# Locating Internet Hosts

Venkat Padmanabhan

Microsoft Research

Harvard CS Colloquium

20 June 2001

# Outline

- Why is user or host positioning interesting?
- Solving the problem in two different domains
  - RADAR: wireless LAN environment
  - IP2Geo: wide-area Internet environment
- Geography as a tool for studying the Internet
- Summary

# Motivation

- Location-aware services help users interact better with their environment
  - Navigational services (in-building, metro area)
  - Resource location (nearest restaurant, nearest printer)
  - Targeted advertising (sales, election canvassing)
  - Notification services (buddy alert, weather alert)
- User positioning is a prerequisite to location-aware services
- But this is a challenging problem

# Our Work

- We have built host location systems for two different environments
  - RADAR: wireless LANs
    - mobile clients (laptops, PDAs) that connect via a wireless LAN
    - typically within buildings
  - IP2Geo: wide-area Internet
    - typically fixed hosts (e.g., desktop machines, home PCs)
- Goal: leverage existing infrastructure

# RADAR

(Joint work with P. Bahl and A. Balachandran)

# Background

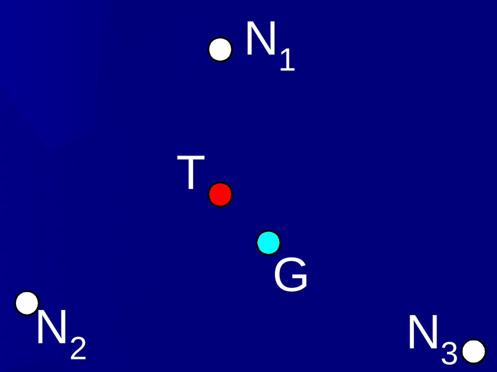
- Focuses on the indoor environment
- Limitations of current solutions
  - global positioning system (GPS) does not work indoors
  - line-of-sight operation (e.g., IR-based Active Badge)
  - dedicated technology (e.g., ultrasound-based Bats)
- Our goal: leverage *existing* infrastructure
  - use off-the-shelf RF-based wireless LAN
  - intelligence in software
  - better scalability and lower cost than dedicated technology

# RADAR Basics

- Key idea: signal strength matching
- Offline calibration:
  - tabulate <location,SS> to construct *radio map*
  - empirical method or mathematical method
- Real-time location and tracking:
  - extract SS from base station beacons
  - find table entry that best matches the measured SS
- Benefits:
  - little additional cost
  - no line-of-sight restriction $\Rightarrow$ better scaling
  - autonomous operation $\Rightarrow$ user privacy maintained

