GLS: A scalable location service

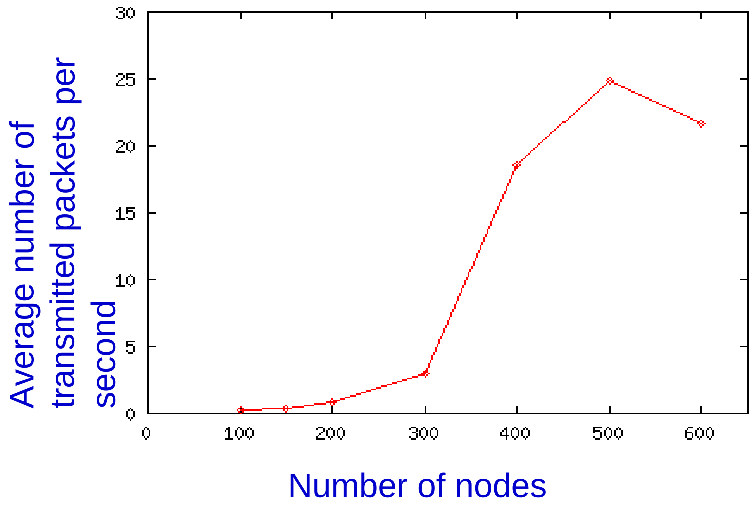
Overview

- Motivation for the GRID Location Service (GLS):
  - Scalable routing in large ad hoc networks containing several 1000 nodes
- Scalability of a protocol:
  - The number of packets that a node must forward and storeage requirement of the nodes for maintaining the states increase slowly as the network increases.

Classical Routing Strategies

- Traditional scalable Internet routing
  - Aggregation of the addresses (CIDR) inhibits the mobility
- Proactive methods (propagating the topology proactively (e.g. DSDV))
  - Reacts slowly to the mobility in larger networks
- On-demand flooding queries (e.g. DSR)
  - Protocol overhead becomes too high in larger networks

Flooding causes large overhead in bigger networks

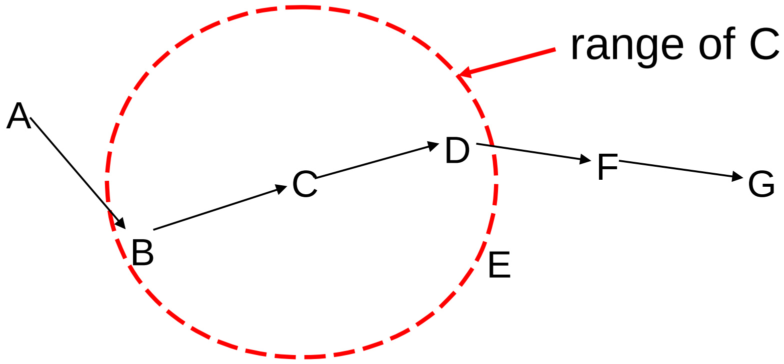


Flooding based „on-demand” routing works well in small networks.

How the routing can be improved without knowing the topology?

Geographic Forwarding is Scalable

- Assume, each node knows its own geographic position.



- A addresses the packet with the position of G
- C must only know the position of its neighbors to forward the packet in the direction of G.
- Geographic forwarding (position based routing) needs a location service!

