SWINBURNE UNIVERSITY OF TECHNOLOGY

Estimating network latency before directly exchanging packets

(A survey of some published ideas)

Grenville Armitage

Centre for Advanced Internet Architectures

Overview

- Why this talk?
- Why is it important to know round trip time (RTT) ?
- Active and passive measurements
- Indirect estimation via measurement
- Indirect estimation via network coordinates
- Conclusion

Why? A partial survey of literature

- Stimulate awareness and discussion
- Perhaps even trigger some new research at CAIA?
- Not an exhaustive review – just a few highlights
  - General descriptions of principle ideas – check referenced links for details and quantitative analyses of each scheme

What is the importance of RTT ?

- Network latency (round trip time, RTT) impacts
  - End to end TCP performance (higher RTT, lower *goodput*)
  - Response times for query/response protocols
  - Interactive performance for VoIP,virtual environments, ...
- If I know the RTT to other places, I can:
  - Optimise my selection of remote or peer content servers
    - Closer servers for quick query/downloads, or more distant servers for fault-tolerant replication of data
  - Map non-linearities in Internet's internal structure
  - Identify closer servers for games
    - ..and so on, and so forth!

(Sidebar: The server selection challenge)

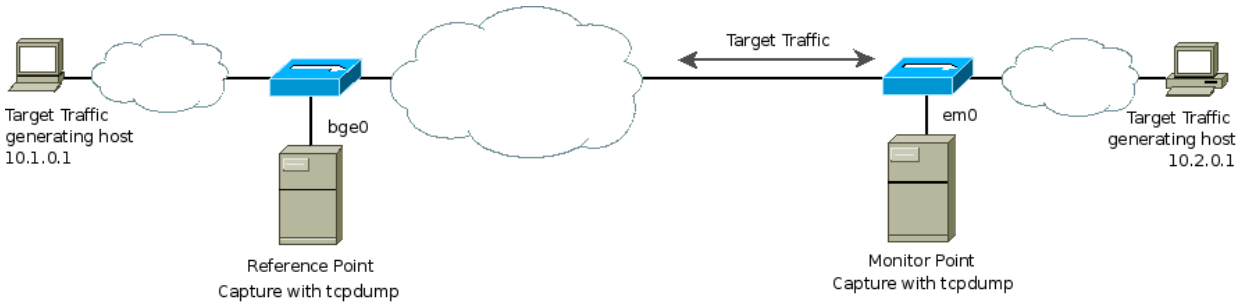
- Replicated storage/delivery has broad application
  - Web content, Peer to peer / Distributed Hash Tables (DHT),
  - Distributed DNS (domain name system),
  - Network Time Protocol (NTP), etc...
- Ideally we direct client queries to the 'closest' node serving the client's desired information/data
  - E.g. active redirection based on client's source IP address, AS number, country code...
  - Or redirect by modifying DNS replies based on client's source IP...
  - OR: Use knowledge (belief?) about the typical RTT from a client to the pool of available servers to identify the 'optimal' server
    - We explore this latter case today

RTT – sum of many components

- Network topology & geography
  - Routing topology → influences geographic distances
  - Geographic distances → speed of light delays
- Serialisation
  - Finite time to send individual packets at 'line rate'
- Queuing
  - The Internet's admission control is benign neglect – statistical multiplexing + queues to cover transient congestion (per hop)
- Relative contributions:
  - Topology & serialisation delays change over long periods, queuing delays change on per-packet time periods

Direct measurements (active & passive)

- Actively measure RTT from an endpoint's perspective
  - Query/response probing: ping, traceroute, App-specific...
  - Inject probe traffic into the network, then...
    - Accurately measure departure and arrival times at diverse monitoring points (e.g. tcpdump, “Endace” DAG cards,...)
- Passively infer RTT using existing traffic
  - E.g. CAIA's SPP toolkit utilises pre-existing (even unrelated) bi-directional traffic observed at diverse monitoring points



http://caia.swin.edu.au/tools/spp/

An issue with direct measurement

- Measurement traffic adds load to the network
  - Consider IP_x probing every member of the set of endpoints $\{IP_{y1}, IP_{y2}, ...\}$ before selecting one to use
  - ...or everyone in $\{IP_{y1}, IP_{y2}, ...\}$ probing each other
- Probing load may outweigh the subsequent benefits
  - Consider probing 100s of endpoints before initiating a short TCP connection...
    - might be better just to randomly pick an endpoint ?

Problem statement

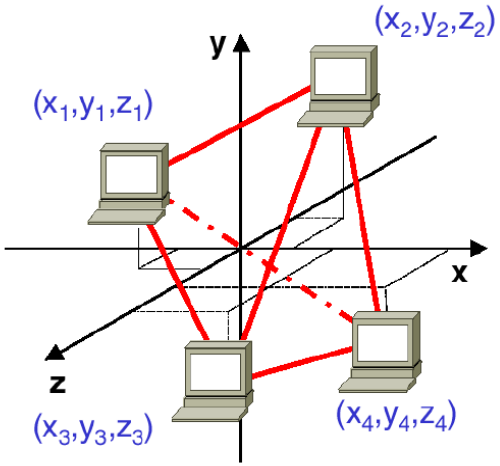
- Assume we're locating services, peers, etc
  - by latency constraint, or ranked by latency, ...
- Typically answering variations of:
  - Estimate the *probable* latency between two IP addresses on the Internet without additional exchange of packets between them!
    - e.g. Estimate the probable latency between IP_x and a known set of endpoints {IP_{y1} , IP_{y2} , ...} without additional probing traffic
  - Find a 'closest' node without additional probing traffic
  - Find rank ordering of other nodes, ranked by RTT