# Determining Location

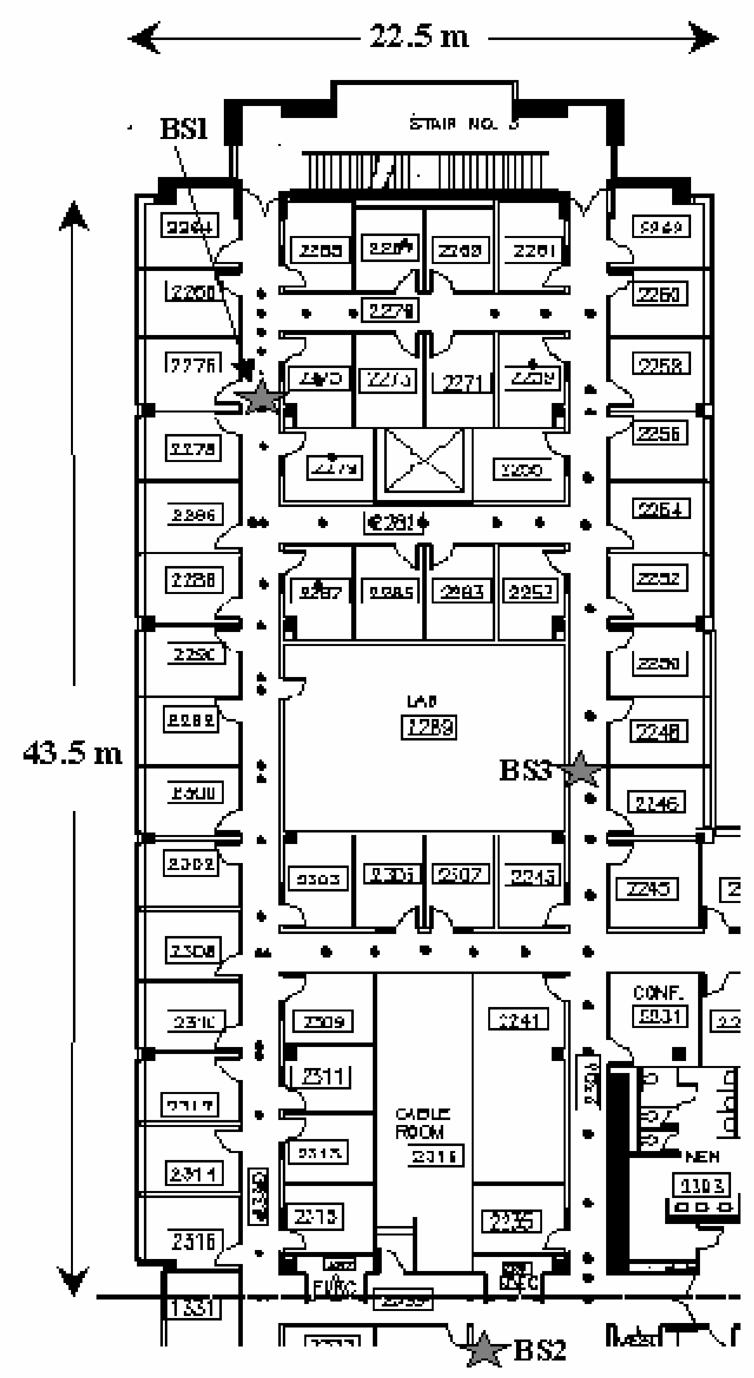
- Find *nearest neighbor in signal space* (NNSS)
  - default metric is Euclidean distance
- Physical coordinates of NNSS ⇒ user location
- Refinement: *k*-NNSS
  - average the coordinates of *k* nearest neighbors