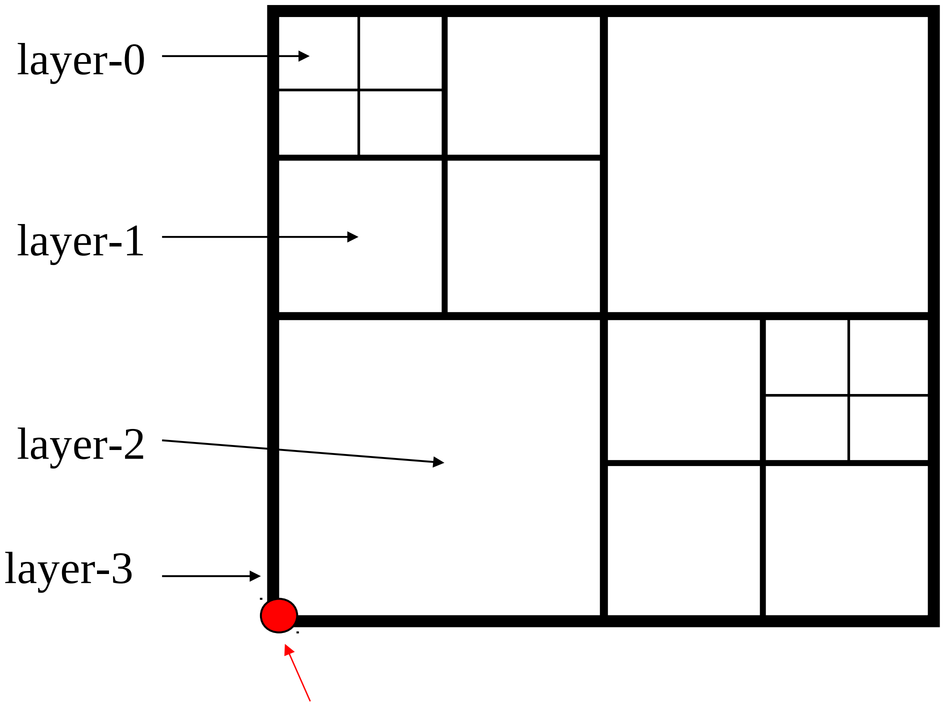
Possible Location Service Solutions

- Flooding for obtain the position of the destination node (LAR, DREAM).
  - Huge amount of flooding messages
- Centralized static location server
  - Not fault tolerant
  - Too high load on the server and on the nodes close to the server
  - The server can be far away even if the position of a nearby node is queried
  - If the network becomes partitioned, the server could be unreachable.
- Each node is serves as location server for some other nodes.
  - The load is well balanced
  - Fault tolerant

Desired Properties of a Distributed Location Service

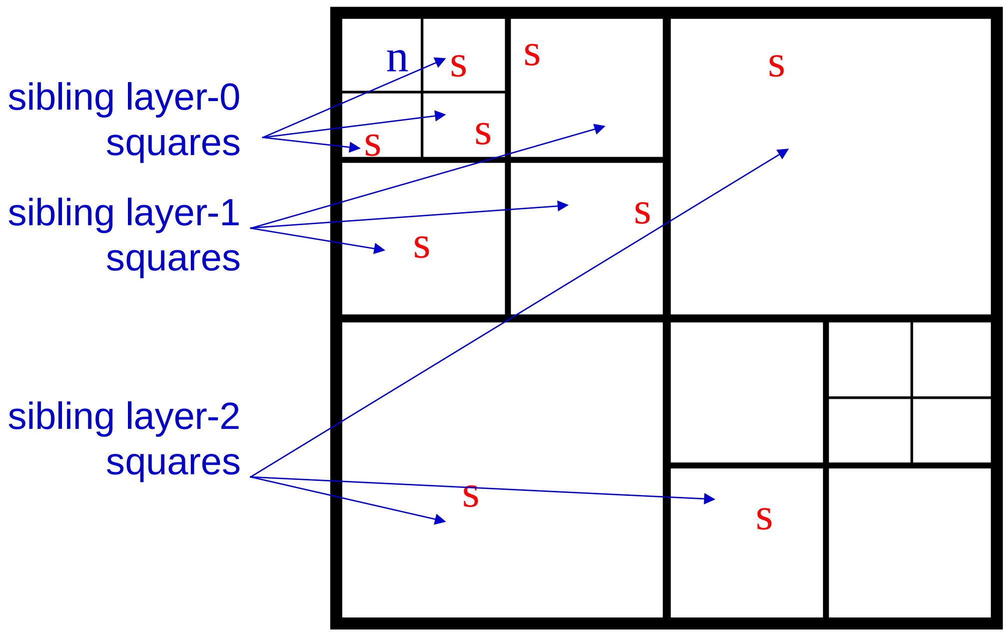
- The load on the nodes is uniformly balanced.
- Fault folerance.
- Queries about nearby nodes have to be local.
- Per node storage and communication costs increase slowly as the network grows.

Spacial Hierarchy in GLS



Each node knows the origin

3 Location Servers per Layer

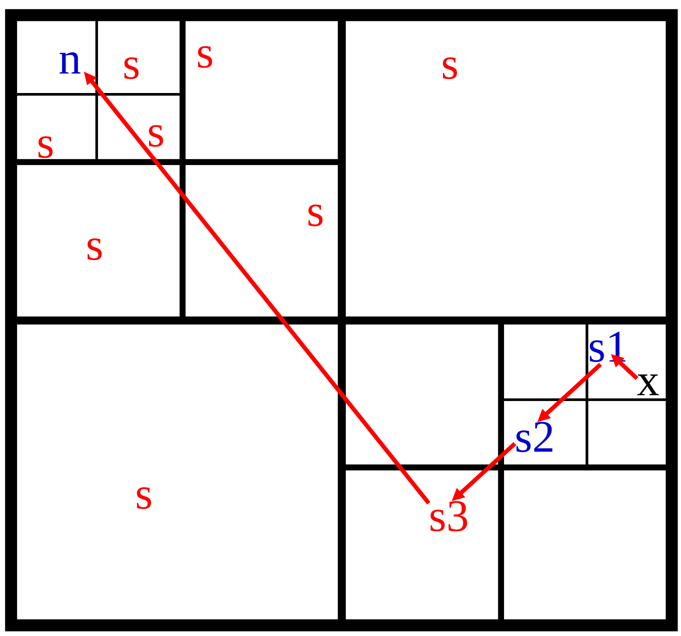


- The location server of n in square R is the node s, whose random ID is greater than the ID of n and the smallest among them (the successor of n in R).
- In each layer-0 square each node knows the position of all other nodes in that square.

