Network Positioning for Wide-Area and Wireless Networks

Emin Gün Sirer

Department of Computer Science
Cornell University
Localization is Critical

Locality information is the building block for novel services in wired and wireless networks

Critical to find out where in the physical world nodes (and other items of interest) are

Locality-aware content, computing, routing, service discovery, event tracking in sensor networks, ...

Critical to select servers based on the position of target nodes

Find closest server, find centrally located node, find node within latency bounds
Sextant

Determining the location of nodes and events in wireless (ad hoc, sensor) networks
Localization in Wireless Networks

Infrastructure-based hardware (GPS) is the traditional solution

- Expensive
- Power-hungry
- Does not work indoors, without infrastructure

How well can we do with intelligent software and cheap, ubiquitous hardware?
Sextant Approach

Treat localization as a constraint-satisfaction problem

- Extract constraints aggressively from the network
- Disseminate them transitively
- Solve in a distributed manner
Sextant Properties

Positive Constraint

Accurate
Negative as well as positive information
Explicit representation

Practical
Constraint extraction
Deployed on Mica-2 motes, PDAs and laptops

Negative Constraint
Sextant Properties

Accurate
Negative as well as positive information
Explicit representation

Practical
Constraint extraction
Deployed on Mica-2 motes, PDAs and laptops

Need not be convex
May have holes
May have disconnected components
Sextant Properties

- Accurate
  - Negative as well as positive information
- Explicit representation

Practical
- Constraint extraction
- Deployed on Mica-2 motes, PDAs and laptops
Node Localization

Positive information
Node Localization

Intersection of Positive information
Node Localization

Negative information
Node Localization

Positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Refining position estimates
Sextant Approach

Location estimate: $\beta_x$

Set of positive constraints: $\Gamma_x$

Set of negative constraints: $\Theta_x$

$$\beta_x = \bigcap (p \in \Gamma_x) \setminus \bigcup (n \in \Theta_x)$$
Sextant Areas

Represent areas explicitly

Use Bezier curves to bound bezier regions
Four control points define a curve
Union and intersection are implemented efficiently

Not a point estimate!
Ideally, applications should take the bezier region as input
Can generate point estimate from bezier regions
Localizing Events

Hot area in sensor networks

The Sextant approach provides a comprehensive, unified framework

Differences from node localization

Constraints from sensors, not wireless radios

Boolean connected/not connected to sensed/not sensed

Annotate resulting areas with probabilities
Event localization

Decompose space into a grid, propagate probabilities

Calculate normalized Bayesian probabilities
Event Localization

Start with initial Sextant node regions
Event Localization

An event occurs
Event Localization

Sextant is used for event localization
Event Localization

Sextant is used for event localization
Event Localization

Event localized
Event Localization

Event used for node localization!
Event Localization

Event used to refine node location!
Event Localization

Event detection helps refine node positions!
Meridian

Selecting nodes based on location
(without knowing their actual location in the real world)
Network Location Service

Select nodes based on a set of network properties

Real-world problems:

- Locate closest game server
- Distribute web-crawling to nearby hosts
- Perform efficient application level multicast
- Satisfy a Service Level Agreement
- Provide inter-node latency bounds for clusters

Underlying abstract problems:

- Finding closest node to target
- Finding the closest node to the center of a set of targets
- Finding a node that is $<r_i$ ms from target $t_i$ for all targets
Current State-of-the-Art: Virtual Coordinates

Maps Internet latencies into low dimensional space
   GNP, Vivaldi, Lighthouse, ICS, VL, BBS, PIC, NPS, etc.

Reduces number of real-time measurements

3 practical problems:
   Introduces inherent embedding error

   A snapshot in time of the network location of a node
      Coordinates become stale over time
      Latency estimates based on coordinates computed at different times can lead to additional errors

   Requires additional P2P substrate to solve network location problems without centralized servers or \( O(N) \) state
Meridian Approach

Solve node selection directly without computing coordinates
  Combine query routing with active measurements

3 Design Goals:
  Accurate: Find satisfying nodes with high probability
  General: Users can fully express their network location requirements
  Scalable: $O(\log N)$ state per node, $O(\log D)$ hops per query

Design tradeoffs:
  Active measurements incur higher query latencies
  Overhead more dependent on query load
Meridian Operation

Framework:

Loosely structured overlay network

Algorithms:

Solve network location problems in $O(\log D)$ hops

Language:

General-purpose language for expressing network location requirements
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially

- Logarithmic # of peers per ring
- Favors nearby neighbors
- Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially

- Logarithmic # of peers per ring
- Favors nearby neighbors
- Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially
- Logarithmic # of peers per ring
- Favors nearby neighbors
- Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially
  - Logarithmic # of peers per ring
  - Favors nearby neighbors
  - Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Closest Node Discovery

Multi-hop search

Similar to finding the closest identifier in DHTs

Replaces virtual identifiers with physical latencies

Each hop exponentially reduces the distance to the target

Reduction threshold $\beta$ for $0 \leq \beta < 1$

Only take another hop if a peer node is $\beta$ times closer

Limits # of probed peers through triangle inequality
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery

The diagram illustrates the concept of finding the closest node. Node C is highlighted as the closest node to the query point represented by T. The distance d between C and T is the shortest among all other nodes in the vicinity.
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Meridian Theoretical Analysis

Analytical guarantees for closest node discovery

Meridian can find the closest node with high probability

Given nodes in a space with a *doubling* metric

As well as a *growth constrained* metric

Scales well with increasing system size

Does not lead to hot spots
Central Leader Election

Select the closest node to the center of a set of targets

Multi-cast trees can place central nodes higher in the hierarchy

Algorithm similar to closest node discovery

Minimizes avg. latency to a set of targets instead of one target

Uses distance metric $d_{\text{avg}}$ instead of $d$

Inter-node latencies of targets not known

Need to be conservative in pruning peers
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Multi-constraint System

Find a node that satisfies a set of latency constraints

ISP can find a server that can satisfy a SLA with a client
Grid users can find a set of nodes with a bounded inter-node latency

There exists a solution space, containing 0 or more nodes

Only a solution point in previous problems

\[
Re(s) = \sum_{i=1}^{u} \max(0, d_i - range_i)^2 \]

Metric \( s \):

\( s = 0 \) when all constraints are satisfied

Sum of squares places more weight on fringe constraints

Allows for faster convergence to solution space

Other metrics can be used, square works well in practice
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Meridian Query Language

Variant of C/Python

  Safe, polymorphic, and dynamically-typed
  Includes an extensive set of library functions

Allows users to:

  Access multi-resolution rings
  Issue latency probes
  Forward queries to peers

Tight resource limits on:

  Execution time of query
  Number of hops
  Amount of memory allocated
Evaluation

Evaluated our system through a large scale simulation and a PlanetLab deployment

Simulation parameterized by real latency measurements

- 2500 DNS servers, latency between 6.25 million node pairs
- DNS servers are authorities name servers for domains found in the Yahoo! web directory

We evaluated system sizes of up to 2000 nodes
- 500 nodes reserved as targets
Evaluation: Closest Node Discovery

Meridian has an order of magnitude less error than virtual coordinate schemes.
Evaluation: Closest Node Discovery

CDF of relative error shows Meridian is more accurate for both typical nodes and fringe nodes.
Evaluation: Closest Node Discovery

With $k = \lfloor \log_{1.6} N \rfloor$, error and query latency remain constant as $N$ increases.

Average query latency determined by closest node in each ring.
Evaluation: Central Leader Election

Meridian incurs significantly less relative error
Evaluation: Multi-constraint System

Categorized multi-constraint queries by its difficulty

Difficulty a measure of the number of nodes in solution space

Success rate for queries that can be satisfied by only 0.5% of the nodes:

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Meridian</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>91%</td>
<td>35%</td>
</tr>
<tr>
<td>3</td>
<td>90%</td>
<td>19%</td>
</tr>
<tr>
<td>4</td>
<td>91%</td>
<td>11%</td>
</tr>
</tbody>
</table>
Evaluation: PlanetLab Deployment

A PlanetLab deployment of 166 nodes shows the closest node discovery accuracy to be very close to the simulation results.

