Vivaldi
Practical, Distributed Internet Coordinates

Frank Dabek Russ Cox Robert Morris Frans Kaashoek

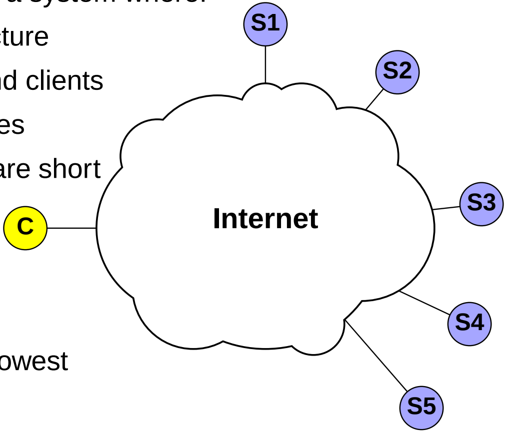
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology

ACM SIGCOMM 2004

Problem: Predicting Round Trip Times on Internet

Example: server selection in a system where:

- no centralized infrastructure
- nodes act as servers and clients
- many thousands of nodes
- exchanges with server are short
- server choice changes for each exchange



Want to choose server with lowest round trip time to client.

How?

Possible Solutions

- Can avoid predictions, wasting time or bandwidth:
  - measure RTT on demand
  - measure RTT in advance
  - talk to multiple servers at once
- Can predict using synthetic coordinates as in GNP (Infocom 2002).

Synthetic Coordinates with GNP

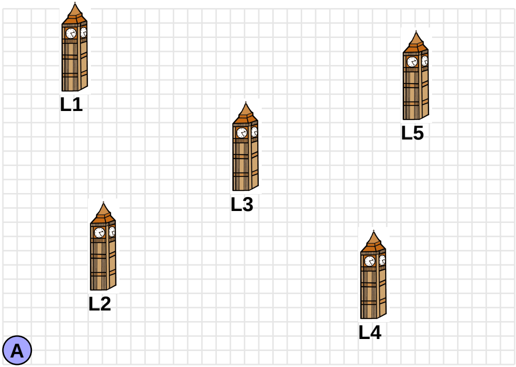
GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Synthetic Coordinates with GNP

GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Node A pings landmarks to compute its own position.

	Coord	Dist
L1	(40,320)	
L2	(60,180)	
L3	(160,250)	
L4	(250,160)	
L5	(280,300)	

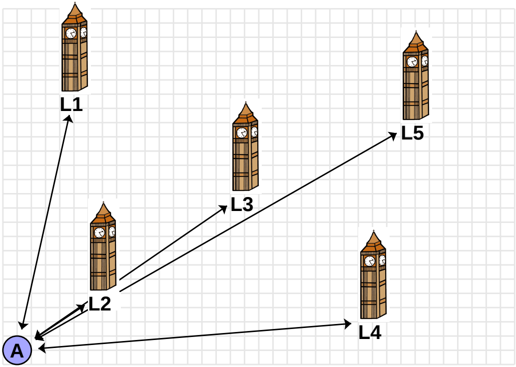


Synthetic Coordinates with GNP

GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Node A pings landmarks to compute its own position.

	Coord	**Dist**
L1	(40,320)	117 ms
L2	(60,180)	201 ms
L3	(160,250)	110 ms
L4	(250,160)	223 ms
L5	(280,300)	143 ms

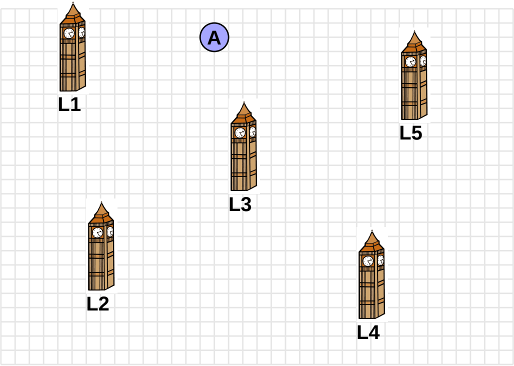


Synthetic Coordinates with GNP

GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Node A pings landmarks to compute its own position.

