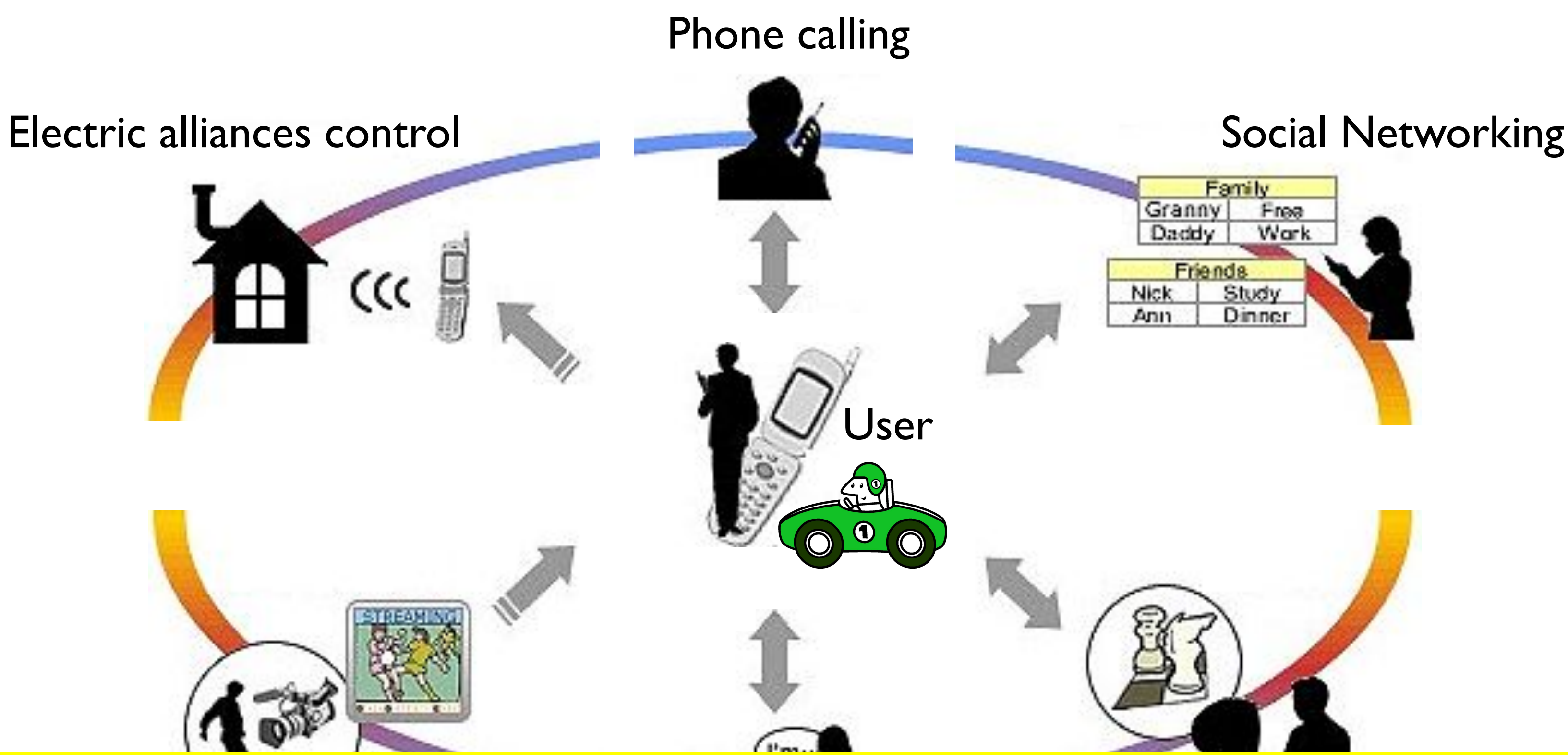
GeoServ: A Distributed Urban Sensing Platform

Jong Hoon Ahnn (UCLA, presenter)
Uichin Lee (KAIST)
Hyun Jin Moon (NEC Lab America)

Mobile Computing Trends in US

- Desktop, laptop, netbook:
  - Now, 1billion users
  - In 2015, 2 billion users
- Mobile phones
  - Now, 4.3 billion users
  - In 2013, 5.8 billion users
- Smartphones
  - Now, 150 million users
  - In 2020, billions of users???
  - Urban Sensing Enablers

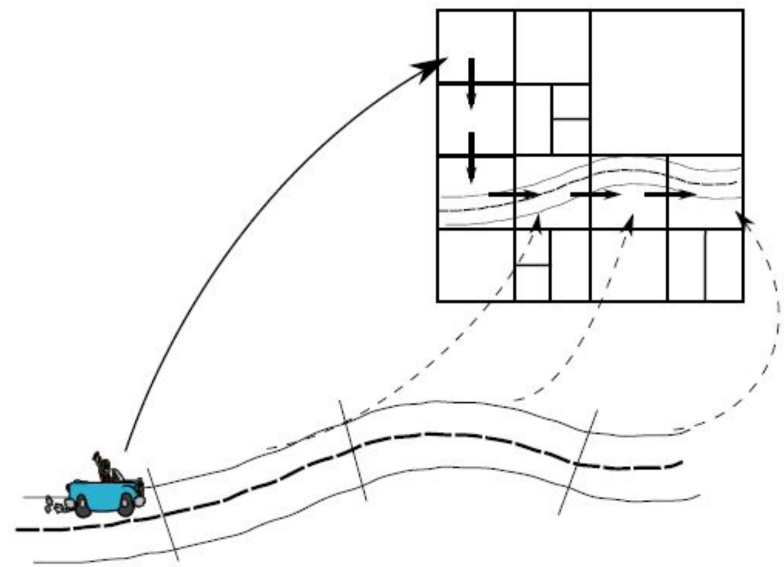
Urban Sensing with Smartphones



- 10 million mobile users generate sensor data updates and queries with the rate of 1KB/sec per user
  - Bandwidth required : >80Gbps, Storage space required: > 36TB/hr

Design Issues

- A mobile-to-mobile overlay network
  - PeerTIS: message routing via DHT [1]
  - Issues:
    - Network address translation (NAT)
    - Additional services such as session initiation protocol (SIP) or P2P proxy servers.
    - Intolerable delays
    - Significant resource consumption
      - e.g., battery, processing power, and bandwidth



- Therefore, consider using the Internet servers for large-scale urban sensing!