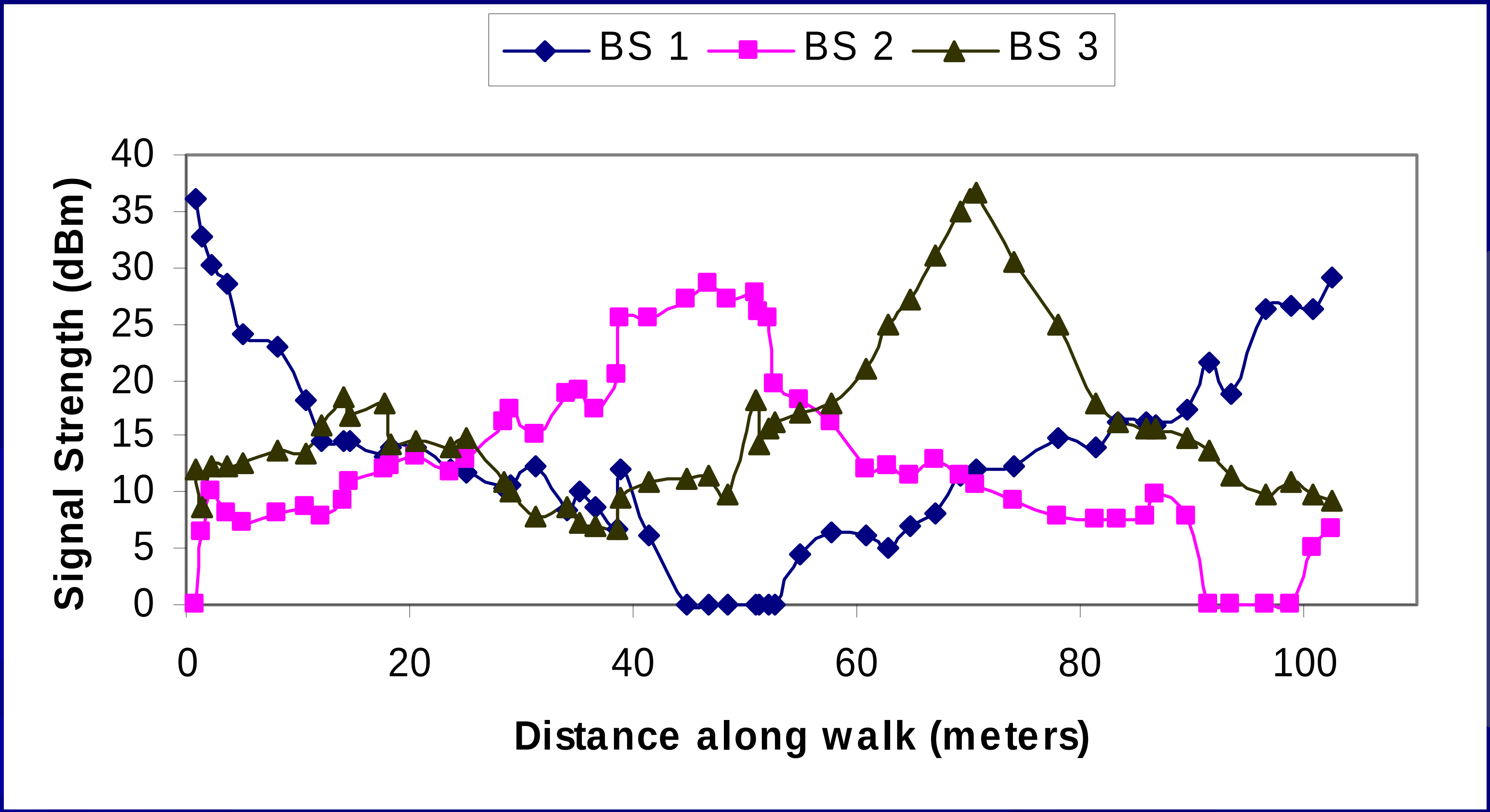

$N_1$, $N_2$, $N_3$: neighbors
T: true location of user
G: guess based on averaging

# Experimental Setting

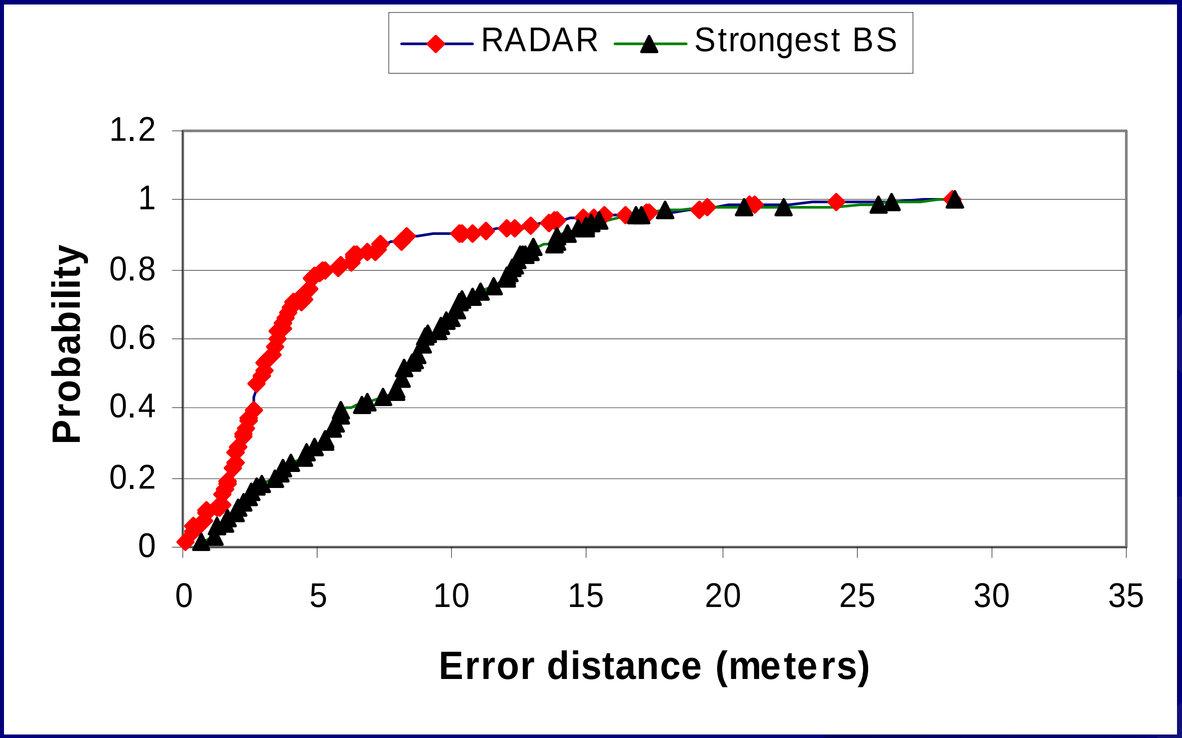
- Digital RoamAbout (WaveLAN)
- 2.4 GHz ISM band
- 2 Mbps data rate
- 3 base stations
- 70x4 = 280 (x,y,d) tuples