	Coord	Dist
L1	(40,320)	117 ms
L2	(60,180)	201 ms
L3	(160,250)	110 ms
L4	(250,160)	223 ms
L5	(280,300)	143 ms

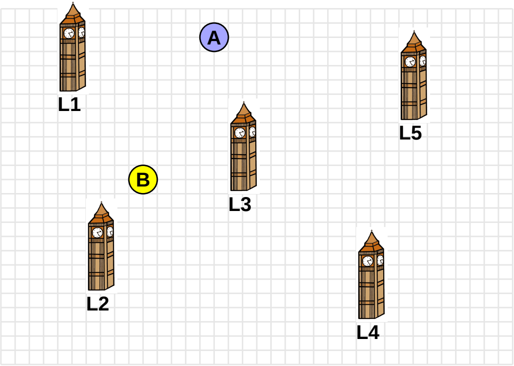


Synthetic Coordinates with GNP

GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Node A pings landmarks to compute its own position.

Node B does the same.



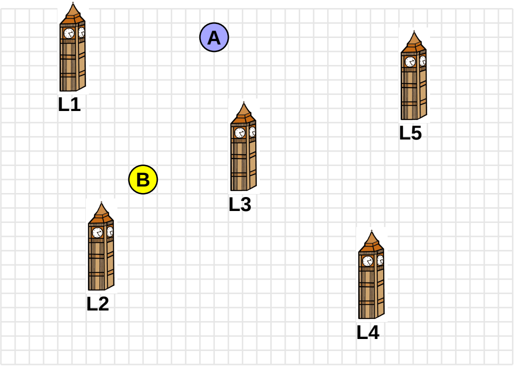
Synthetic Coordinates with GNP

GNP assigns Euclidean coordinates to nodes such that coordinate distance predicts round trip time.

Node A pings landmarks to compute its own position.

Node B does the same.

RTT between A and B
is predicted by the distance
between their coordinates,
without direct measurement.



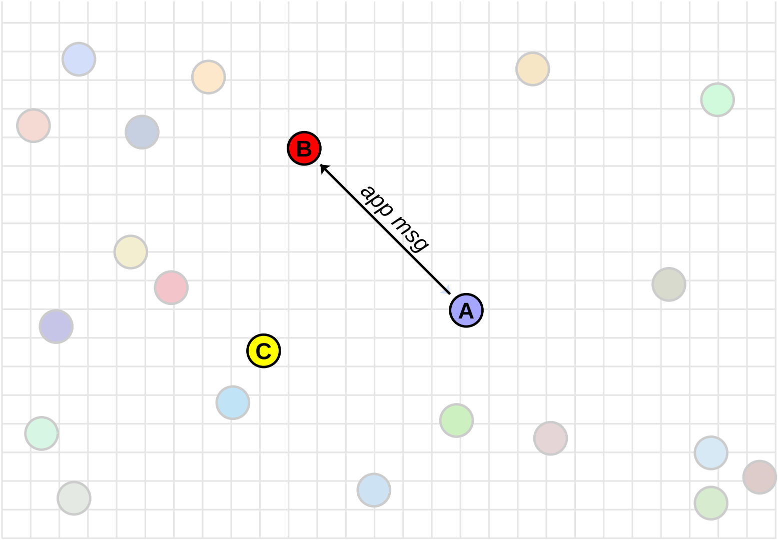
Vivaldi Overview

Vivaldi is a decentralized method for computing synthetic coordinates

- Piggyback on application traffic
- Node updates its own coordinates in response to sample
- Each node need only contact a small fraction of the other nodes

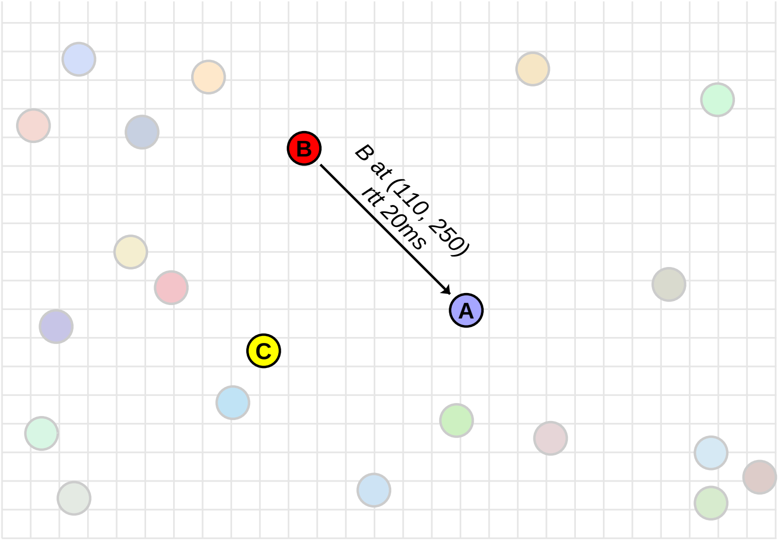
Vivaldi Example

Follow node A through a sequence of communications.