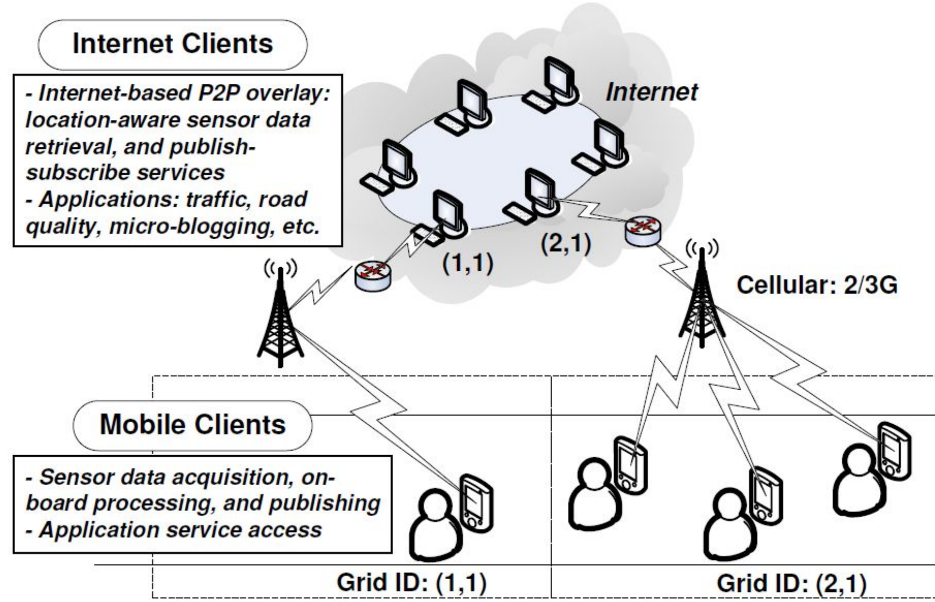
Design Issues (cont.)

- Existing Approach (in using Internet servers)
  - Centralized multi-tier architecture
    - Provide scalable services by provisioning more servers and bandwidth at the datacenters.
  - Semi-hierarchical architecture
    - Each organization maintains database servers for its own "stationary sensors."
    - Appropriate for participatory sensing by sharing users' computing resources.

- Still, need a scalable architecture to support billions of mobile users!
- Consider **a two-tier sensor networking architecture** for large-scale participatory sensing.

GeoServv: Two-Tier Architecture

- Two-tier Architecture
  - 1st tier: Internet-based fixed servers : form a distributed P2P sensor networking overlay
  - 2nd tier: mobile users can publish/access sensor data through Internet servers.



- The design emphasis for urban sensing applications is scalability and location-preservation for sensor data.

Two Services using GeoServ

1. *Location-aware* sensor data retrieval service
2. *Location-aware* publish-subscribe service

- Applications
  - Street-level traffic flow information service
  - Vehicular safety warning service
  - Ride quality monitoring service

System Design Requirements

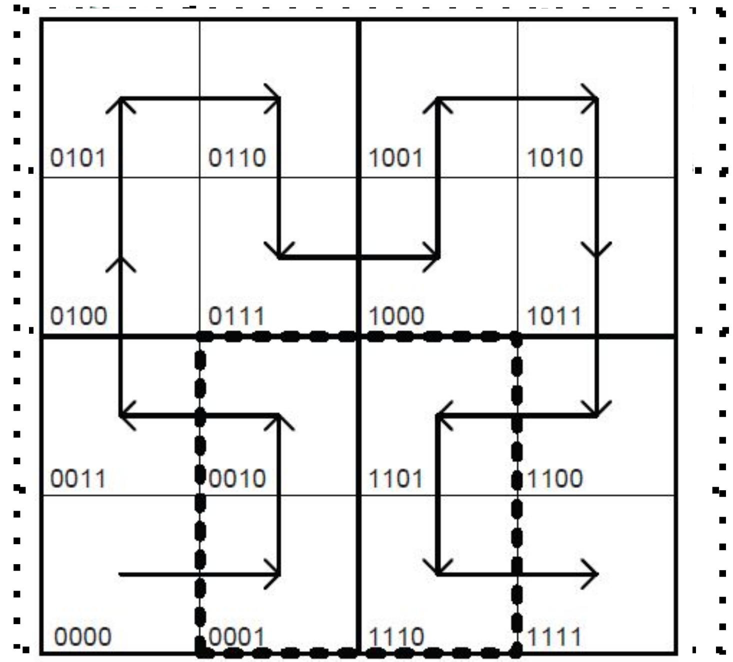
1. Seamlessly access to the sensor storage based on geographic locations.
2. Load balancing due to skewed distribution of vehicles on the road.
3. Publish-subscribe services
4. Dynamic mobile user association to the Internet server based on one's current location.

GeoServv Approach

- GeoTable
  - Key Based Routing (KBR): key space is geographic location.
  - Apps running on Internet servers and mobile clients communicate one another via GeoTable.
- GeoPS
  - An efficient publish-subscribe routing method based on the geographic routing via GeoTable.
- Other Components
  - GeoServDB
    - Manages sensor data of the grid space.
  - GeoServMobile
    - Provides control and transparent access of sensing resources.

