehow) decide when to shift from just using one location to 4, or 16, or...

15d_Ratnasamy02b: CSci551 SP2006 © John Heidemann

12

Comparing to Other p2p Systems

- search:
 - discovering names of event: not addresss (hard-coded)
 - name->location: geographic hash
 - finding location: GPSR
- update: same as search
- redundancy: perimeter replication (reliability), structure replication (performance)
- other features:
 - claim: works well for sensornets where nodes have geographic locations

Challenges in the Real World

- assumes *planar* communcation
 - real wireless world is not like that
- addressed in follow-on work

Other questions/observations?

- XXX