Queries

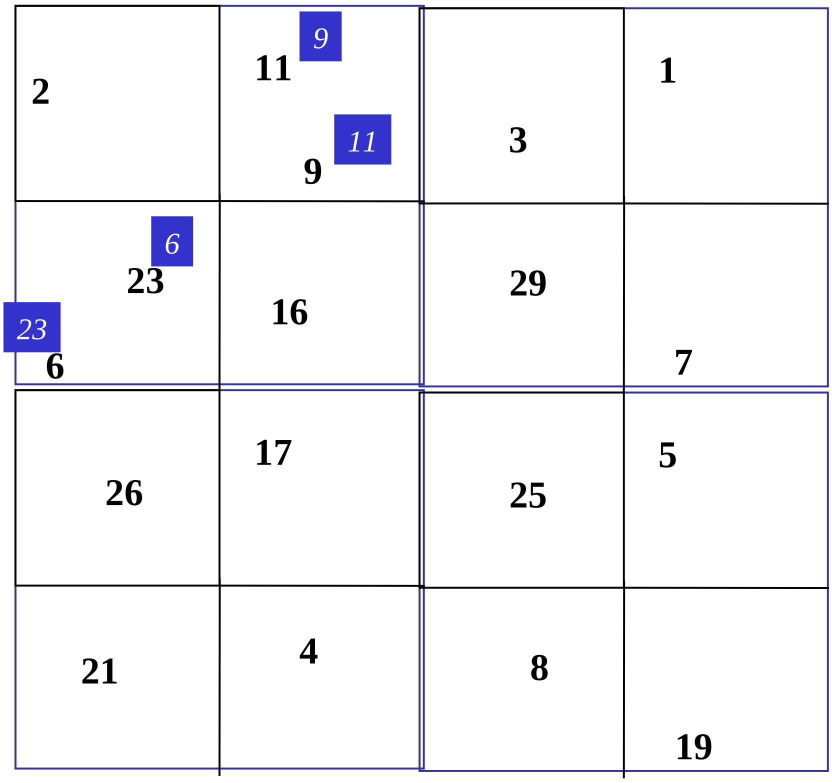
How x detects the position of n?

- x sends a location query packet
- Let i=0 and s_0=x
- If s_i stores the position of n, we are done
- Let R be the layer-i square containing s_i
- Let s_{i+1} be the node in R, whose position is stored at s_i and which is the successor of n in R (whose ID is greater than the ID of n and smallest among them).
- i ← i+1



Path of the location query

GLS Update (layer-0)

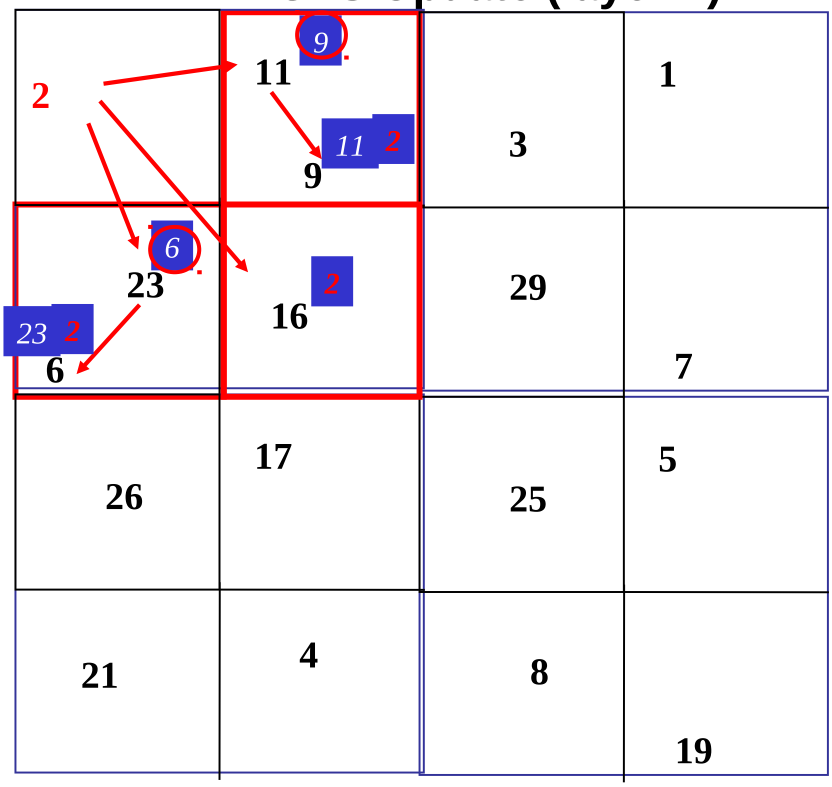


Invariant (for each layer):
For each node *n* in a square,
the successor of *n* in each sibling square „knows" *n.*