LOCATION-AWARE SENSOR DATA RETRIEVAL SERVICE

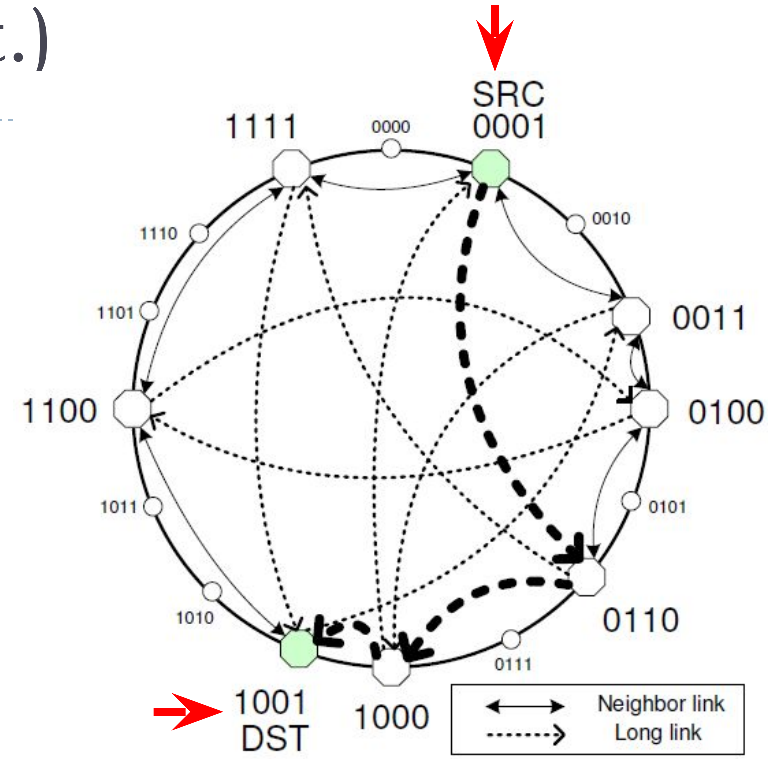
- In GeoServ, we divide the geographic area of interest into fixed size grids.
  - $2^M x 2^M$ grids
    - *M* is the smallest exponent that covers the entire area.
  - E.g., size of the contiguous U.S. is approximated as 3000km X 3000km
    - 2^{13} X 2^{13} fixed grids where R is given as 1km.

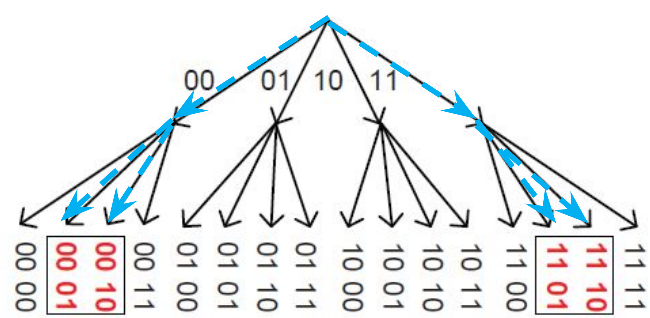


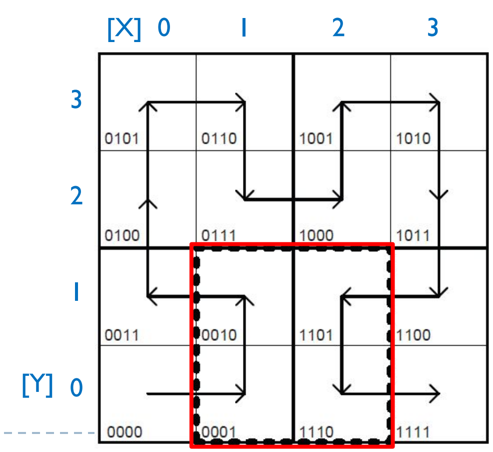
- How to construct overlay? Hilbert space filling curve!
  - A linear mapping function where successive points are nearest neighbors in the 2D grid.
  - The construction of the Hilbert curve is recursively defined.

LOCATION-AWARE SENSOR DATA RETRIEVAL SERVICE (cont.)

- Routing Semantics
  - Geocasting to a single grid point
  - Geocasting to multiple grid points
    - Simple rectangular area based addressing as {(x1; y1); (x2; y2)}.
  - Concurrent Geocasting (for GeoPS)
    - Due to overhead of sequential search, we utilize how the Hilbert curve is constructed.