Have expanded deployment to 325 PlanetLab nodes supporting all 3 applications and MQL.
Implementation

Includes query language and the 3 protocols

Works with firewalled hosts

Can use DNS queries, TCP connect times, and Meridian UDP packets to measure latency

Optimizations:

  Measurement cache reduces query latency

  Ring management scheme to select more diverse peers
ClosestNode.com is a DNS redirection service that returns the IP address of closest node to the client

e.g. cobweb.closestnode.com will resolve to the closest CobWeb DHT node to the requesting client

Requires minimal changes to the service

Linking the Meridian library and calling one function at startup

Or add standalone Meridian server to start script

No changes required for the client

Can register your service at:

http://www.closestnode.com
Meridian Summary

A lightweight accurate system for selecting nodes
Combines query routing with active measurements
An order of magnitude less error than virtual coordinates
Solves the network location problem directly
Does not need to be paired with CAN

Code, data, demos and more information at
http://www.cs.cornell.edu/People/egs/meridian
Octant

Determining the physical location of Internet nodes in the real world

*(Combining Sextant with Meridian...)*
Octant

Often need to determine the physical location of a machine on the Internet

- Provide customized services
- Trace user activity
- Perform monitoring and locate attackers

Need to map from IP Address to geographic location

- IP to Zip Code: Static, Course-grained, Inaccurate

Need a dynamic, accurate way of finding physical location of machines

- Must work even if host is behind NAT, firewall or in a VPN
Octant Approach

Find general dependency between network latency and physical distance

Set up a system of constraints based on latency measurements to known landmark nodes

- Aggressively extract constraints
- Use both positive and negative information

Solve the system geometrically, yielding the set of physical areas on the globe where a target may be located
Latency-Distance Relationship

Internet latencies correlated with distance
Positive and Negative Information

A latency probe establishes the minimum and maximum distances between a target T and chosen landmarks.

Geometric intersection yields target location.
Cylindrical Equidistant Projection

- Use Bézier curves to bound the areas in which a node can appear
- Map curves onto projected 2D globe
Summary

Octant is a dynamic and accurate Internet host localization service.

Achieves high fidelity by using both positive and negative information.

Can be used to determine the physical location of any node without user input.
Network Positioning for Wide-Area and Wireless Networks

Emin Gün Sirer

Department of Computer Science
Cornell University
Localization is Critical

Locality information is the building block for novel services in wired and wireless networks

Critical to find out where in the physical world nodes (and other items of interest) are
  Locality-aware content, computing, routing, service discovery, event tracking in sensor networks, ...

Critical to select servers based on the position of target nodes
  Find closest server, find centrally located node, find node within latency bounds
Sextant

Determining the location of nodes and events in wireless (ad hoc, sensor) networks
Localization in Wireless Networks

Infrastructure-based hardware (GPS) is the traditional solution

- Expensive
- Power-hungry
- Does not work indoors, without infrastructure

How well can we do with intelligent software and cheap, ubiquitous hardware?
Sextant Approach

Treat localization as a constraint-satisfaction problem
  Extract constraints aggressively from the network
  Disseminate them transitively
  Solve in a distributed manner
Sextant Properties

Accurate
Negative as well as positive information
Explicit representation

Practical
Constraint extraction
Deployed on Mica-2 motes, PDAs and laptops
### Sextant Properties

<table>
<thead>
<tr>
<th>Accurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Need not be convex</td>
</tr>
<tr>
<td>May have holes</td>
</tr>
<tr>
<td>May have disconnected components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint extraction</td>
</tr>
<tr>
<td>Deployed on Mica-2 motes, PDAs and laptops</td>
</tr>
</tbody>
</table>
Sextant Properties

Accurate
- Negative as well as positive information
- Explicit representation

Practical
- Constraint extraction
- Deployed on Mica-2 motes, PDAs and laptops
Node Localization

Positive information
Node Localization

Intersection of Positive information
Node Localization

Negative information
Node Localization

Positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Transitive dissemination of positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Combining negative and positive information
Node Localization

Refining position estimates
Sextant Approach

Location estimate: $\beta_x$

Set of positive constraints: $\Gamma_x$

Set of negative constraints: $\Theta_x$

$$\beta_x = \bigcap (p \in \Gamma_x) \setminus \bigcup (n \in \Theta_x)$$
Sextant Areas

Represent areas explicitly
  Use Bezier curves to bound bezier regions
  Four control points define a curve
  Union and intersection are implemented efficiently

Not a point estimate!
  Ideally, applications should take the bezier region as input
  Can generate point estimate from bezier regions
Localizing Events