Base case:
Each node in a layer-0 square knows all other nodes in the same square

Content of location table

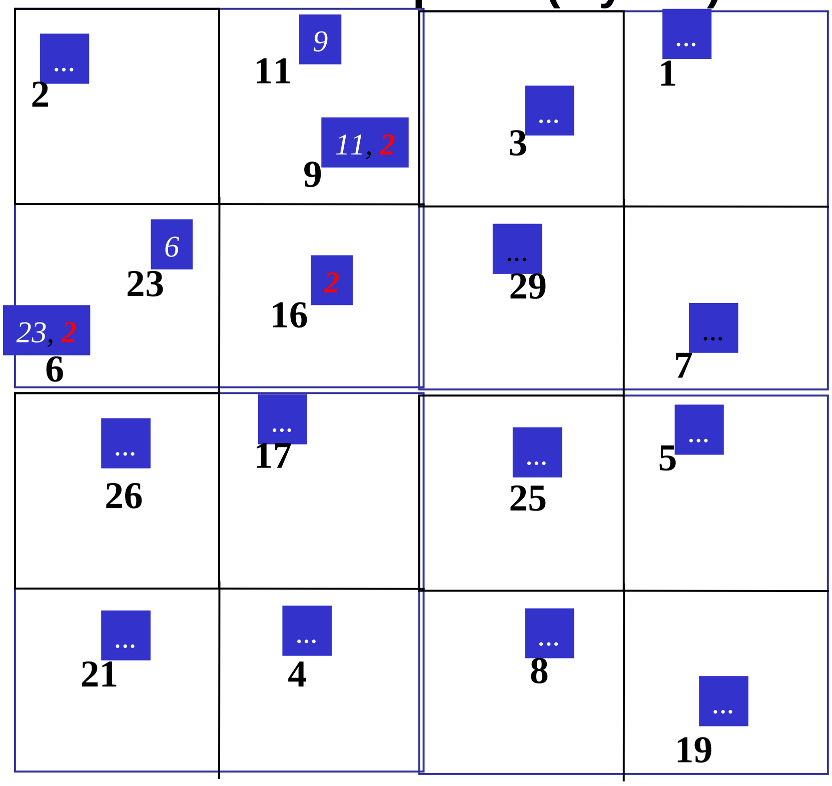
GLS Update (layer-1)



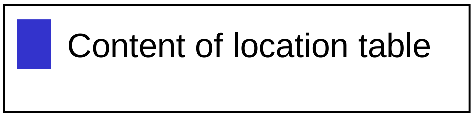
Invariant (for each layer):
For each node *n* in a square,
the successor of *n* in each sibling square „knows" *n.*

Content of location table
Location update

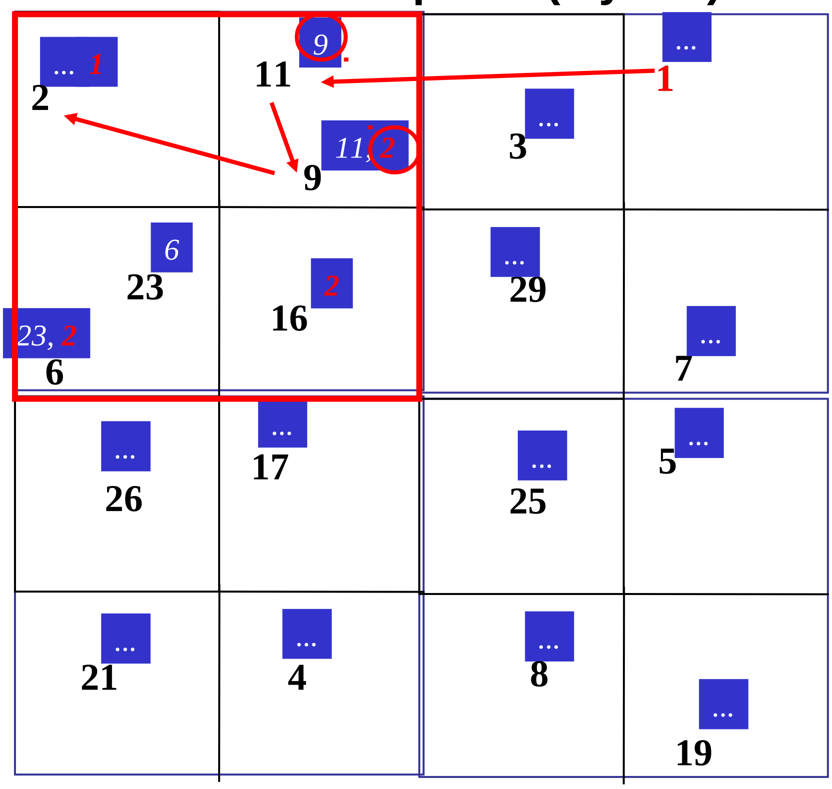
GLS Update (layer-1)