Estimation of network delay

- Indirect measurement
  - Select groups of nodes regularly probe each other
  - We infer RTT between IP_x and IP_y based on the RTT between nodes near IP_x and IP_y (*triangle inequality*)
- Network coordinates
  - Hosts virtually embedded in a coordinate space
  - Assign coordinates to hosts such that distance approximates RTT. E.g. for two points (x1,y1,z1) (x2,y2,z2) in a Euclidean 3D space:

$$RTT \propto \sqrt{(x1-x2)^2+(y1-y2)^2+(z1-z2)^2}$$

  - (Chicken-and-egg -- but communication patterns of some apps allow indirect distribution of coordinates)



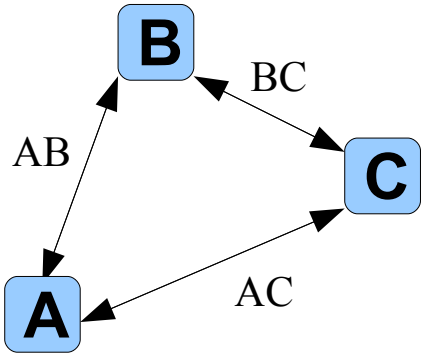
(Fig1. http://dx.doi.org/10.1109/INFCOM.2002.1019258)

Indirect measurement schemes

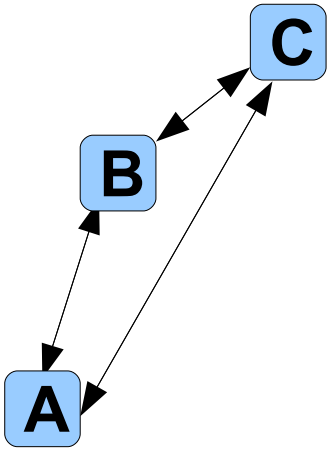
- The assumption of Triangle Inequality
- IDMaps - 2001
  - P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, “Idmaps: a global internet host distance estimation service,” IEEE/ACM Transactions on Networking, vol. 9, no. 5, pp.525--540, Oct 2001
- King – 2002
  - K. P. Gummadi, S. Saroiu, and S. D. Gribble, “King: estimating latency between arbitrary internet end hosts,” in IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement New York, NY, USA: ACM, 2002, pp. 5--18
- Meridian – 2005 ('indirect' in a slightly different way)
  - B. Wong, A. Slivkins, and E. G. Sirer, “Meridian: a lightweight network location service without virtual coordinates,” SIGCOMM '05, New York, NY, USA: ACM, 2005,pp. 85--96

The Triangle Inequality

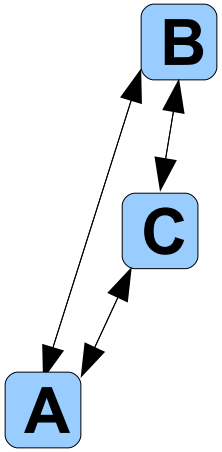
- Consider triangle A, B and C
  - If XY is scalar *distance* between X and Y
  - Then $|AB-BC| \leqslant AC \leqslant AB+BC$
    - (assuming we know AB and BC)



$AC \leqslant AB+BC$



$|AB-BC| \leqslant AC$

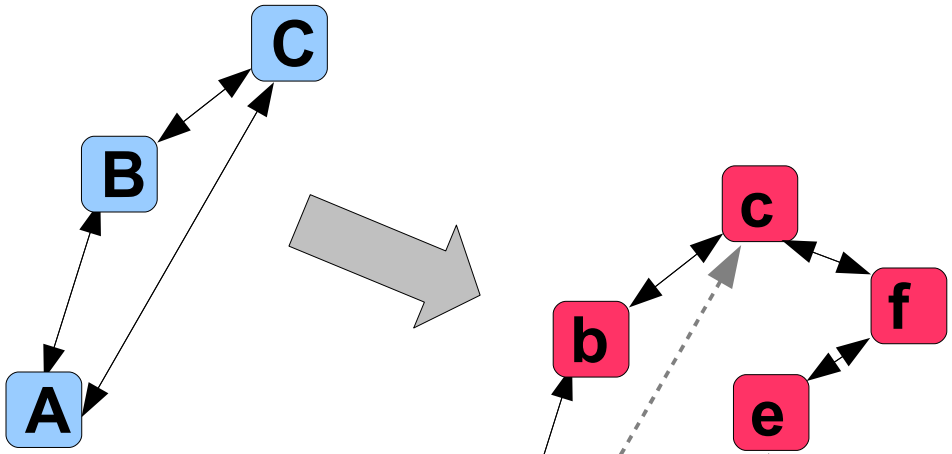


Thus: measured distances for AB and BC provides bounds on (and an estimate of) distance AC

Note: If AB or BC is small, the estimate of AC is fairly accurate

Breaking the Triangle Inequality

- "Distance == RTT" assumes *efficient* routing
  - e.g. Internet finds low-latency paths, routes to nearby hosts will not be drastically different from each other...
- These assumptions may be violated in real networks



IP topology

L2 topology

If routing or policy chooses a convoluted Layer 2 path for A to C, it is possible that:

$$AC \geqslant AB + BC$$

(May also occur that AC is a low speed or physically longer path)

Yet the triangle inequality assumption is *generally* considered to hold....

IDMaps - 2001

- Early proposal based on active probing by *Tracers* scattered around the Internet [1]

- Internet is divided into Address Prefixes (APs) – ranges of IP addresses within which all hosts are roughly equidistant to the rest of the Internet.
- Tracers share the RTT results of regularly probing each other and APs
- These raw RTT measurements are called 'virtual links'
- Tracers need to be located near APs