Hot area in sensor networks
The Sextant approach provides a
comprehensive, unified framework
Differences from node localization
  Constraints from sensors, not wireless radios
  Boolean connected/not connected to sensed/not sensed
  Annotate resulting areas with probabilities
Event localization

- Decompose space into a grid, propagate probabilities
- Calculate normalized Bayesian probabilities
Event Localization

Start with initial Sextant node regions
Event Localization

An event occurs
Event Localization

Sextant is used for event localization
Event Localization

Sextant is used for event localization
Event Localization

Event localized
Event Localization

Title: sextant
Creator: Tgif-4.1.43-0
CreationDate: Sun M

Event used for node localization!
Event Localization

Event used to refine node location!
Event Localization

Event detection helps refine node positions!
Meridian

Selecting nodes based on location

*(without knowing their actual location in the real world)*
Network Location Service

Select nodes based on a set of network properties

Real-world problems:

Locate closest game server
Distribute web-crawling to nearby hosts
Perform efficient application level multicast
Satisfy a Service Level Agreement
Provide inter-node latency bounds for clusters

Underlying abstract problems

Finding closest node to target
Finding the closest node to the center of a set of targets
Finding a node that is \(< r \), ms from target \( t \), for all targets
Current State-of-the-Art: Virtual Coordinates

Maps Internet latencies into low dimensional space
  GNP, Vivaldi, Lighthouse, ICS, VL, BBS, PIC, NPS, etc.

Reduces number of real-time measurements

3 practical problems:
  Introduces inherent embedding error
  A snapshot in time of the network location of a node
    Coordinates become stale over time
  Latency estimates based on coordinates computed at different times can lead to additional errors
  Requires additional P2P substrate to solve network location problems without centralized servers or $O(N)$ state
Meridian Approach

Solve node selection directly without computing coordinates
   Combine query routing with active measurements

3 Design Goals:
   Accurate:  Find satisfying nodes with high probability
   General:   Users can fully express their network location requirements
   Scalable:  O(log N) state per node, O(log D) hops per query

Design tradeoffs:
   Active measurements incur higher query latencies
   Overhead more dependent on query load
Meridian Operation

Framework:
Loosely structured overlay network

Algorithms:
Solve network location problems in $O(\log D)$ hops

Language:
General-purpose language for expressing network location requirements
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially

Logarithmic # of peers per ring
Favors nearby neighbors
Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery

\[ r_s = s^2 \]
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially

- Logarithmic # of peers per ring
- Favors nearby neighbors
- Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Multi-resolution Rings

Organize peers into small fixed number of concentric rings

Radii of rings grow outwards exponentially

- Logarithmic # of peers per ring
- Favors nearby neighbors
- Retains a sufficient number of pointers to remote regions

Gossip protocol used for peer discovery
Multi-resolution Rings

- Organize peers into small fixed number of concentric rings
- Radii of rings grow outwards exponentially
  - Logarithmic # of peers per ring
  - Favors nearby neighbors
  - Retains a sufficient number of pointers to remote regions
- Gossip protocol used for peer discovery

\[ r = s^2 \]
Closest Node Discovery

Multi-hop search

Similar to finding the closest identifier in DHTs
Replaces virtual identifiers with physical latencies
Each hop exponentially reduces the distance to the target
Reduction threshold $\beta$ for $0 \leq \beta < 1$
Only take another hop if a peer node is $\beta$ times closer
Limits # of probed peers through triangle inequality
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Closest Node Discovery
Meridian Theoretical Analysis

Analytical guarantees for closest node discovery

Meridian can find the closest node with high probability

Given nodes in a space with a *doubling* metric
As well as a *growth* constrained metric

Scales well with increasing system size

Does not lead to hot spots
Central Leader Election

Select the closest node to the center of a set of targets
- Multi-cast trees can place central nodes higher in the hierarchy

Algorithm similar to closest node discovery

Minimizes avg. latency to a set of targets instead of one target
- Uses distance metric $d_{avg}$ instead of $d$

Inter-node latencies of targets not known
- Need to be conservative in pruning peers
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Central Leader Election
Multi-constraint System

Find a node that satisfies a set of latency constraints

- ISP can find a server that can satisfy a SLA with a client
- Grid users can find a set of nodes with a bounded inter-node latency