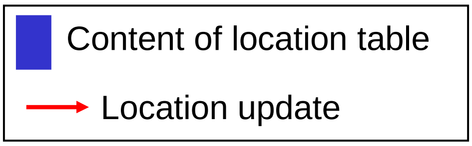
Invariant (for each layer):
For each node *n* in a square,
the successor of *n* in each sibling square „knows” *n.*



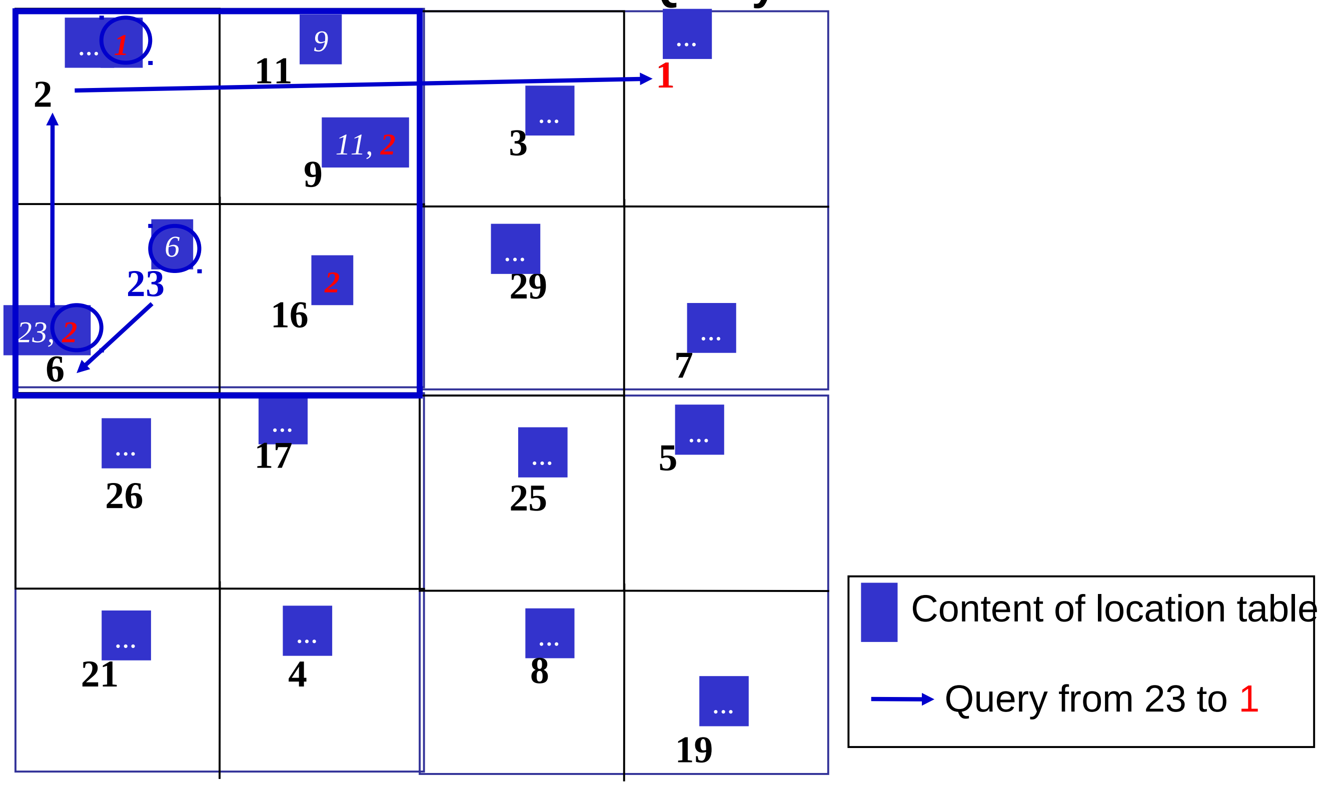
GLS Update (layer-2)



Invariant (for each layer):
For each node *n* in a square,
the successor of *n* in each sibling square „knows” *n.*



GLS Query



Challenges in a Mobil Network

- Obsolate information in the location servers.
- Tradeoff between the accuracy of the position information and the required update messages.
  - Determining the update rate based on the velocity.
  - Update distant servers rarer than nearby servers.
  - When a node leaves a square, we leave a forwarding pointer on the nodes until the update reaches the new server.