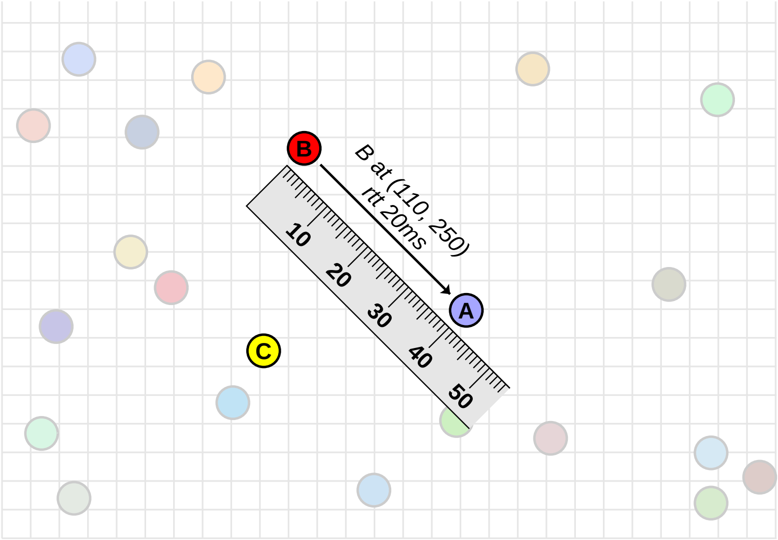
Vivaldi Example

A obtains B's coordinates, RTT.



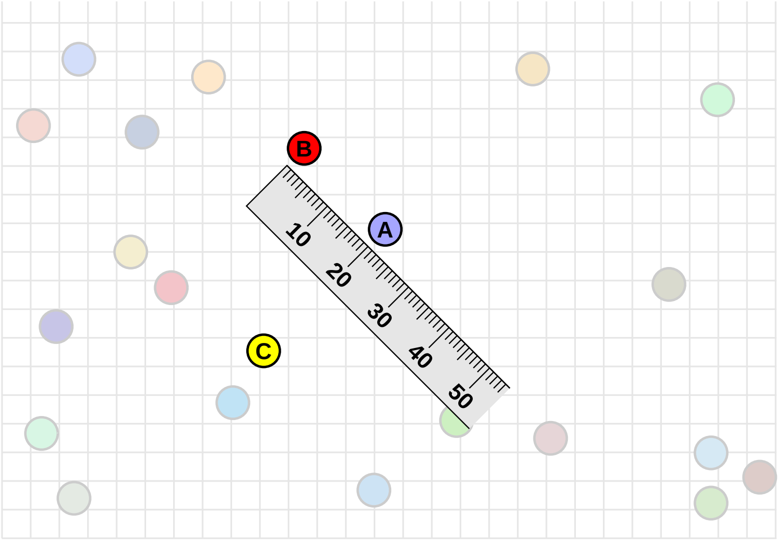
Vivaldi Example

A computes distance to B in coordinate space.



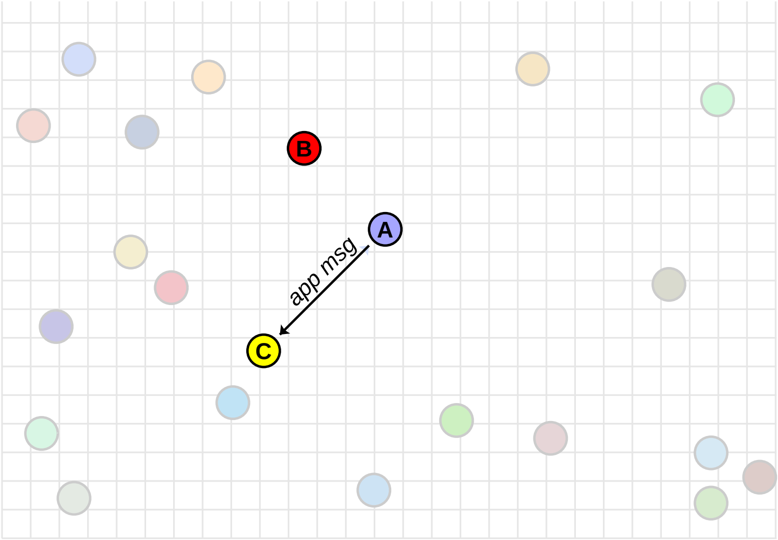
Vivaldi Example

A adjusts coordinates so distance matches actual RTT.