There exists a solution space, containing 0 or more nodes

- Only a solution point in previous problems

\[ R_{\text{met}} = \sum_{i=1}^{n} \max(0, d_i - range)^2 \] metric \( s \):

- \( s = 0 \) when all constraints are satisfied
- Sum of squares places more weight on fringe constraints
- Allows for faster convergence to solution space

Other metrics can be used, square works well in practice
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Multi-constraint System
Meridian Query Language

Variant of C/Python
- Safe, polymorphic, and dynamically-typed
- Includes an extensive set of library functions

Allows users to:
- Access multi-resolution rings
- Issue latency probes
- Forward queries to peers

Tight resource limits on:
- Execution time of query
- Number of hops
- Amount of memory allocated
Evaluation

Evaluated our system through a large scale simulation and a PlanetLab deployment

Simulation parameterized by real latency measurements
- 2500 DNS servers, latency between 6.25 million node pairs
- DNS servers are authorities name servers for domains found in the Yahoo! web directory

We evaluated system sizes of up to 2000 nodes
- 500 nodes reserved as targets
Evaluation: Closest Node Discovery

Meridian has an order of magnitude less error than virtual coordinate schemes.
Evaluation: Closest Node Discovery

CDF of relative error shows Meridian is more accurate for both typical nodes and fringe nodes.
Evaluation: Closest Node Discovery

With \( k = \lceil \log_{1.6} N \rceil \), error and query latency remain constant as \( N \) increases.
Evaluation: Central Leader Election

Meridian incurs significantly less relative error
Evaluation: Multi-constraint System

Categorized multi-constraint queries by its difficulty

Difficulty a measure of the number of nodes in solution space

Success rate for queries that can be satisfied by only 0.5% of the nodes:

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Meridian</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>91%</td>
<td>0.5%</td>
</tr>
<tr>
<td>3</td>
<td>35%</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>91%</td>
<td></td>
</tr>
</tbody>
</table>

Percentage of nodes that can satisfy constraints (CD)
Evaluation: PlanetLab Deployment

A PlanetLab deployment of 166 nodes shows the closest node discovery accuracy to be very close to the simulation results.

Have expanded deployment to 325 PlanetLab nodes supporting all 3 applications and MQL.
Implementation

Includes query language and the 3 protocols

Works with firewalled hosts

Can use DNS queries, TCP connect times, and Meridian UDP packets to measure latency

Optimizations:

- Measurement cache reduces query latency
- Ring management scheme to select more diverse peers
ClosestNode.com

ClosestNode.com is a DNS redirection service that returns the IP address of closest node to the client

  e.g. cobweb.closestnode.com will resolve to the closest CobWeb DHT node to the requesting client

Requires minimal changes to the service

  Linking the Meridian library and calling one function at startup
  Or add standalone Meridian server to start script

No changes required for the client

Can register your service at:

  http://www.closestnode.com
Meridian Summary

A lightweight accurate system for selecting nodes
Combines query routing with active measurements
An order of magnitude less error than virtual coordinates
Solves the network location problem directly
Does not need to be paired with CAN
Code, data, demos and more information at
http://www.cs.cornell.edu/People/egs/meridian
Octant

Determining the physical location of Internet nodes in the real world

(Combining Sextant with Meridian...)

96
Octant

Often need to determine the physical location of a machine on the Internet

- Provide customized services
- Trace user activity
- Perform monitoring and locate attackers

Need to map from IP Address to geographic location

- IP to Zip Code: Static, Course-grained, Inaccurate

Need a dynamic, accurate way of finding physical location of machines

- Must work even if host is behind NAT, firewall or in a VPN
Octant Approach

Find general dependency between network latency and physical distance

Set up a system of constraints based on latency measurements to known landmark nodes

  Aggressively extract constraints
  Use both positive and negative information

Solve the system geometrically, yielding the set of physical areas on the globe where a target may be located
Latency-Distance Relationship

Internet latencies correlated with distance
Positive and Negative Information

A latency probe establishes the minimum and maximum distances between a target $T$ and chosen landmarks.

Geometric intersection yields target location.
Cylindrical Equidistant Projection

- Use Bézier curves to bound the areas in which a node can appear
- Map curves onto projected 2D globe
Summary

Octant is a dynamic and accurate Internet host localization service

Achieves high fidelity by using both positive and negative information

Can be used to determine the physical location of any node without user input