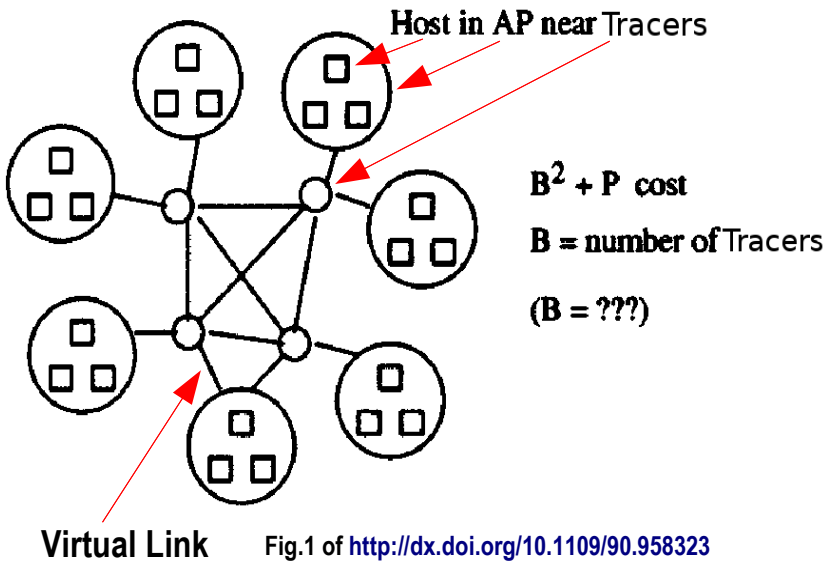
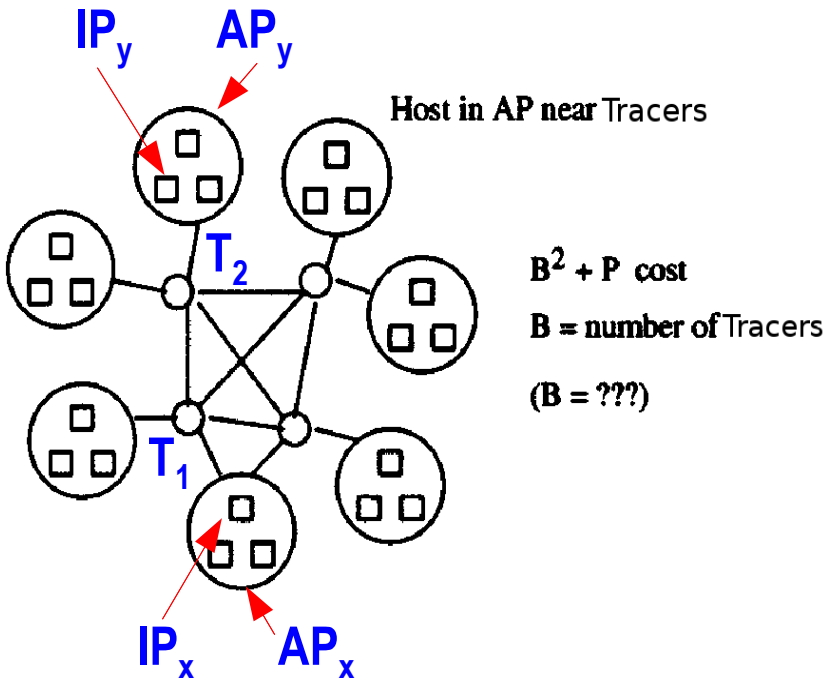


Fig.1 of http://dx.doi.org/10.1109/90.958323

[1] Francis, P., et al, " IDMaps: a global Internet host distance estimation service", http://dx.doi.org/10.1109/90.958323

IDMaps (cont'd)

- Given arbitrary endpoints IP_x and IP_y belonging to AP_x and AP_y respectively:



- Assume IP_x wants the RTT to IP_y
- $IP_x \rightarrow IP_y$ is approximated by the sum

  $(AP_x \rightarrow T_1) + (T_1 \rightarrow T_2) + (T_2 \rightarrow AP_y)$
- Knowing Tracer topology allows shortest path computation when there are multiple 'virtual link' paths between T_1 and T_2
- Individual endpoints actually query via IDMaps Clients (iCs), which then answer using information regularly advertised by Tracers

IDMaps – reducing Tracer probing

- If we have multiple Tracers in different cities, and some are quite close to each other...
- Rather than a matrix of virtual links between all Tracers, assume triangle inequality holds and estimate some links from a subset of probes (bold links)

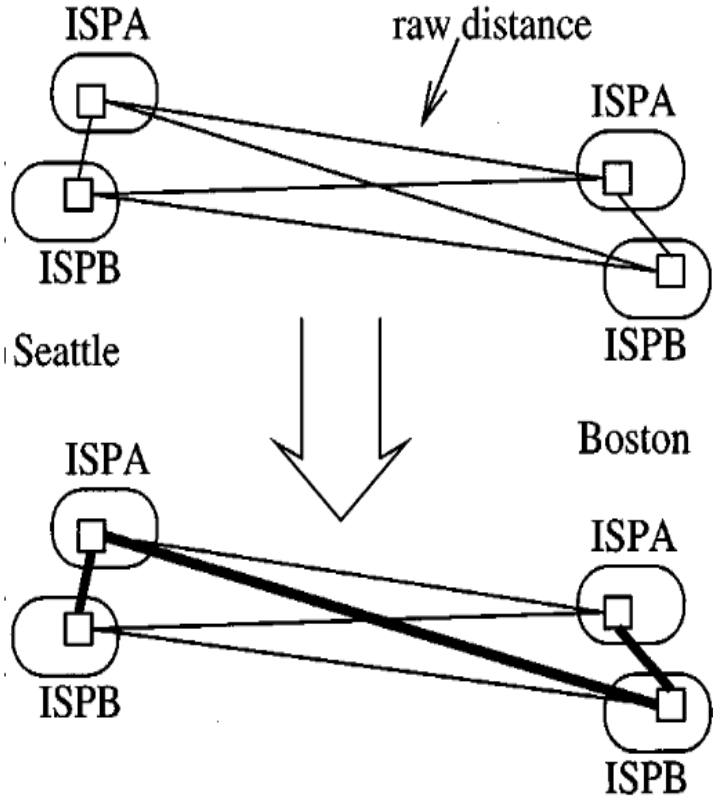


Fig.6 of http://dx.doi.org/10.1109/90.958323

King - 2002

- King uses DNS servers as a 'free' infrastructure for inferring the RTTs between regions of the Internet [2]