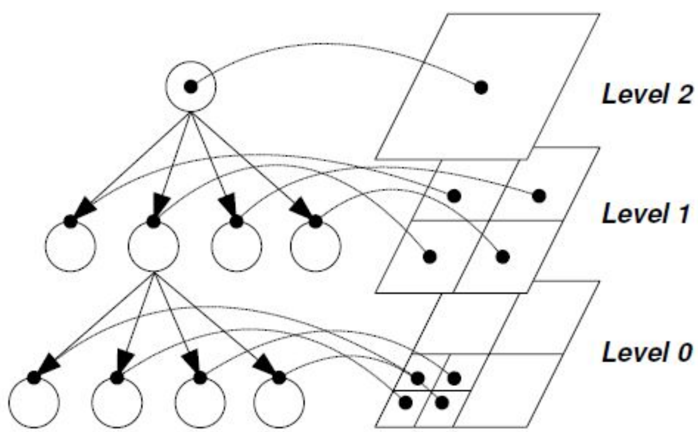




{(1;0);(2;1)}

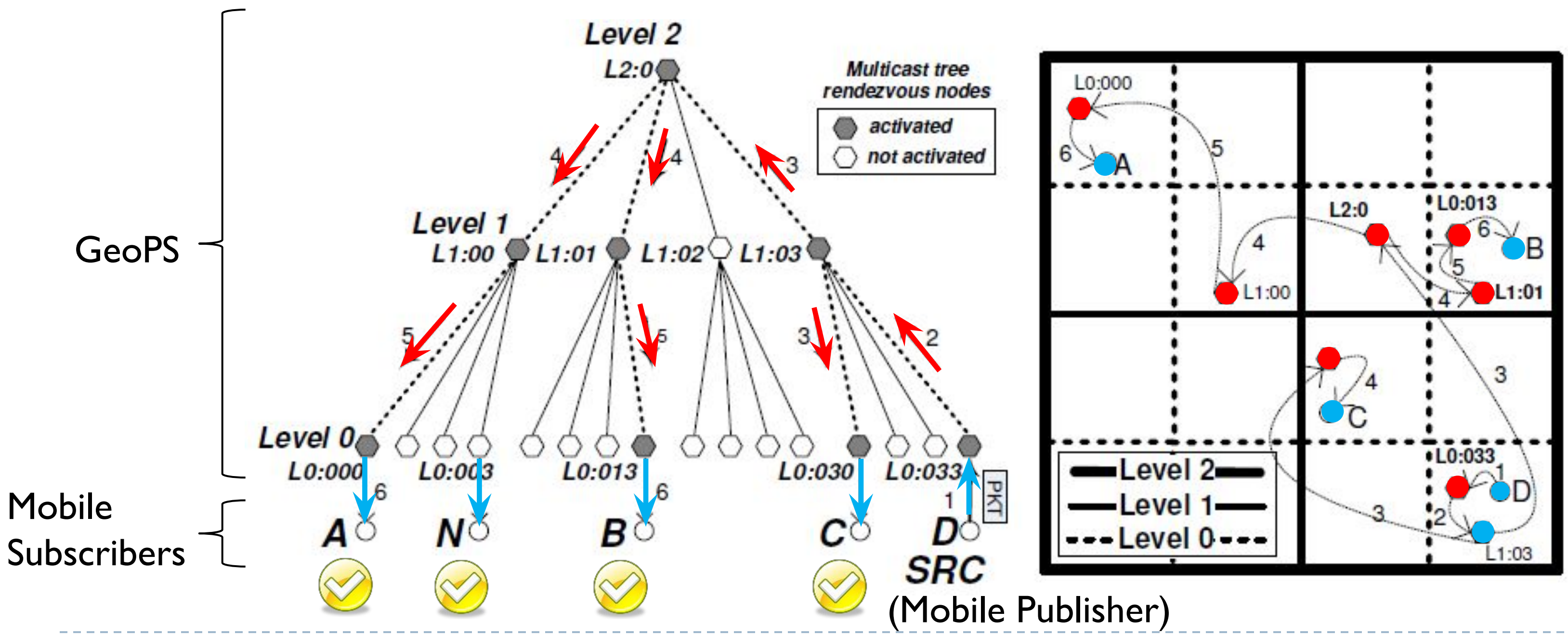
LOCATION-AWARE PUBLISH-SUBSCRIBE SERVICE

- GeoPS
  - A publish-subscribe service where the data updates on a region are published subscribers.
- A key design issue
  - To build a multicast tree a multicast tree that exploits the geographic locality of the group members.
- The idea of hierarchical geographic location
  - Inspired by hierarchical geographic location services (HGLS) in mobile ad hoc networks such as GLS [2] and HIGH-GRADE [3].



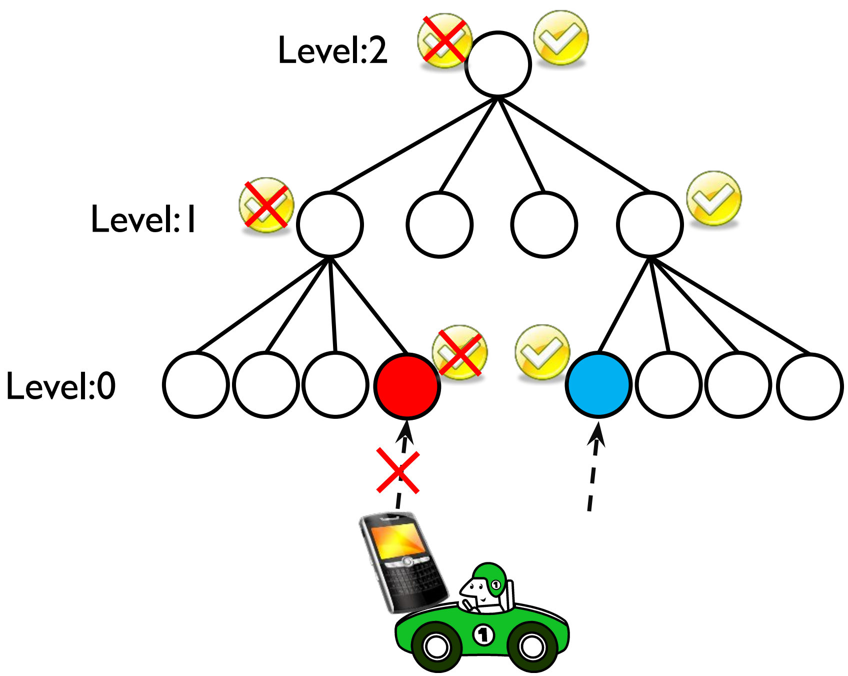
LOCATION-AWARE PUBLISH-SUBSCRIBE SERVICE (cont.)

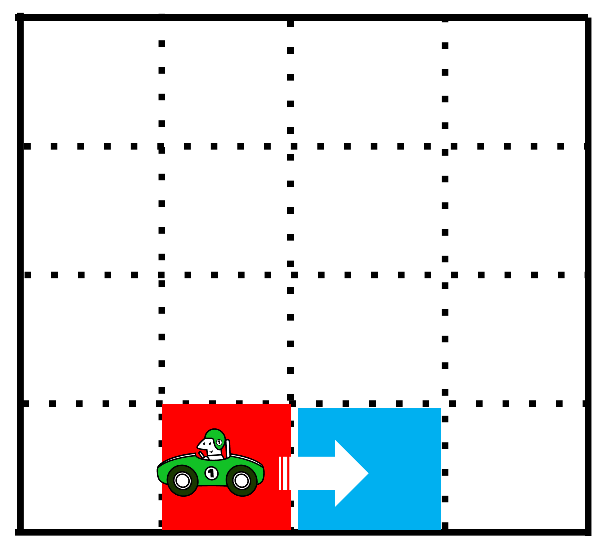
- A multicast tree construction in GeoPS
  - Build a multicast tree over this geographic hierarchy
  - Use our geocasting algorithm over the tree to preserve geographic locality.



LOCATION-AWARE PUBLISH-SUBSCRIBE SERVICE (cont.)

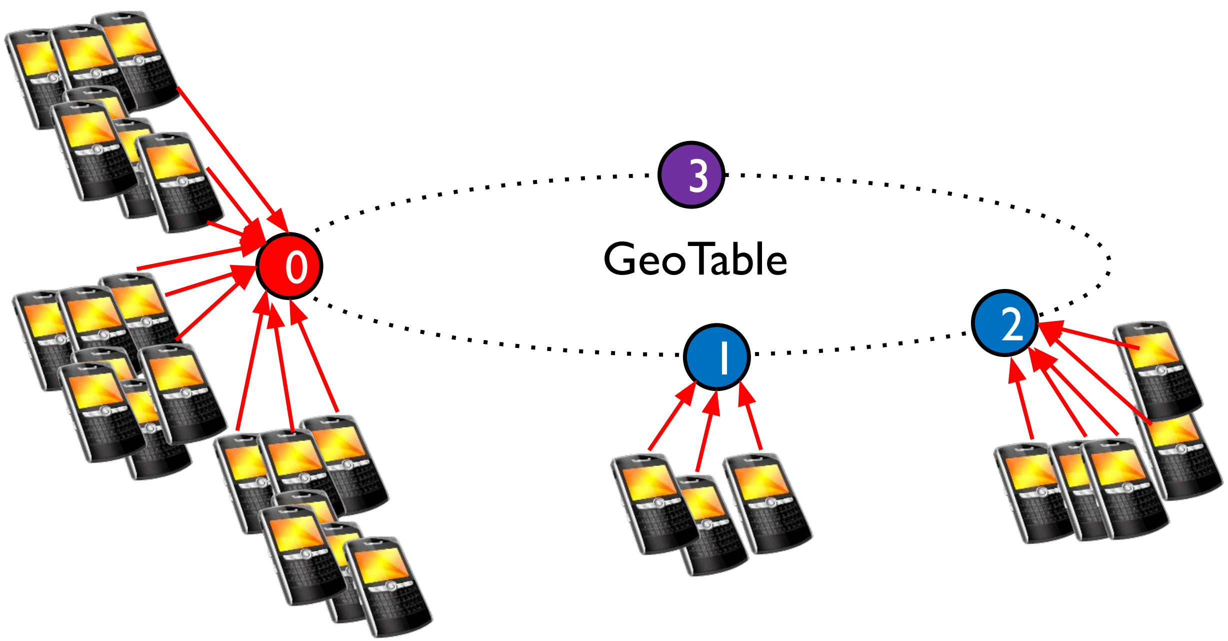
- Mobility handling in GeoPS
  - A mobile client's subscription needs to be updated (to upper layers) whenever the client crosses the level boundary (via explicit leave and join).
  - When mobile client C moves to the adjacent grid on the left (crossing level 1 boundary), rendezvous points at level 0, 1, 2 are updated.