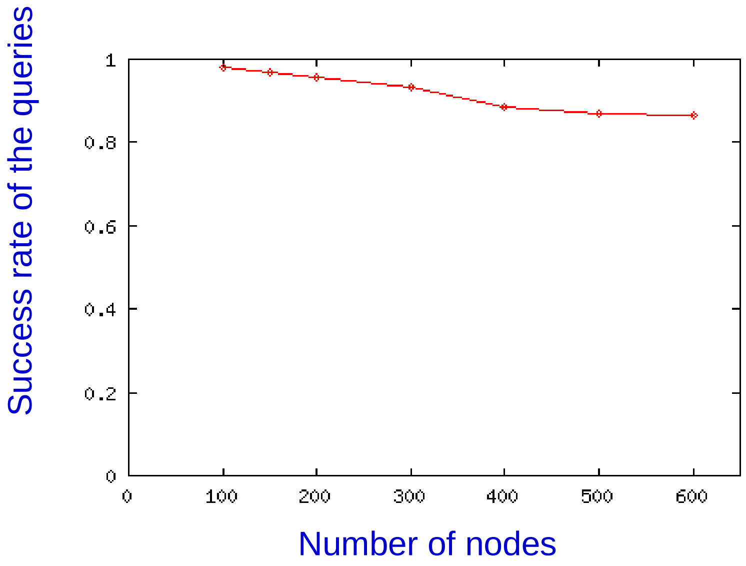
Performance

- How GLS works with mobility?
- How scalable is GLS?
- How GLS behaves, when nodes disappear?
- How local are the location queries for nearby nodes?

Simulation Environment

- Simulation: ns, with CMU wireless extension (IEEE 802.11)
- Mobility model:
  - random way-point (details later) velocity: 0-10 m/s
- Square area increases with the number of nodes.
- GLS layer-0 square: 250m x 250m
- 300 sec. per simulation

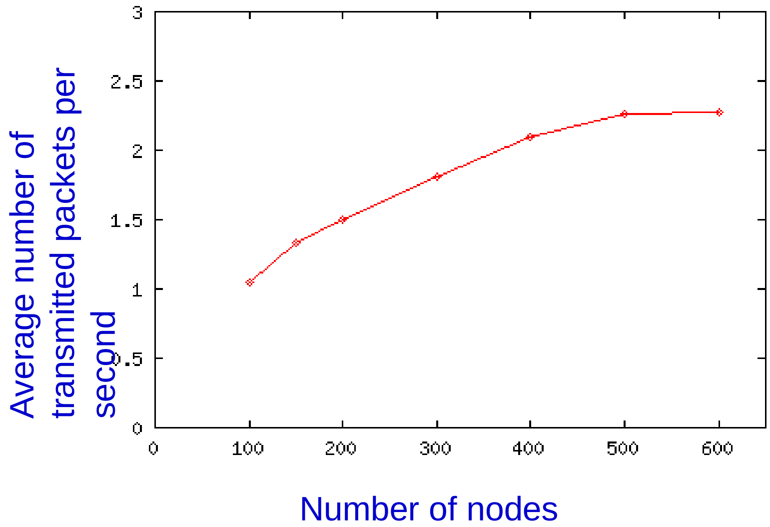
GLS Finds the Nodes in Large Networks



Biggest simulated network: 600 nodes, 2900x2900m (4-layers grid hierarchy)

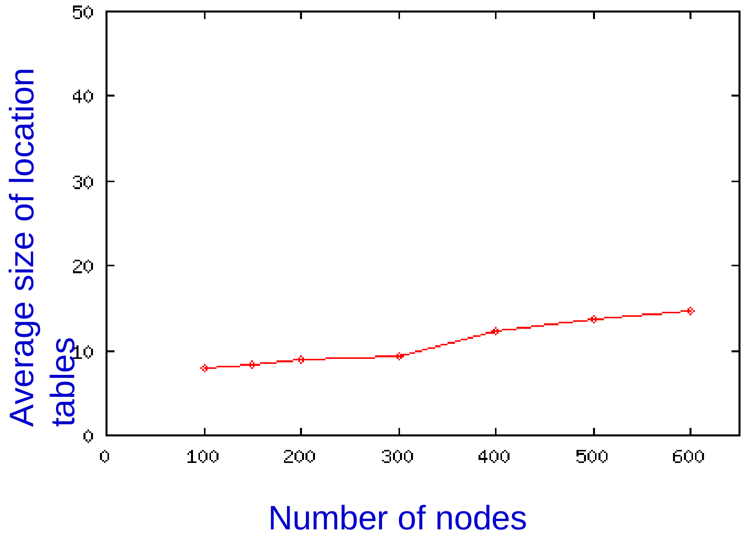
- no retransmission of unsuccesssful queries
- reason of an unsuccesful query: obsolate information about the destination node or the next server