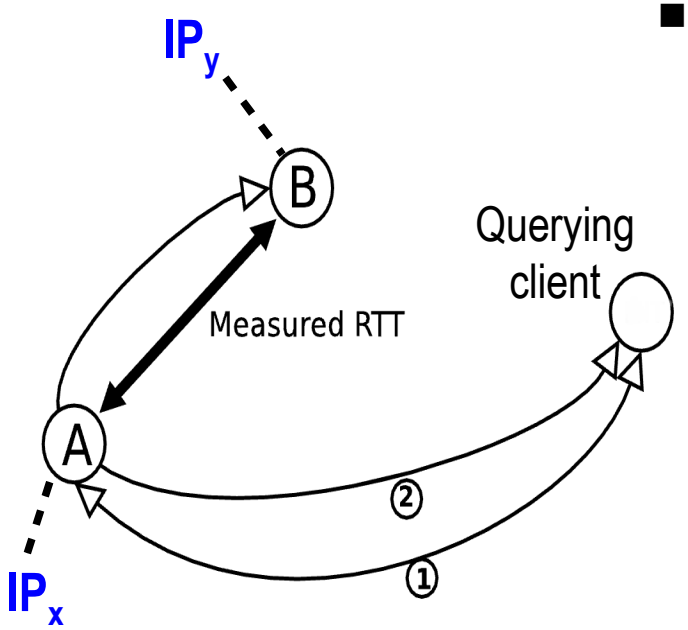


Fig.4 of http://doi.acm.org/10.1145/1015467.1015471

- Assume IP_x wants the RTT to IP_y, IP_x is near (and served by) DNS server A, and IP_y is near (and served by) DNS server B
  - (1) Time a query to A for a domain name served by A
  - (2) Time a query to A for a domain name served by B (forcing a recursive query)
  - Time difference between (2) and (1) is roughly the RTT from IP_x to IP_y
    - (repeat 3 times, client exchanges 6 packets with server A)

[2] Gummadi, K., et al, "King: estimating latency between arbitrary internet end hosts", http://doi.acm.org/10.1145/637201.637203

King (cont'd)

- Key assumptions:

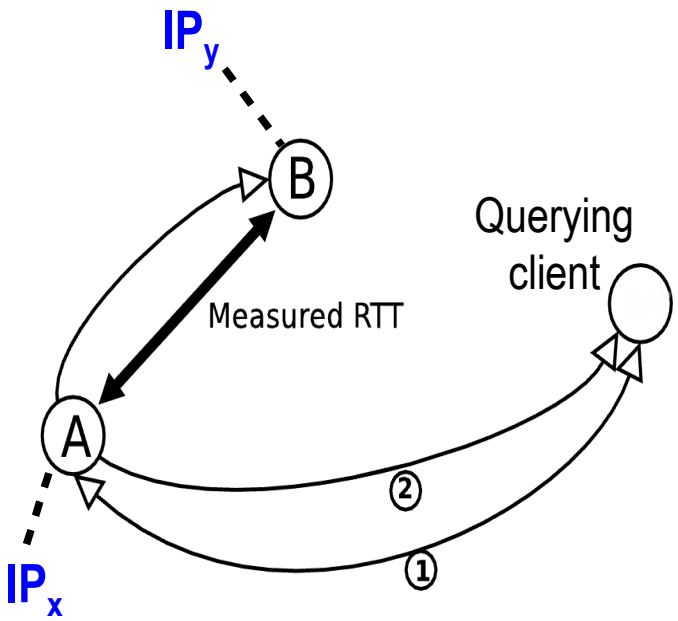


Fig.4 of http://doi.acm.org/10.1145/1015467.1015471

- Authoritative DNS servers are 'close' to the hosts of interest
- Query load for RTT estimation doesn't annoy DNS server operators
- Server A will perform recursive queries, and B will answer them
- Multiple B to chose from?

See http://doi.acm.org/10.1145/637201.637203 for answers....

Meridian - 2005

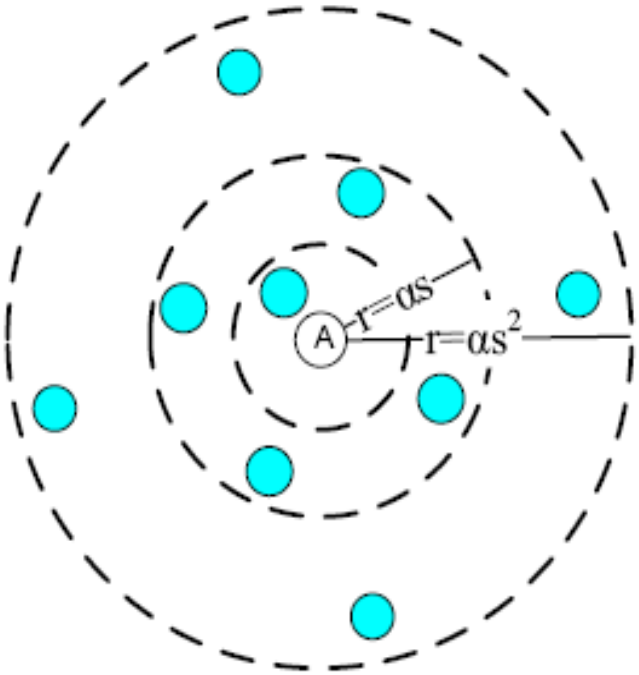
- Meridian is
  - "...for performing node selection based on network location." [3]
- Solves:
  - "closest node discovery"
  - "central leader election"
  - "locating nodes that satisfy target latency constraints"
- "Framework consists of:
  - An overlay network structured around multi-resolution rings
  - Query routing with direct measurements, and
  - Gossip protocols for dissemination."

