A Scalable Location Service for Geographic Ad Hoc Routing (2000)

Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, Robert Morris
MIT Laboratory for Computer Science

Presented by Nathan Farrington
Overview

Motivation for Grid:
scalable routing for large ad hoc networks
downtown metropolitan area, 1000s of nodes

Protocol Scalability:
The number of packets each node has to forward and the amount of state kept at each node grow slowly with the size of the network.
Current Routing Strategies

- **Traditional scalable Internet routing**
  - address aggregation hampers mobility

- **Pro-active topology distribution (e.g. DSDV)**
  - reacts slowly to mobility in large networks

- **On-demand flooded queries (e.g. DSR)**
  - too much protocol overhead in large networks
Flooding causes too much packet overhead in big networks.

Flooding-based on-demand routing works best in small nets. Can we route without global topology knowledge?
Geographic Forwarding Scales Well

- Assume each node knows its geographic location.

A addresses a packet to G’s latitude, longitude
- C only needs to know its immediate neighbors to forward packets towards G.
- Geographic forwarding needs a location service!
Possible Designs for a Location Service

- Flood to get a node’s location (LAR, DREAM).
  - excessive flooding messages

- Central static location server.
  - not fault tolerant
  - too much load on central server and nearby nodes
  - the server might be far away for nearby nodes or inaccessible due to network partition.

- Every node acts as server for a few others.
  - good for spreading load and tolerating failures.
Desirable Properties of a Distributed Location Service

- Spread load evenly over all nodes.
- Degrade gracefully as nodes fail.
- Queries for nearby nodes stay local.
- Per-node storage and communication costs grow slowly as the network size grows.
GLS’s spatial hierarchy

level-0

level-1

level-2

level-3

All nodes agree on the global origin of the grid hierarchy
3 Servers Per Node Per Level

- s is n’s successor in that square.
  (Successor is the node with “least ID greater than” n )
Queries Search for Destination’s Successors

Each query step:
visit \( n \)'s successor at each level.
GLS Update (level 0)

<table>
<thead>
<tr>
<th>2</th>
<th>11</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>23</th>
<th>6</th>
<th>29</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>26</th>
<th>17</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>21</th>
<th>4</th>
<th>8</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td></td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

**Invariant (for all levels):**
For node $n$ in a square, $n$’s successor in each sibling square “knows” about $n$.

**Base case:**
Each node in a level-0 square “knows” about all other nodes in the same square.
GLS Update (level 1)

Invariant (for all levels): For node $n$ in a square, $n$’s successor in each sibling square “knows” about $n$. 

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>11</th>
<th>9</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11</td>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>29</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>16</td>
<td>26</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>17</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>8</td>
<td>19</td>
<td></td>
</tr>
</tbody>
</table>
GLS Update (level 1)

Invariant (for all levels): For node $n$ in a square, $n$’s successor in each sibling square “knows” about $n$. 

location table content
GLS Update (level 2)

Invariant (for all levels): For node $n$ in a square, $n$’s successor in each sibling square “knows” about $n$. 

Location table content

Location update
Challenges for GLS in a Mobile Network

- Out-of-date location information in servers.
- Tradeoff between maintaining accurate location data and minimizing periodic location update messages.
  - Adapt location update rate to node speed
  - Update distant servers less frequently than nearby servers.
  - Leave forwarding pointers until updates catch up.
Performance Analysis

• How well does GLS cope with mobility?
• How scalable is GLS?
• How well does GLS handle node failures?
• How local are the queries for nearby nodes?
Simulation Environment

- Simulations using *ns* with CMU’s wireless extension (IEEE 802.11)
- Mobility Model:
  - random way-point with speed 0-10 m/s (22 mph)
- Area of square universe grows with the number of nodes in the network.
  - Achieve spatial reuse of the spectrum
- GLS level-0 square is 250m x 250m
- 300 seconds per simulation
GLS Finds Nodes in Big Mobile Networks

- Failed queries are not retransmitted in this simulation
- Queries fail because of out-of-date information for destination nodes or intermediate servers

Biggest network simulated: 600 nodes, 2900x2900m (4-level grid hierarchy)
GLS Protocol Overhead Grows Slowly

- Protocol packets include: GLS update, GLS query/reply
Average Location Table Size is Small

- Average location table size grows extremely slowly with the size of the network
Non-uniform Location Table Size

Simulated universe

Node 3 is to be the location server for all other nodes

The complete Grid hierarchy of level 3

Possible solution: dynamically adjust square boundaries
GLS is Fault Tolerant

• Measured query performance immediately after a number of nodes crash simultaneously. (200-node-networks)
Query Path Length is proportional to the distance between source and destination
Performance Comparison between Grid and DSR

**DSR (Dynamic Source Routing)**
- Source floods route request to find the destination.
- Query reply includes source route to destination.
- Source uses source route to send data packets.

**Simulation scenario:**
- 2Mbps radio bandwidth
- CBR sources, 4 128-byte packets/second for 20 seconds.
- 50% of nodes initiate over 300-second life of simulation.
• Geographic forwarding is less fragile than source routing.
• Why does DSR have trouble with > 300 nodes?
Protocol Packet Overhead

- DSR prone to congestion in big networks:
  - Sources must re-flood queries to fix broken source routes
  - These queries cause congestion
- Grid’s queries cause less network load.
  - Queries are unicast, not flooded.
  - Un-routable packets are discarded at source when query fails.
Conclusion

- GLS enables routing using geographic forwarding.
- GLS preserves the scalability of geographic forwarding.
- Current work:
  - Implementation of Grid in Linux

http://pdos.lcs.mit.edu/grid