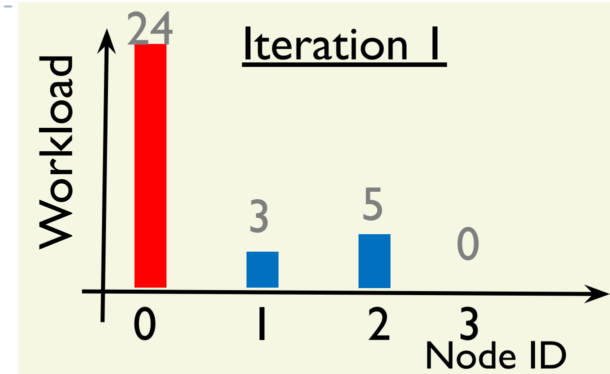


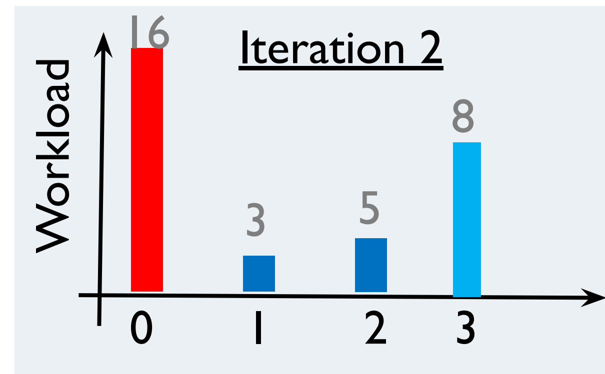


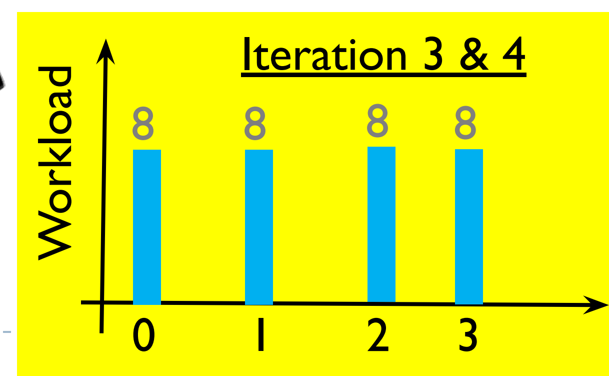
LOCATION-AWARE PUBLISH-SUBSCRIBE SERVICE (cont.)

- Load balancing in GeoPS
  - Dynamic load balancing
    - In highly populated regions with mobile clients,
    - GeoTable re-organizes overlay nodes such that those regions are served by more number of nodes.







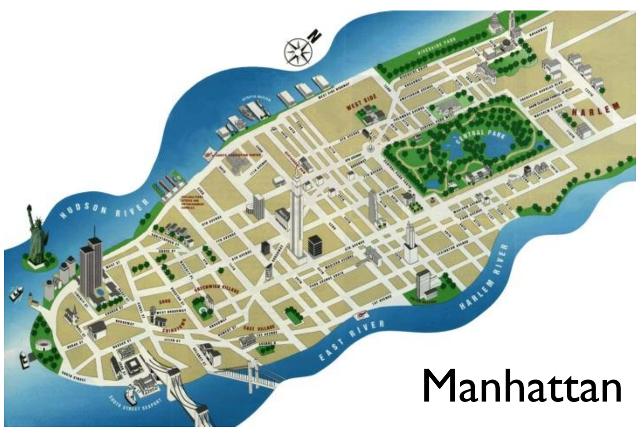


Evaluation List

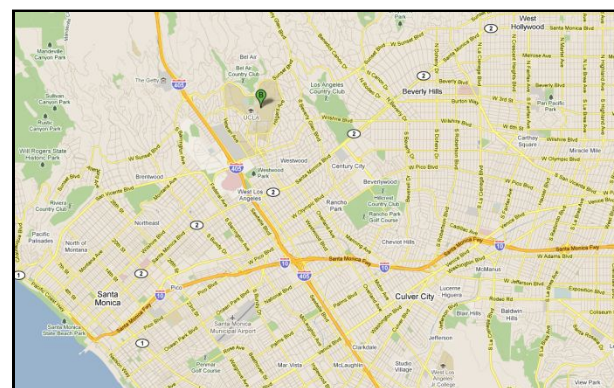
1. Location-aware data retrieval
2. Impact of query region sizes and concurrent geocasting
3. Load balancing
4. Location-aware publish-subscribe service

Evaluation Setup

1. An event-driven discrete-time simulator in C#
   - Supports dynamic node generation/join/leave, load balancing, and publish/subscribe features.
   - Long links are set to 5, as recommended in Symphony DHT [4]
2. A large-scale participatory vehicular sensing scenario
   - 100 different simulations
3. Realistic mobility generation: VanetMobiSim