(Meridian aims to solve the problems for which we might otherwise need lots of RTT measurements in large systems)

[3] Wong, B., et al, "Meridian: A lightweight network location service without virtual coordinates", http://portal.acm.org/citation.cfm?id=1080091.1080103

Meridian (cont'd)

- 'Meridian nodes' track smalls sets of peers organised into concentric rings of exponentially increasing radii
  - Radii is directly measured RTT
  - Nodes gossip about peers to other nodes
  - Ring membership is updated based on knowledge learned via gossiping
- Nodes aim for 'diversity' amongst the peers they keep in their rings
  - More closer peers, less distant peers



(Fig.1 of http://portal.acm.org/citation.cfm?id=1080091.1080103)

Meridian – closest node discovery

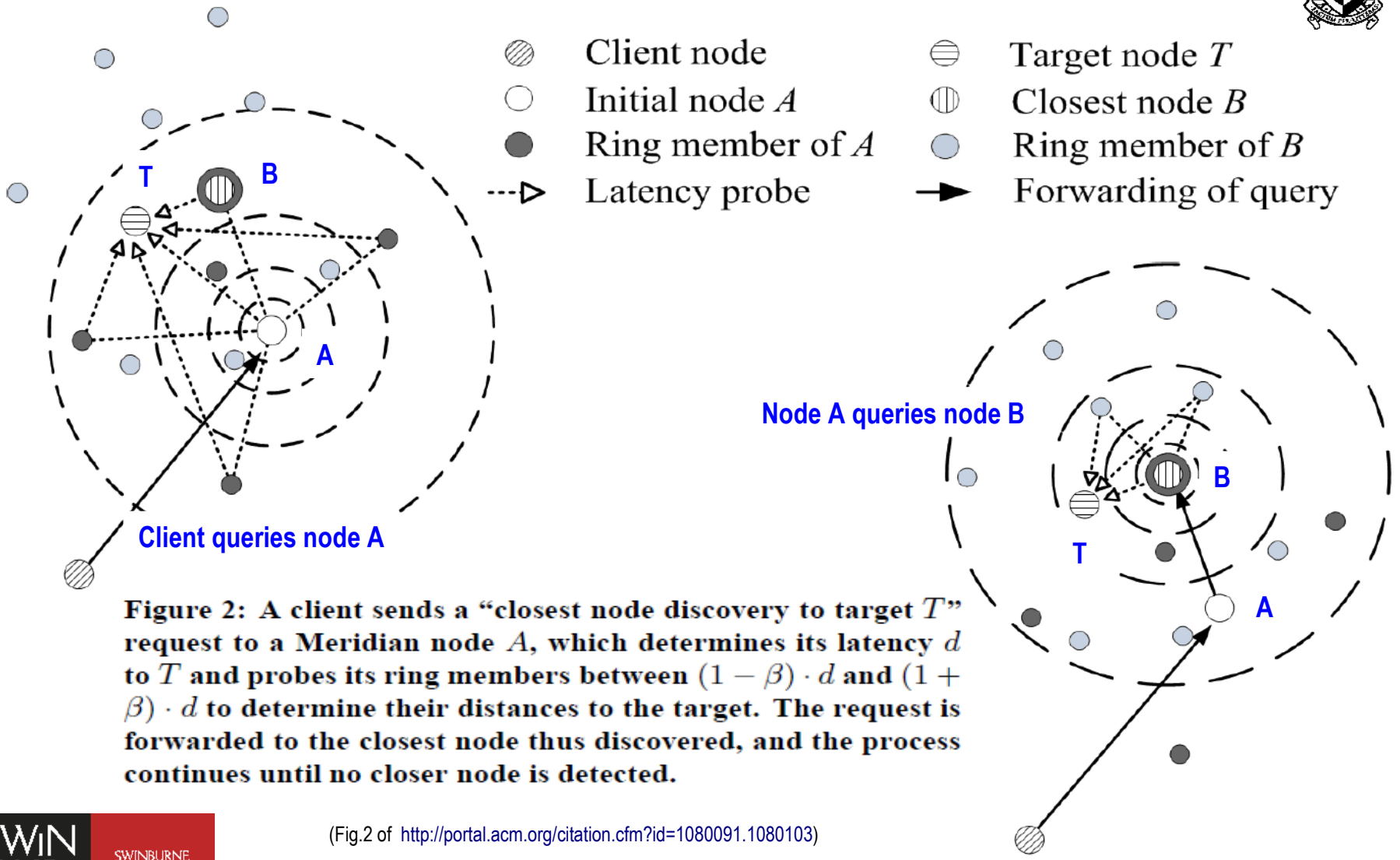
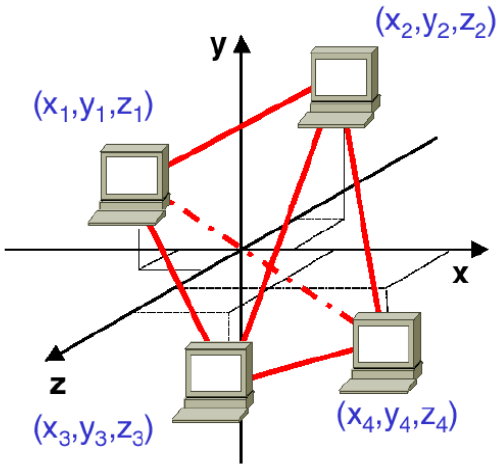


Figure 2: A client sends a "closest node discovery to target T" request to a Meridian node A, which determines its latency d to T and probes its ring members between $(1-\beta)\cdot d$ and $(1+\beta)\cdot d$ to determine their distances to the target. The request is forwarded to the closest node thus discovered, and the process continues until no closer node is detected.

(Fig.2 of http://portal.acm.org/citation.cfm?id=1080091.1080103)

Network coordinate schemes

- Embedding in a coordinate space
- Landmarks
  - GNP – 2001
    - T. S. E. Ng and H. Zhang, “Towards global network positioning,” in IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. New York, NY, USA: ACM, 2001, pp. 25--29.
- Simulations
  - Vivaldi – 2004
    - F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: a decentralized network coordinate system,” SIGCOMM '04:, New York, NY, USA: ACM, 2004, pp. 15--26.