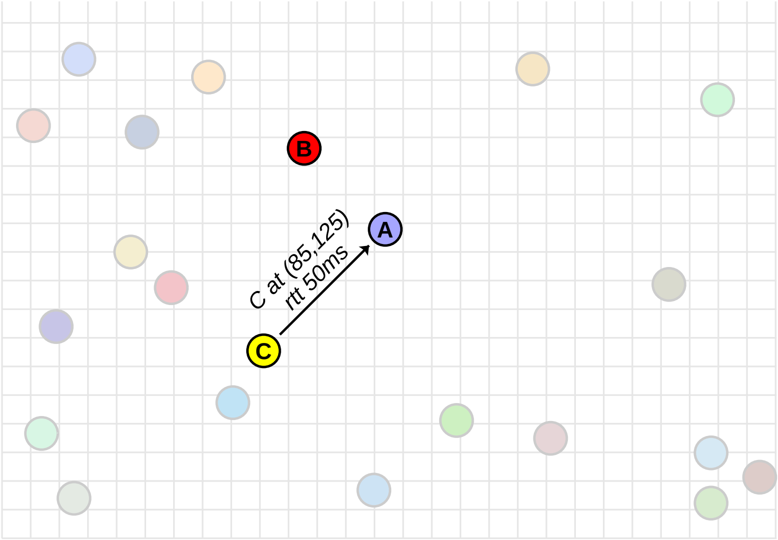
Vivaldi Example

Follow node A through communication with C.



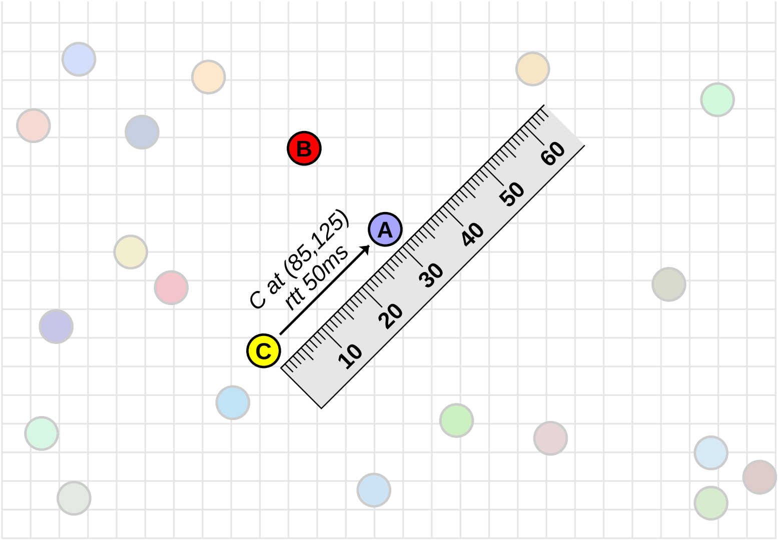
Vivaldi Example

A obtains C's coordinates, RTT.



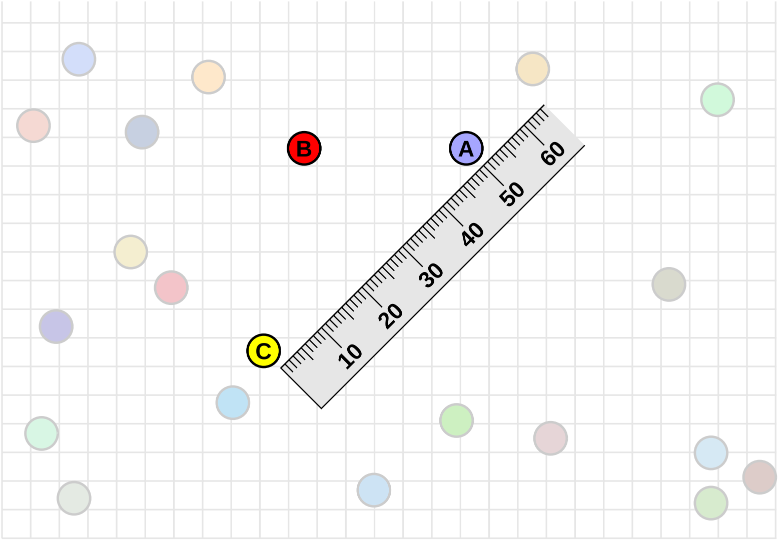
Vivaldi Example

A computes distance to C in coordinate space.