GLS Protocol Overhead Increases Slowly



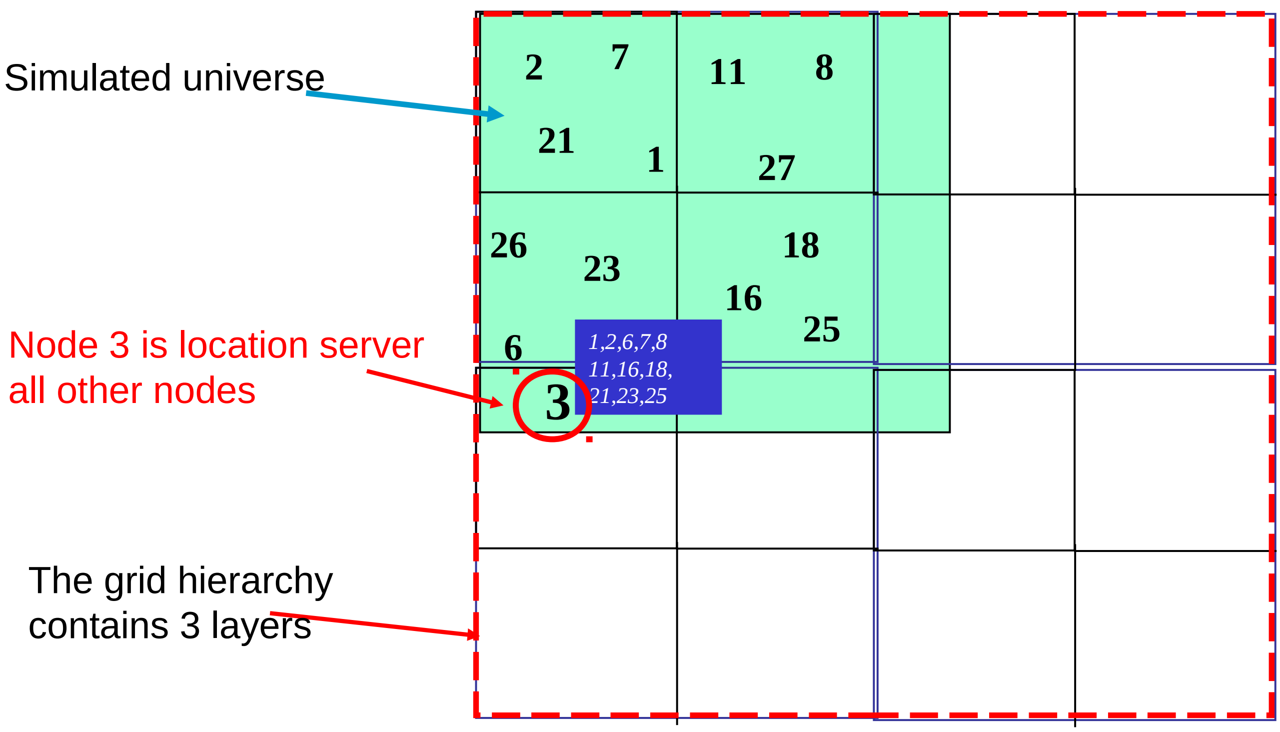
- Transmitted packets: GLS update, GLS query/reply