(Fig1. http://dx.doi.org/10.1109/INFCOM.2002.1019258)

Embedding in a coordinate space

- Geometric coordinates for hosts allows simplifications:
  - Distance function → straightforward & fast calculation
  - Intrinsic structure of coordinate space can simplify the solution of problems like 'nearest neighbour search'
  - If you have mesh of H hosts, the data shared scales as
    - O(H*C) for coordinate space with dimensionality C, versus
    - $O(H^2)$ to share a matrix of H*(H-1)/2 RTT measurements
- Euclidean? Spherical?
  - Different geometric spaces may be modelled. Euclidean with C dimensions is easy to visualise.
    - More than 2 dimensions usually required to model non-optimal realities of Internet routing

Establishing & distributing coordinates

- Establishment
  - Assume a set of H potential peers
  - Each host's coordinates are established through direct communication with a subset (< H) of the other peers
    - Subset contains at least C+1 other peers, assuming C dimensions
- Distribution
  - For peer to peer apps, coordinates may be shared by piggy-backing onto existing peer-discovery mechanism
    - A free ride!
  - In general, share coordinates via same mechanism used to share/distribute IP addresses

GNP (Global Network Positioning) - 2001

- First proposal to use coordinates [4][5]
- Two tier system:
  - Set of Landmarks probe each other & calculate coordinates
  - Clients probe subset of Landmarks, and establish their coordinates relative to Landmark coordinates

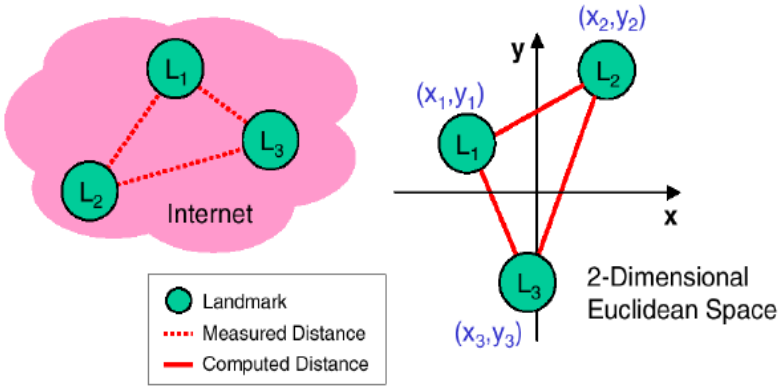


Fig. 2. Part 1: Landmark operations
(http://dx.doi.org/10.1109/INFCOM.2002.1019258)

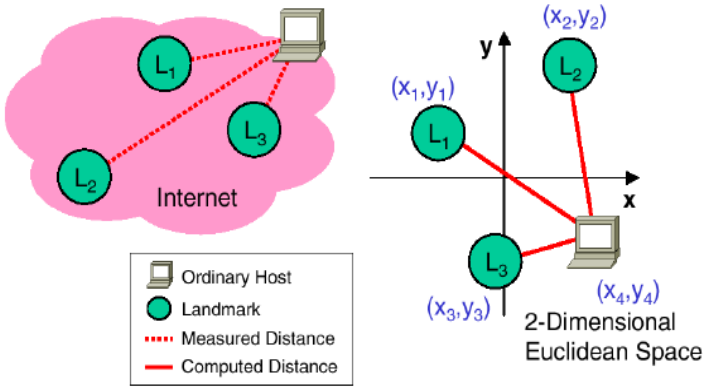


Fig. 3. Part 2: Ordinary host operations
(http://dx.doi.org/10.1109/INFCOM.2002.1019258)

[4] Ng, Zhang, "Towards global network positioning" http://doi.acm.org/10.1145/505202.505206

[5] http://dx.doi.org/10.1109/INFCOM.2002.1019258

GNP (continued)

- Landmarks calculate coordinates offline
  - "...*the computation of the coordinates can be cast as a generic multi-dimensional global minimization problem that can be approximately solved by many available methods such as the Simplex Downhill method.*"
  - i.e. find the set of coordinates that minimises the difference between measured RTTs and 'distance' derived from coordinates
  - Coordinates are then distributed among Landmarks, and to clients who wish to use Landmarks
- Location of Landmarks impacts accuracy of GNP
  - Landmarks may also be a performance bottleneck

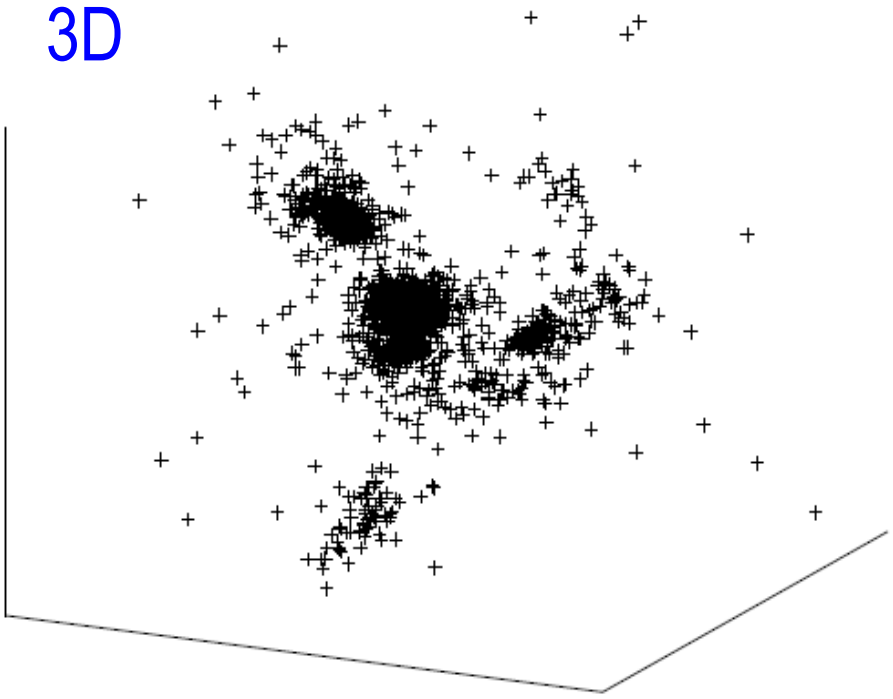
See also Lim, Hou, Choi, " Constructing internet coordinate system based on delay measurement", 2003, http://doi.acm.org/10.1145/948205.948222