Vivaldi Example

A adjusts coordinates so distance matches actual RTT.
(Now A is wrong distance from B.)



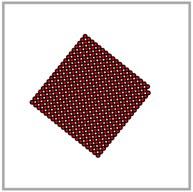
Challenges of Decentralization

Without centralized control, must consider:

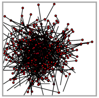
- will the system converge to an accurate coordinate set?
- how long will the system take to converge?
- will the system be disturbed by new nodes joining the system?

Tuning Vivaldi: Convergence

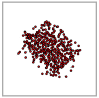
- Run Vivaldi on round trip times derived from grid.



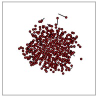
- As described, algorithm never converges.



- To cause convergence, damp motion.

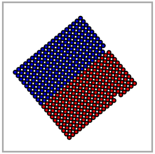


- To speed convergence, vary damping with estimate of prediction accuracy.

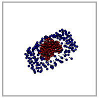


Tuning Vivaldi: Naive Newcomers

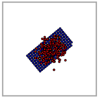
- Run Vivaldi on round trip times derived from grid.
  Blue nodes start first, stabilize.
  Red nodes join the system.



- High-accuracy nodes are displaced by new, low-accuracy nodes joining the system



- To avoid this, vary damping with ratio of local node's accuracy and sampled node's accuracy.



Vivaldi Algorithm

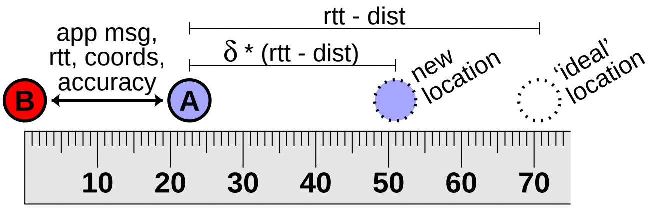
Given the coordinates, round trip time, and accuracy estimate of a node:

- Update local accuracy estimate.
- Compute ‘ideal’ location.
- Compute damping constant δ using local and remote accuracy estimates.
- Move δ of the way toward the “ideal” location.

Vivaldi Algorithm

Given the coordinates, round trip time, and accuracy estimate of a node:

- Update local accuracy estimate.
- Compute ‘ideal’ location.
- Compute damping constant δ using local and remote accuracy estimates.
- Move δ of the way toward the “ideal” location.



Evaluating Synthetic Coordinates on the Internet

- Cannot evaluate by comparing to “correct” coordinate set.
- Evaluate *predictions* made using a coordinate set.
- Predictions of Internet will never be perfect.
    - violations of triangle inequality, ...

Evaluating Vivaldi on the Internet

- How accurate are Vivaldi's predictions?
- How quickly does Vivaldi converge to a coordinate set?
- How quickly can Vivaldi adapt to network changes?
- How does choice of coordinate space affect error?
- How does Vivaldi work in real-world apps?

Evaluation Methodology

Use simulator seeded with real Internet measurements.

- pairwise RTTs for 192 PlanetLab nodes
- use RTT matrix as input to simulator
- run various algorithms on simulator

Each Vivaldi node queries others as fast as it can

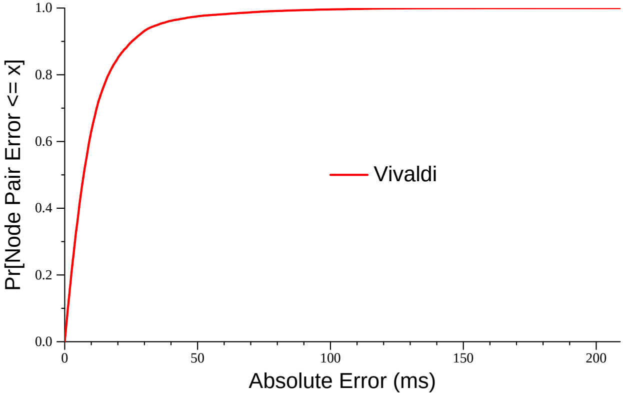
- one message outstanding at a time
- each node has a small fixed neighbor set

Vivaldi's Absolute Prediction Error

Look at distribution of absolute prediction error, defined as

$$|\text{actual RTT} - \text{predicted RTT}|,$$

over all node pairs in the system.