Manhattan

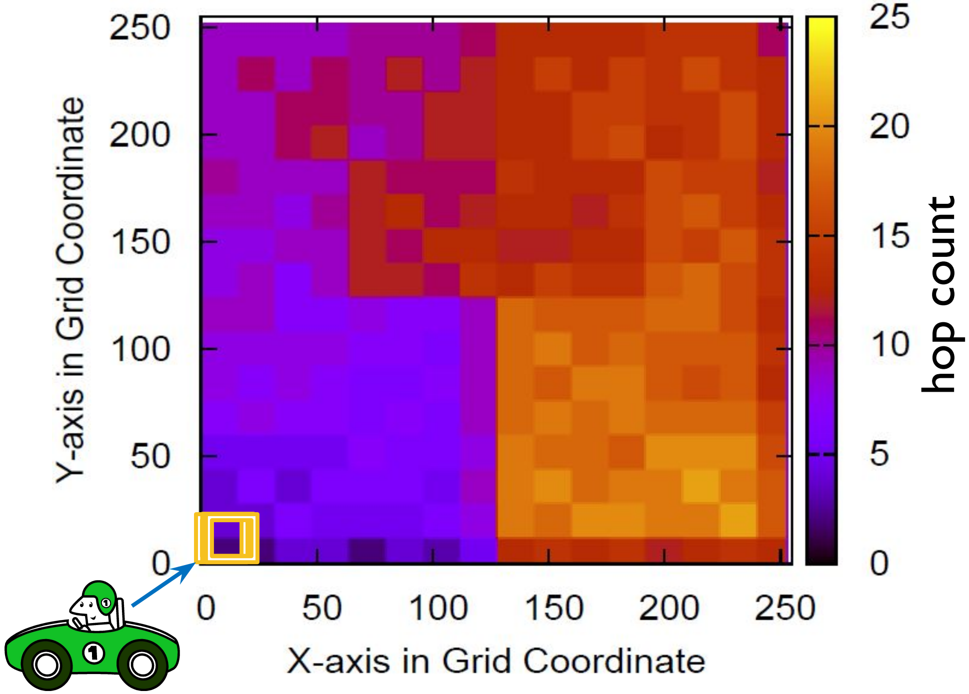


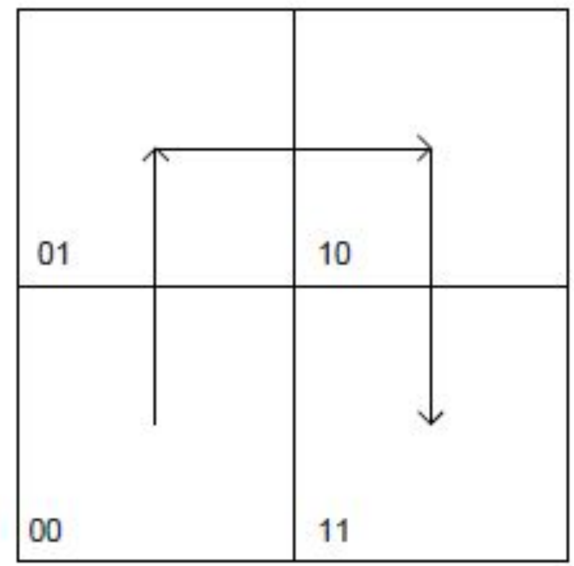
Westwood (Tigermap)

Simulation Setup: (1) Location-aware data retrieval

- Goal: To show that GeoTable preserves geographic locality
- Set-up:
  - Grid size: 256x256
  - Place a querying node at a grid point (0; 0)
  - Measure the hop count of a remote query with the square area of size 4x4,
  - Vary the location of the query from (0,0) to (252,252).
  - The number of overlay nodes is set to 1000.

