Location Services

Otmar Caduff
The papers

- A Scalable Location Service for Geographic Ad Hoc Routing
  - J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, R. Morris; MobiCom 2000

- LLS: a Locality Aware Location Service for Mobile Ad Hoc Networks
  - I. Abraham, D. Dolev, D. Malkhi; DIALM-POMC 2004
Overview

- Introduction
  - Ad Hoc Networks
  - Problem overview
- The GLS approach
- The LLS approach
- Conclusion
Introduction

What is Ad Hoc?

- connection method for Wireless Networks
- dynamic
- spontaneous
- mobile
- without any additional infrastructure (i.e. no base stations, no routers, no directories)
  - nodes themselves have to contribute to routing
Applications

- Taxi/Police/Fire squad fleet
- Wearable computing
- Disaster relief and Disaster alarm
- Military/Security
- Meeting room/conference
- ...

...
How to route in Ad Hoc Networks?
How to route in Ad Hoc Networks?

- Simple solution: simply route towards the destination
How to route in Ad Hoc Networks?

- Simple solution: simply route towards the destination
How to route in Ad Hoc Networks?

- Geographic routing algorithms assume that besides the position of the source the position of the destination is also known.

- How to get this position?
  - Well known Location Server

<table>
<thead>
<tr>
<th></th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>35.5446</td>
<td>-105.9309</td>
</tr>
<tr>
<td>Bob</td>
<td>43.2113</td>
<td>3.8892</td>
</tr>
<tr>
<td>Carol</td>
<td>61.2113</td>
<td>-45.2203</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Get the Position of the destination

- What do we need:
  - The network must somehow know where a node is
    - a node has to publish its location
  - A node must be able to find out another node’s location, given its network identifier
    - a node wants to lookup another node’s location
- A lookup service offers publish and lookup primitives
A simple publish-driven approach

- When a node comes up or changes its position, it simply floods the network with its new location
  - Each node has to hold information about every other node in the network
    - memory...
    - traffic...
A simple lookup-driven approach

- When a node wants to know another node’s position, it simply floods the network with a query
  - The corresponding node answers with its position, also by flooding
    - no memory needed
    - even more traffic
Requirements

- No node should be a bottleneck
- The failure of a node should not affect the reachability of many other nodes
- Queries for the locations of nearby hosts should be satisfied with correspondingly local communication.
- The per-node storage and communication cost of the location service should grow as a small function of the total number of nodes.
Overview

- Introduction
  - Ad Hoc Networks
  - Problem overview
- The GLS approach
- The LLS approach
- Conclusion
GLS – Grid’s Location Service

- Each node has several other nodes as location servers and also acts as a location server on behalf of some other nodes.
- GLS uses geographic forwarding:
  - Here: simply route towards the destination (also known as greedy routing)
  - Could also be another geographic routing protocol
Partitioning the world

Invariant: a node is located in exactly one square of each size (no overlapping)
An order-x square contains always 4 order-(x-1) squares
A wants to find B (lookup)

- B has ID 17 (computed through hash function known by every node)
- A sends a request to the closest node to B (ID 17) for which A has location information. Next node does the same... Until B is reached.
- **Def.:** Node **closest** to B in ID space: node with least ID greater than B
- Circular ID space:

  1 → 4 → 17 → 25 → 29 → 41
B publishes its location

- B (ID 17) chooses three location servers for each level of the grid hierarchy in its related squares.
- B recruits nodes with IDs “close” to its own ID to serve as its location servers.
Problems...

- Works fine for uniform distributed nodes over the whole area
- neighboring points over border?
- mobility?
Overview

- Introduction
  - Ad Hoc Networks
  - Problem overview
- The GLS approach
- The LLS approach
- Conclusion
LLS – Locality-aware Location Service

- Locality-aware publish algorithm:
  - Cost of updating the location service due to a node moving from x to y is proportional to the distance between x and y.

- Locality-aware lookup algorithm:
  - Cost of a lookup operation is proportional to the cost of routing between source and target when the destination is known.
How it works – The spiral algorithm

- lookup: perform a spiral like search path
- a node publishes its location information in a set of virtual points that form a virtual spiral
- the lookup finds the location of the destination where the two spirals intersect
Why do the spirals intersect?

- Hierarchy of lattices with squares of size $2^k \times 2^k$ and $H$(node ID) as its origin
- When a node $t$ performs a publish, it aligns its lattices to $H$(node ID of $t$) and stores its location information to the 4 lattice points that are closest to $t$ (for each hierarchy-level)
- When another node wants to find out the position of $t$, it performs a lookup, aligned to $H$(node ID of $t$)

\[
W_k(t.id,x) = \{w_1, w_2, w_3, w_4\}
\]
Why is it locality-aware?

Lemma: Let $k$ be the minimal index such that $|st|=d<2^k$ then at least one of the nodes in $W_k(t.\text{id},s)$ contains a location pointer to node $t$. 

![Diagram showing nodes $W_1$, $W_2$, $W_3$, and $W_4$ with a point $s$ in the k-th phase with distance $2^k$.]
Why is it locality-aware?

- Theorem: For networks in which routing is $\Delta$-locality-aware, for any source $s$ and destination $t$ the expected cost of locating $t$ is $O(|st|)$

$$\sum_{i \leq k} 2 \cdot 4 \cdot 2^i \cdot \Delta = O(2^k) = O(|st|)$$
The spiral-flood algorithm

- Problem in the basic spiral algorithm: path cost from source to destination is low, but cost from source to first virtual point is high.

- Solution (spiral-flood algorithm): do spiral lookup as long as the cumulative cost is less than $4^{\text{phase}}$, then flood with depth $2^{\text{phase}}$. 
The LLS algorithm

- At each level, publish to 16 virtual points instead of only 4.
- Instead of publishing the location in $Z_i$, publish pointers to $W_{i-1}$. Store the location only in $Z_0$.

$$Z_k(t.\text{id},x) = \{w_1, \ldots, w_{16}\}$$
Overview

- Introduction
  - Ad Hoc Networks
  - Problem overview
- The GLS approach
- The LLS approach
- Conclusion
Conclusion

- Privacy?
- Own position always known?