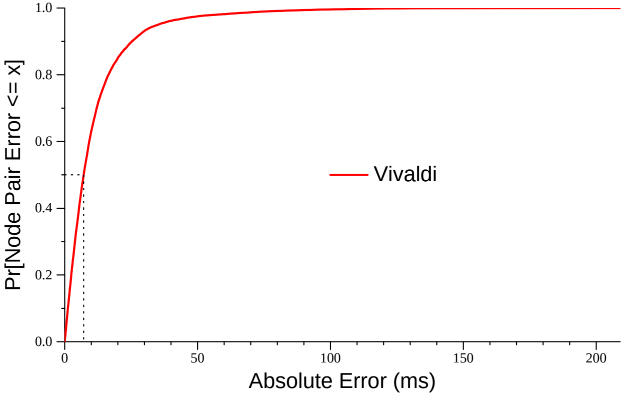


Vivaldi's Absolute Prediction Error

Look at distribution of absolute prediction error, defined as

$$|\text{actual RTT} - \text{predicted RTT}|,$$

over all node pairs in the system.

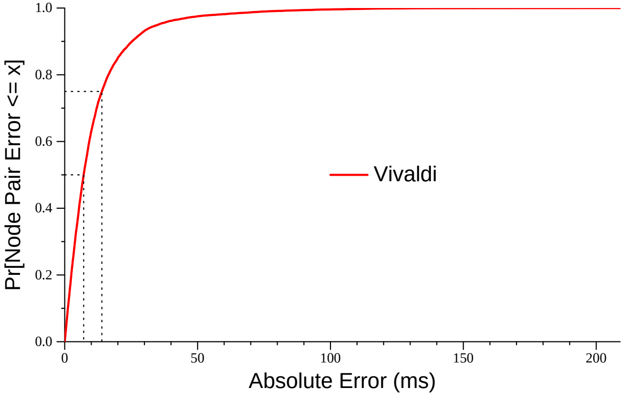


Vivaldi’s Absolute Prediction Error

Look at distribution of absolute prediction error, defined as

$$\left|\text{actual RTT} - \text{predicted RTT}\right|,$$

over all node pairs in the system.

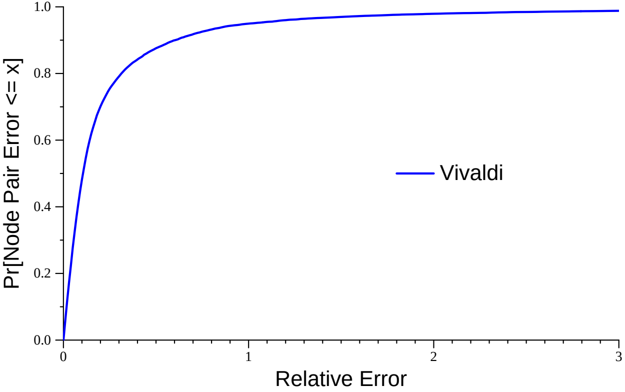


Vivaldi's Relative Prediction Error

Look at relative error, defined as

$$\frac{|\text{actual RTT} - \text{predicted RTT}|}{\min(\text{actual RTT}, \text{predicted RTT})},$$

over all node pairs in the system.

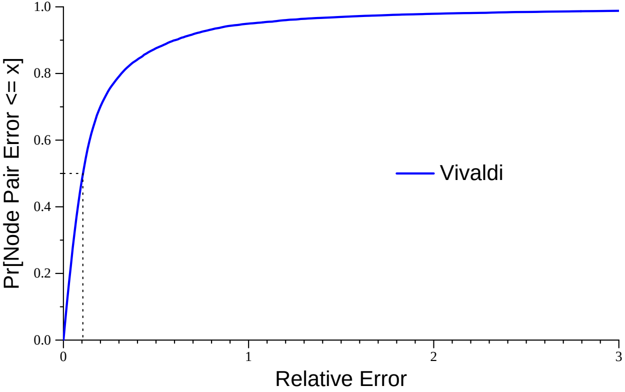


Vivaldi's Relative Prediction Error

Look at relative error, defined as

$$\frac{|\text{actual RTT} - \text{predicted RTT}|}{\min(\text{actual RTT}, \text{predicted RTT})},$$

over all node pairs in the system.