Vivaldi - 2004

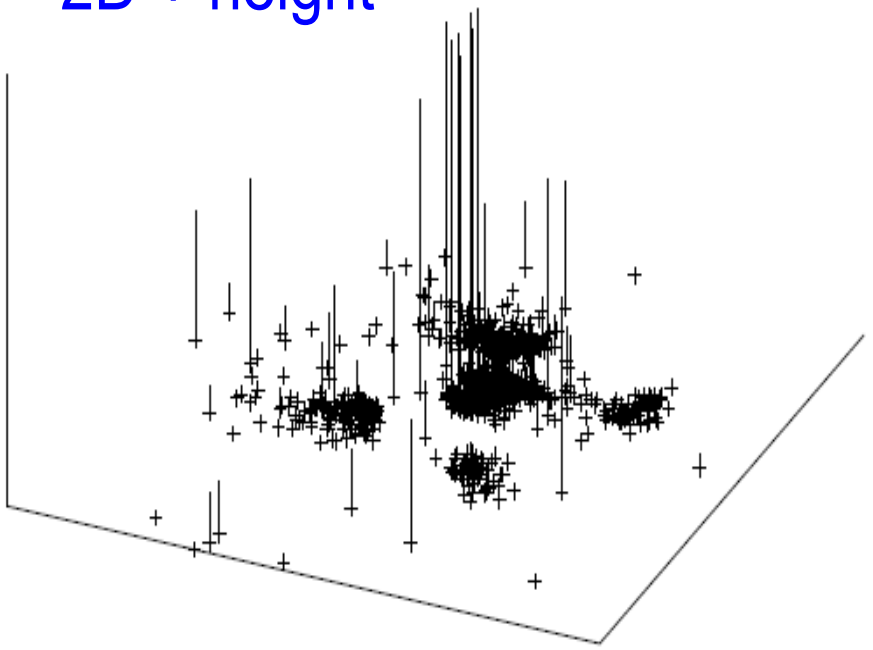
- Incremental, distributed calculation of coordinates [6]
  - RTT estimates are constructed as peers talk to each other
  - Simulated *mass-spring* model used to nudge host coordinates towards low-energy equilibrium positions
  - As new RTT estimates occur between peers, coordinates are refined and shared with neighbouring peers
- Unique use of 2D + "height" coordinates
  - Modelling backbones as 2D space + "height" representing RTT contributed by access links
    - 'Distance' is euclidean 2D distance + each node's height

[6] Dabek, F, et al, "Vivaldi: A Decentralized Network Coordinate System"
http://doi.acm.org/10.1145/1015467.1015471

Vivaldi (cont'd)



(Fig.16(b) of http://doi.acm.org/10.1145/1015467.1015471)



(Fig.16(d) of http://doi.acm.org/10.1145/1015467.1015471)

Vivaldi (cont'd)