Simulation Result: (1) Location-aware data retrieval



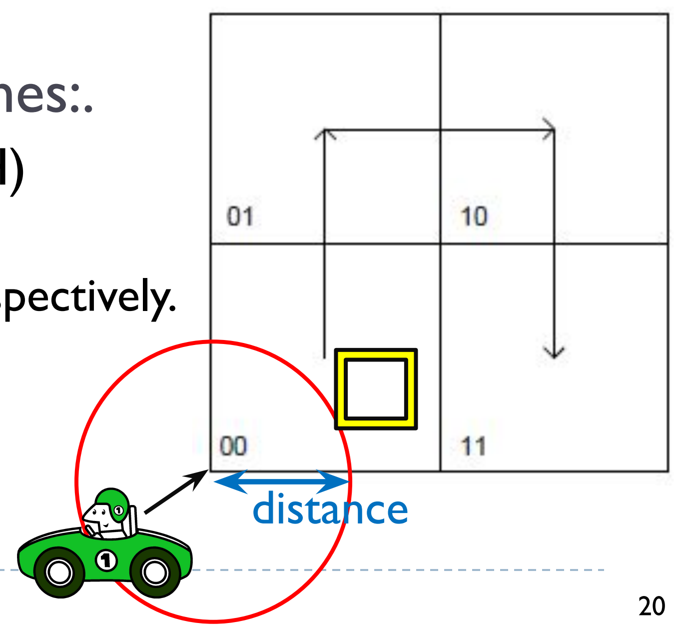


Hilbert curve construction

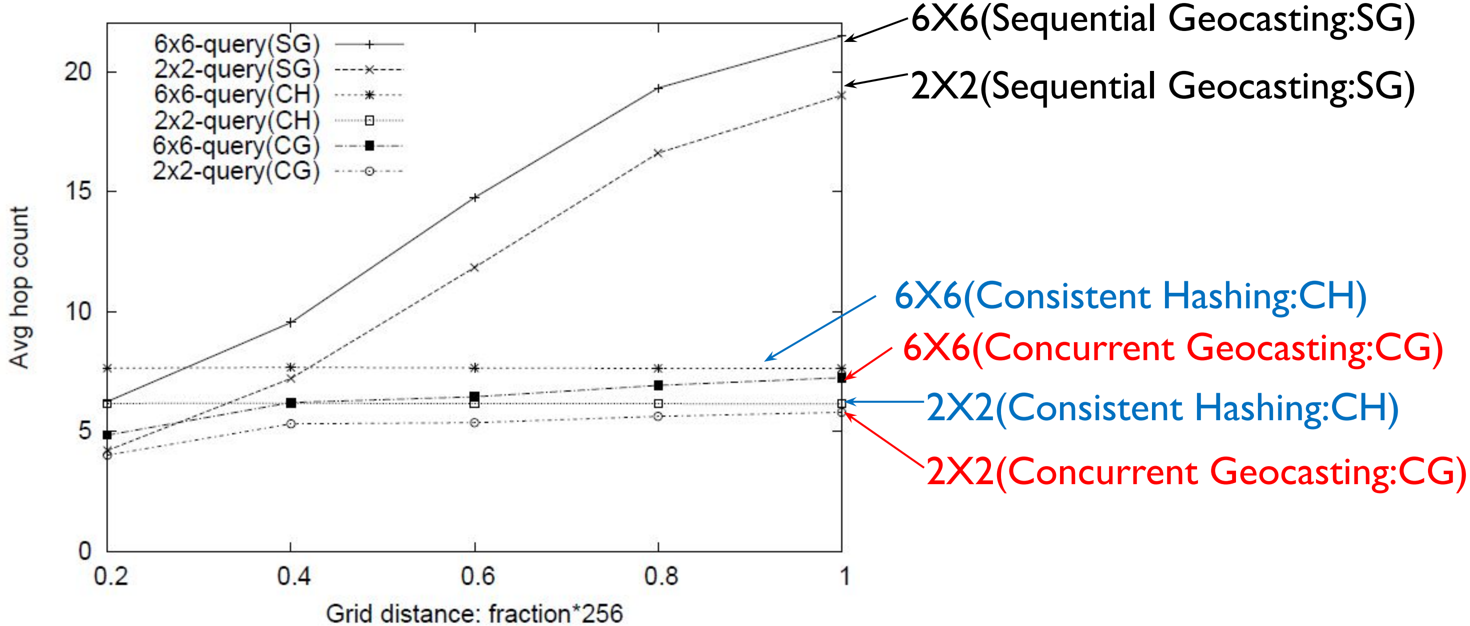
- As distance from (0; 0) increases, the hop counts increases (getting brighter).
- Non-uniform colors is that some degree of locality is lost after linearization, and long links are randomly assigned.
- Due to the recursive construction property of the Hilbert curve, the average hop count increases as we move clockwise.
- Locality is preserved at the higher level.

Simulation Setup: (2) Impact of query region sizes and concurrent geocasting

- Goal: to show the sensitivity of routing cost with different query region sizes: 2x2 and 6x6.
- Set-up:
  - The querying node is located at grid (0, 0).
  - And it sends queries with different area sizes.
  - Vary the distance between the querying node and the query region from 0 to 255.
  - Compare the performance of various schemes:.
    - Conventional DHT with consistent hashing (CH)
      - Each grid is randomly mapped into the key space;
      - 2x2 and 6x6 queries require 4 and 36 unicasts, respectively.
    - Sequential geocasting (SG)
    - Concurrent geocasting (CG)



Simulation Result: (2) Impact of query region sizes and concurrent geocasting



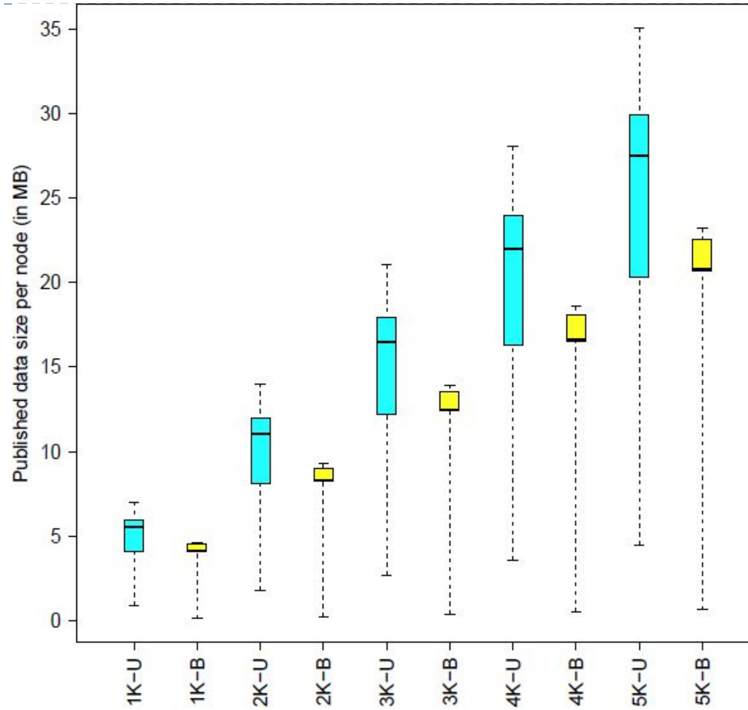
- As the grid distance increases, the average hop count of SG and CG increases (CG is better than SG).
  - But, no change on CH with the distance due to lack of locality.
- Note that delay improvement of CG against SG comes at the cost of more packet forwarding.
  - Still, far more efficient than issuing individual unicast to each segment.

Simulation Setup: (3) Load balancing

- Goal: to show how heterogeneous distribution of mobile clients influences the overall load imbalance
- Set-up
  - Mobile clients publish sensor data to the overlay nodes.
  - Road topology information from Tiger Los Angeles map.
    - The area size is 12,800m X 12,800m, centered at the UCLA campus.
    - The grid size of 50m X 50m, composed of 256x256 grids.
  - Mobile Clients
    - The different numbers of mobile clients from 1000 to 5000 with a gap of 1000 nodes.
    - Data generation with the rate of 128 Bytes/sec.
  - Overlay nodes
    - The number of overlay nodes is fixed to 1000 nodes.
  - Measurement: the total published data size per node to see how per-node workload is unbalanced.