The Average Location Table Size is Small



- The average size of the location tables increases slowly

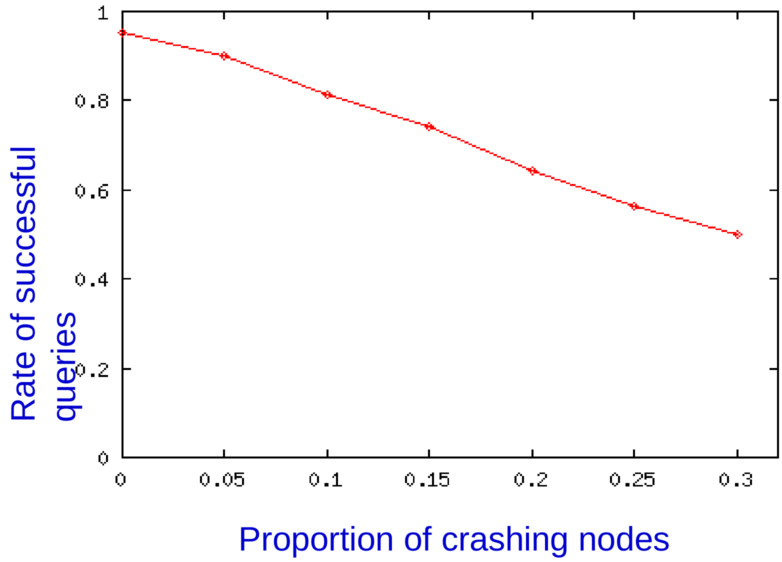
Non-Uniform Location Table Size



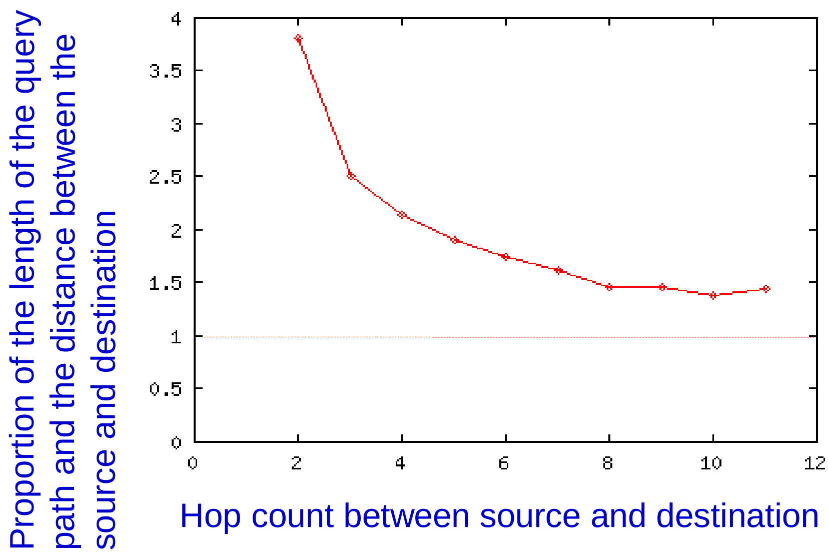
Possible solution: dynamically adjusting the boundary of the squares

GLS is Fault Tolerant

•The rate of successful location queries imediately after a fraction of nodes crashes simultaneously.
(The network initially contains 200 nodes)



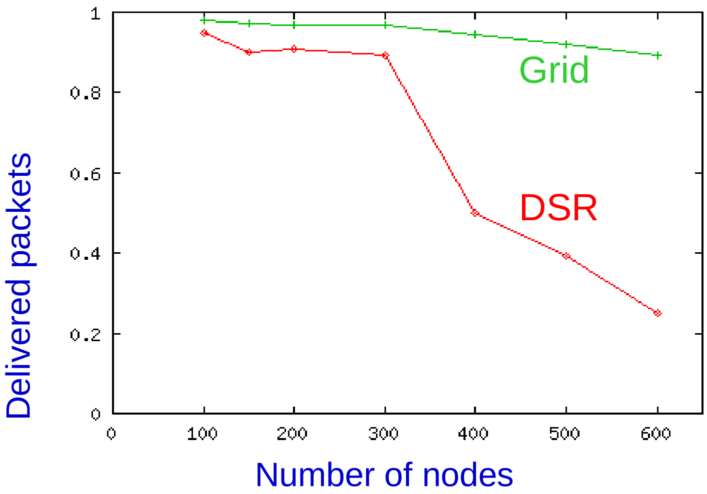
Length of Location Queries is Proportional to the Distance between the Source and Destination



GLS versus DSR

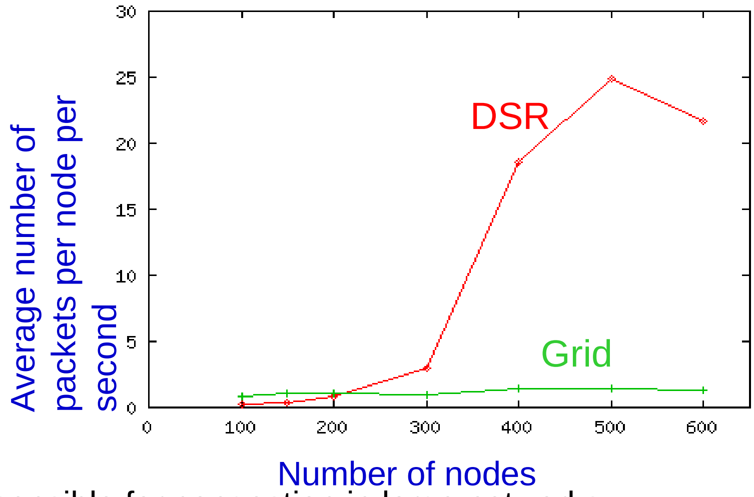
- DSR (Dynamic Source Routing)
  - In order to find the destination node, the source floods the network with a path query.
  - The answer to the query contains the path from the source to the destination
  - The source uses this path for sending data packets
- Simulation scenario:
  - 2Mbps radio bandwidth
  - 4 CBR sources. 128-byte packets/second for 20 seconds.
  - 50% of the nodes will be initialized during the 300 seconds of the simulation.

Proportion of Delivered Packets



- Geographic forwarding is more robust than source routing.
- DSR has problems with > 300 nodes?

Protocol Overhead



- DSR is sensible for congestion in large networks:
  - In order to find a new path instead of the broken path, the source must flood the network repeatedly
  - This causes congestion
- GLS queries cause a much lower load.
  - The queries use unicast, no flooding.
  - If the location query is not successful, data packets will not be sent.

Summary

- GLS allows the usage of geographic forwarding (position based routing)
- GLS keeps the scalability of geographic forwarding.
- GLS implementation for Linux: http://pdos.lcs.mit.edu/grid

Literature

- Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, Robert Morris: Scalable Location Service for Geographic Ad Hoc Routing, Proc. Mobicom, 2000.