- Vivaldi positioning 115 US Planetlab nodes

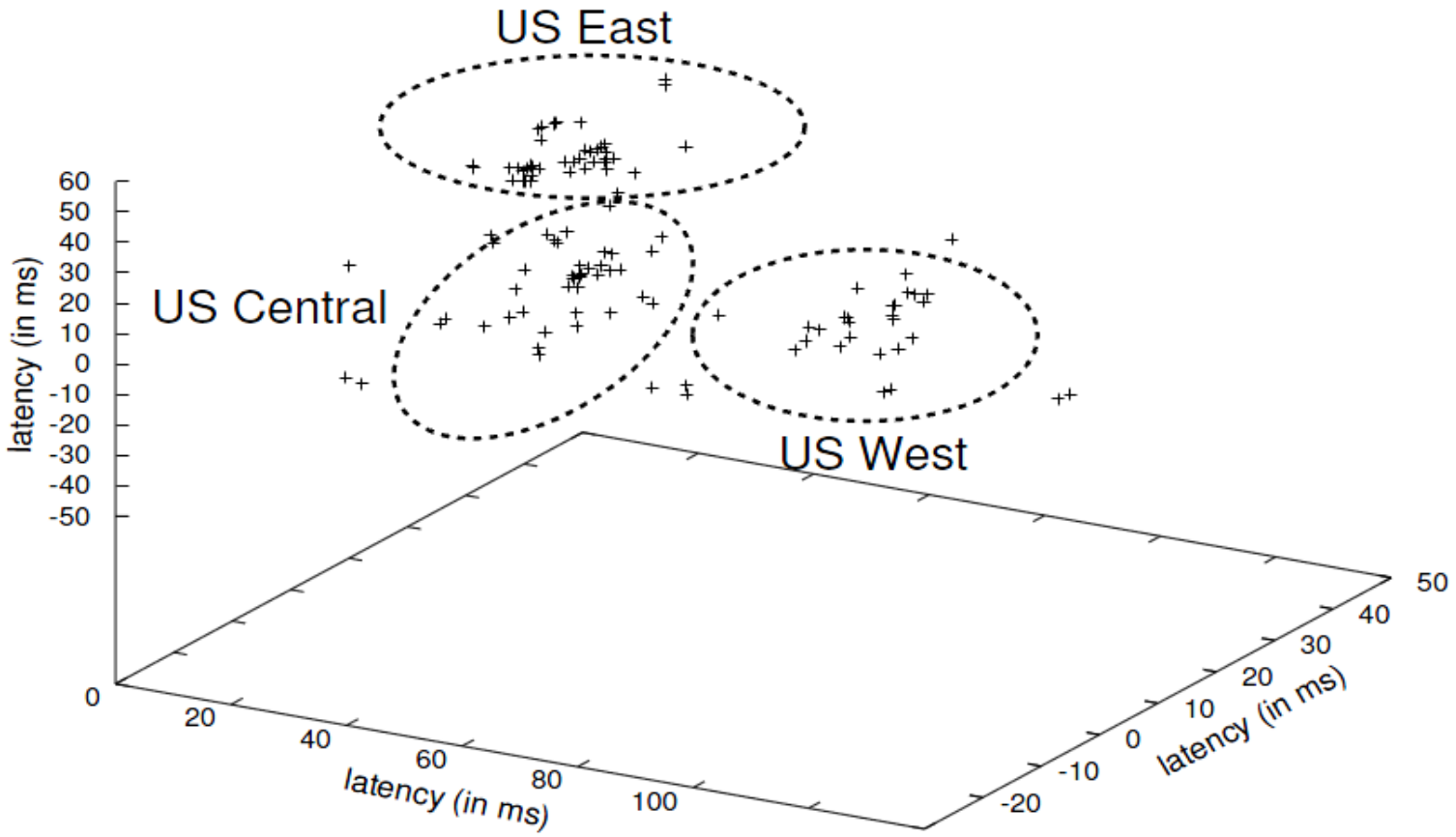


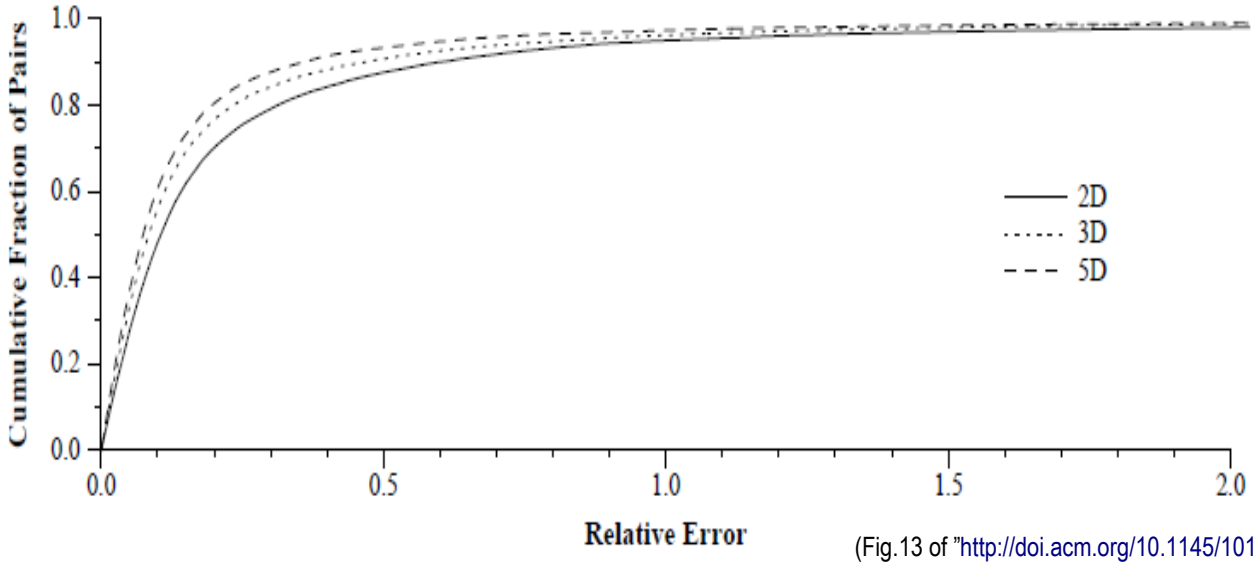
Fig.2 of http://www.eecs.harvard.edu/~michaelm/postscripts/iwdds06.pdf

Vivaldi and p2p software

- Nice match to peer-to-peer environments
  - No Landmarks or other special hosts required
- E.g. Vivaldi is used by Azureus (bit torrent) p2p software
  - http://azureuswiki.com/index.php/Vivaldi_View
  - http://www.eurecom.fr/~btroup/vivaldiazureus/

How many dimensions is enough?

- Different opinions (of course!)
- More dimensions allows embedding to mould around oddities in Internet routing (policies, L2 paths, etc)
  - Diminising returns.... increased computational load...
  - E.g. Vivaldi with 2D, 3D and 5D euclidean coordinates



(Fig.13 of "http://doi.acm.org/10.1145/1015467.1015471)

Challenges/limitations

- Does a given scheme work better at predicting Rank order versus absolute RTT ?
- RTT always changing
  - Coordinates become 'stale' → tolerable embedding errors?
  - Frequency & compute load of updating coordinates ?
- How often are assumptions violated
  - e.g. triangle inequality, positions of landmarks, tracers, ...

Conclusions

- Identifying host/node 'positions' on the Internet can be valuable in a number of scenarios
- Direct RTT measurements can be prohibitive when solving “pick best X among set of Y nodes” (and similar) questions
- Estimation techniques have emerged:
  - Indirect measurement (+assumptions)
  - Virtual coordinate systems
  - Overlays to directly solve node discovery questions
- No silver bullet – tradeoffs everywhere

Some additional reading

- 2003: "Constructing internet coordinate system based on delay measurement"
  - http://doi.acm.org/10.1145/948205.948222
- 2004: "Triangulation and Embedding Using Small Sets of Beacons"
  - http://dx.doi.org/10.1109/FOCS.2004.70
- 2005: "On the accuracy of embeddings for internet coordinate systems"
  - http://portal.acm.org/citation.cfm?id=1251097&dl=GUIDE&coll=GUIDE&CFID=36757426&CFTOKEN=76760405
- 2008: "On the internet delay space dimensionality"
  - http://doi.acm.org/10.1145/1452520.1452541
- 2008: "Practical large-scale latency estimation"
  - http://dx.doi.org/10.1016/j.comnet.2007.11.022