Simulation Result: (3) Load Balancing

- boxplot shows min, 25%, median, 75%, and max;
- U: unbalanced, B: balanced
- 1K: 1000



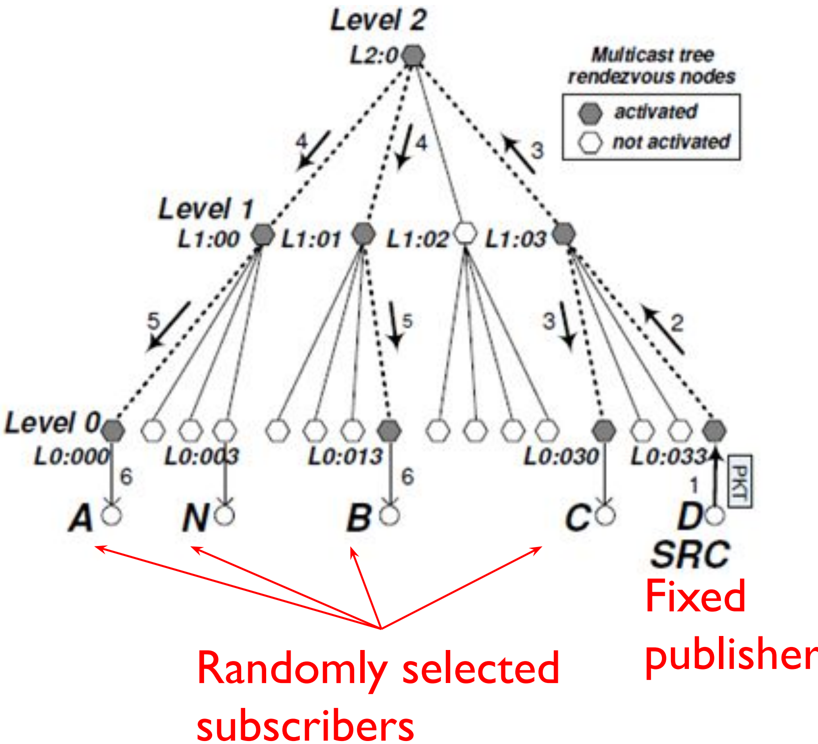
mobile clients

- As expected, the total data size increases linearly, as the number of mobile clients increases.
- The case without load balancing shows much higher variation as opposed to the case with load balancing.
- Still, minor variations in the balanced.
  - Because load balancing requires several iterations of leave/join-based load balancing operations.

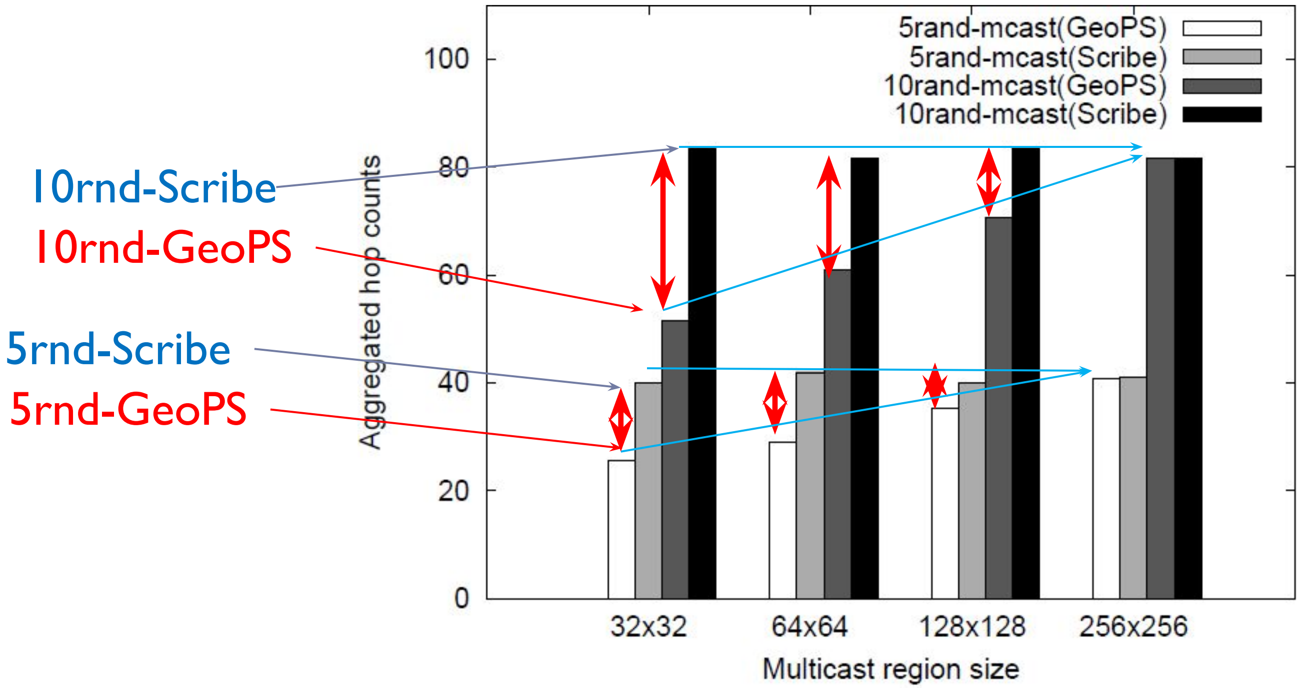
Simulation Setup: (4) Location-aware publish-subscribe service

- Goal: to show the geographic locality of our subscription-based multicast routing.

- Set-up:
  - The region size ranges from 32x32 to 256x256 grids.
  - Vary the origin of the region from (0, 0) to the maximum allowable.
    - E.g., for 32x32, it is (224, 224), and report the average hop count.
  - Randomly choose 5 or 10 random grids
  - Measure the aggregated number of hop counts.
  - Compare the performance of GeoPS with Scribe.
    - Note that Scribe destroys the locality by using consistent hashing.



Simulation Result: (4) Location-aware publish-subscribe service



- GeoPS exploits the locality of receivers.
- As the area size (where the subscribers lie) increases, geographic locality among subscribers disappears,
- Accordingly, the cost of GeoPS increases, converging to that of Scribe in the case of 256x256.

Concluding Remarks

- A scalable sensor networking system design
  - To enable location-relevant sensor data sharing among mobile users with smartphones.
- A two-tier sensor networking system: GeoServ
  - Exploits the Internet infrastructure.
  - Mobile users publish/access sensor information through the Internet-based distributed sensor storage.
- Mathematical Proofs in the paper
  - We proved that GeoServ protocols preserve geographic locality and validated their performance via extensive simulations.

References

[1] J. Rybicki, B. Scheuermann, M. Koegel, and M. Mauve. PeerTIS: a Peer-to-Peer Traffic Information System. In VANET, 2009.

[2] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In MobiCom, 2000.

[3] Y. Yu, G.-H. Lu, and Z.-L. Zhang. Enhancing Location Service Scalability with HIGH-GRADE. In MASS, 2004.

[4] G. S. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed Hashing in a Small World. In *USITS*, 2003.

[5] USA to Add 80 Million New Smartphone Users by 2011. http://twittown.com/mobile/mobile-blog/usa-add-80-million-new-smartphone-users-2011.

Thank you.