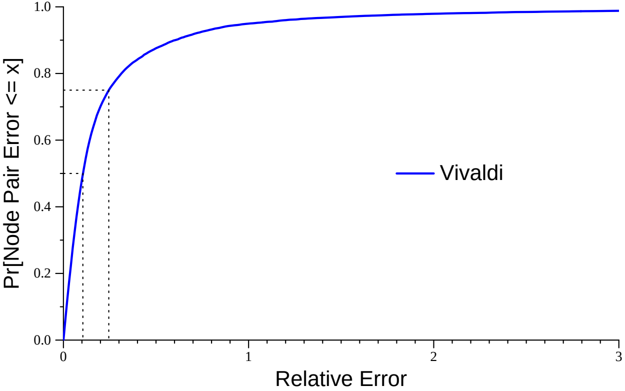


Vivaldi's Relative Prediction Error

Look at relative error, defined as

$$\frac{|\text{actual RTT} - \text{predicted RTT}|}{\min(\text{actual RTT}, \text{predicted RTT})},$$

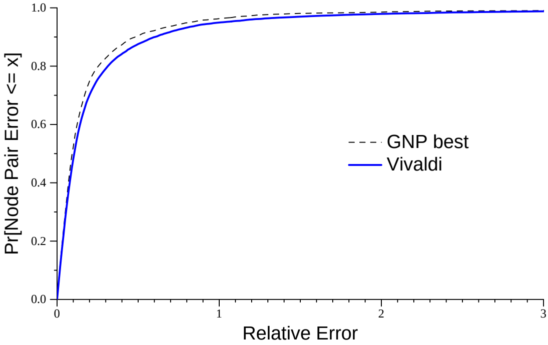
over all node pairs in the system.



Vivaldi Compared to GNP on Relative Error

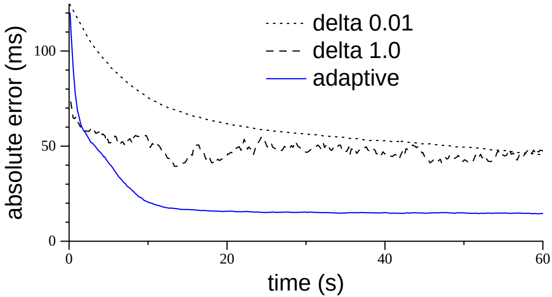
Compare to GNP's predictions.

- GNP sensitive to landmark choice. Use best of 64 random landmark sets.



Vivaldi's Convergence Time

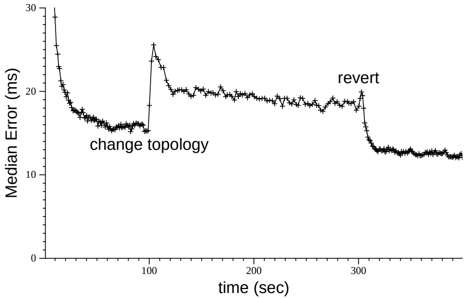
- Depends on choice of δ, the damping constant.



- Using adaptive δ, Vivaldi converges in under 20 seconds (60 measurements per node).

Vivaldi's Time to Adapt to Network Changes

- Vivaldi nodes are always adjusting their coordinates.
- Test adapting speed with synthetic topology change: lengthen one link by factor of ten.



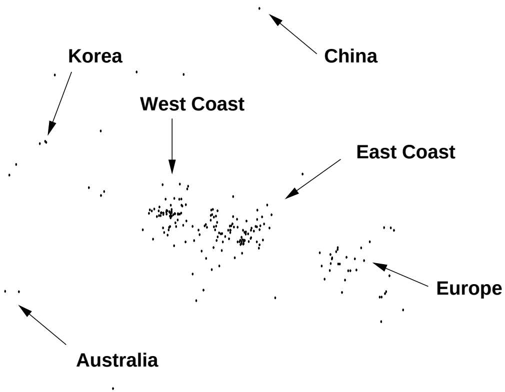
- Vivaldi adapts in about twenty seconds.

Other Coordinate Spaces

- *A priori*, it's not clear why *any* coordinate system should fit the Internet well.
- GNP showed that Euclidean coordinates work well.
- Why do they work?
- Are there better coordinate systems?
  - Obvious other candidates: globe, 3D, 4D, ...

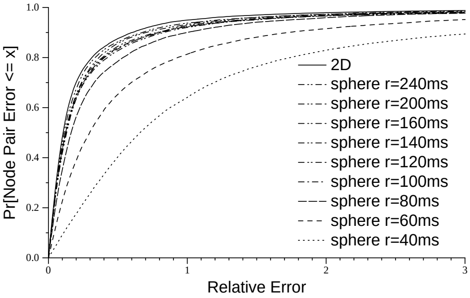
Vivaldi's 2D Assignment for PlanetLab

Placement in 2D mirrors physical geography.



Globe Coordinates vs. 2D Euclidean

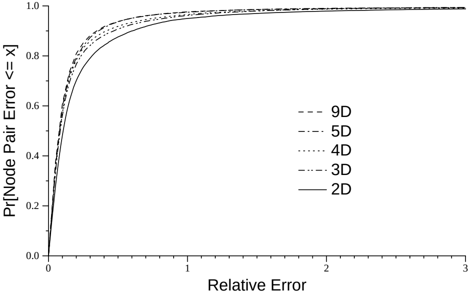
- Globe coordinates (latitude, longitude) place nodes on surface of a sphere.
- Great circle distance between two nodes on the sphere depends on radius.



- Coordinate sets are using one part of the sphere as a rough approximation to a 2D plane.

Higher Euclidean Dimensions

- If two are good, more should be better.



- Why are they better?

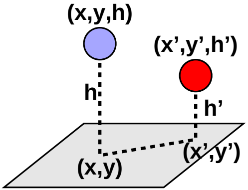
Higher Euclidean Dimensions Explained

- In 2D, some nodes need to be farther away from all others.
- In 3D, these “hard-to-place” nodes can move up or down from the 2D plane to get away from everyone.
- Each new dimension adds an independent direction.
- Accomodates per-node overhead: server load, access links.

Problem: how can we accomodate “hard-to-place” nodes without an arbitrary number of dimensions?

Height Vectors

- Give “hard-to-place” nodes their own way to get away from everyone.
- Height vectors place nodes at some height above a 2D transit plane.
- Directly models per-node overhead.

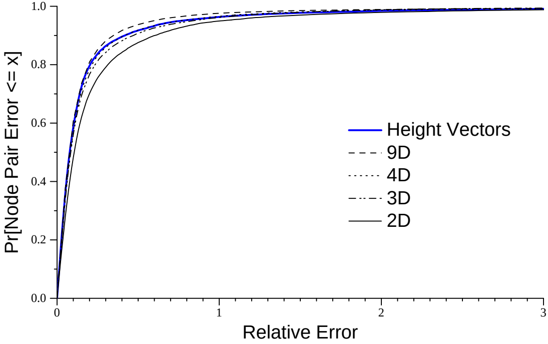


- Distance from (x, y, h) to (x', y', h') is

$$h + \sqrt{(x - x')^2 + (y - y')^2} + h'.$$

Height Vectors Work Well

- Height Vectors outperform 2- and 3-D Euclidean.



- Works to view Internet as geographically-accurate core with access links attached.

Real-World Application

- Vivaldi is easy to deploy, because it:
  - requires no infrastructure
  - is simple to implement
  - piggybacks on application-level communication
- We modified DHash to use Vivaldi.
  - block fetch time on PlanetLab reduced by 40% (NSDI 2004).
- Vivaldi is also used by:
  - Bamboo Distributed Hash Table
  - SWORD Resource Discovery system

Related Work

- Other location techniques (IDMaps, IP2Geo) use static data (AS maps, guesses at physical location).
- Centralized coordinate systems (GNP, Lighthouse) need well-known landmark nodes.
- Decentralized coordinate systems (PIC, NPS) have been developed concurrently.
- Tang and Crovella (IMC 2003) analyze best Euclidean models to use; Shavitt and Tankel (Infocom 2004) suggest using hyperbolic geometries.

Conclusions

- Vivaldi:
  - accurately predicts round trip time
    between node pairs not directly measured.
  - works without centralized infrastructure.
  - improves the performance of a real system.
- Height vectors are a promising coordinate space for the Internet.