# How well does signal strength correlate with location?



# RADAR Performance



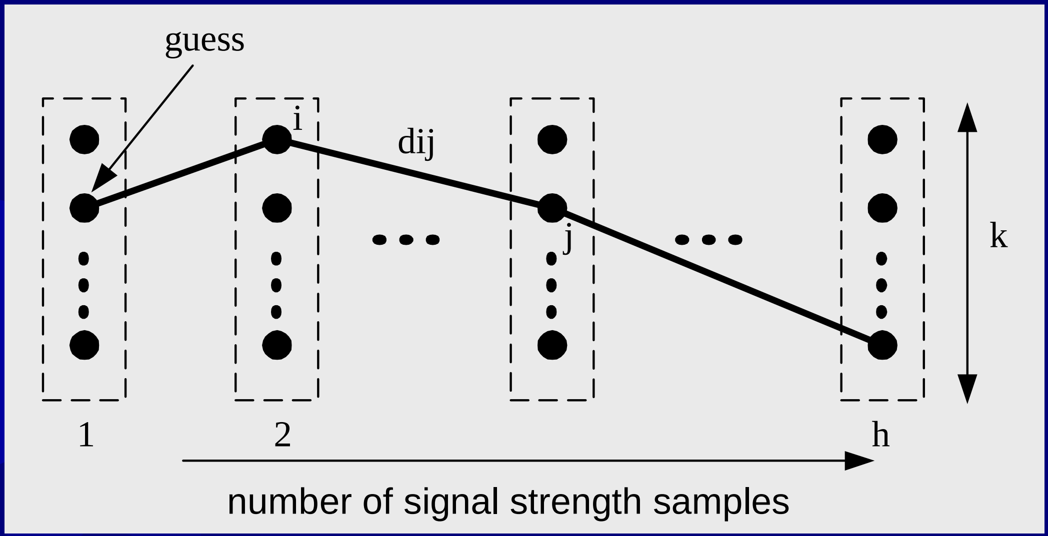

Median error distance is 2.94 m. Averaging (*k*=3) brings this down to 2.13 m

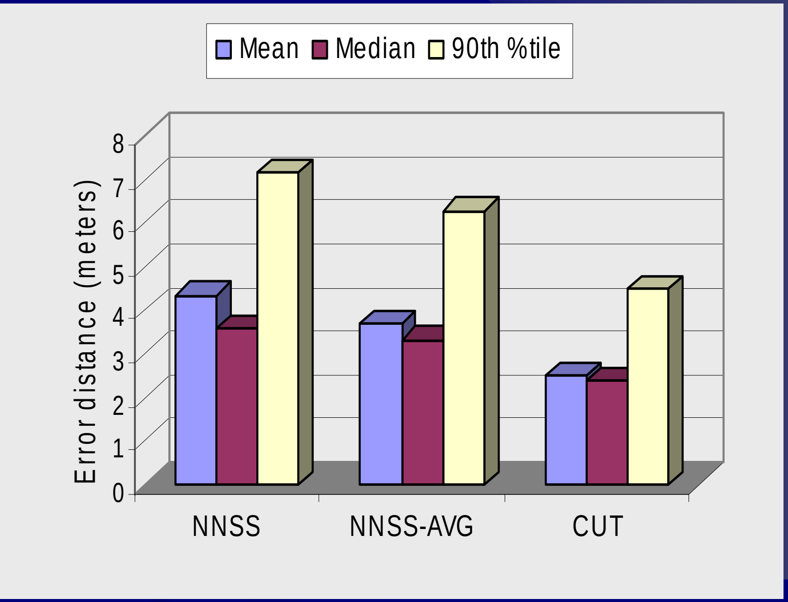
# Dynamic RADAR System

- Enhances the base system in several ways
  - mobile users
  - changes in the radio propagation environment
  - multiple radio channels
- DRS incorporates new algorithms
  - continuous user tracking
  - environment profiling
  - channel switching

# Continuous User Tracking

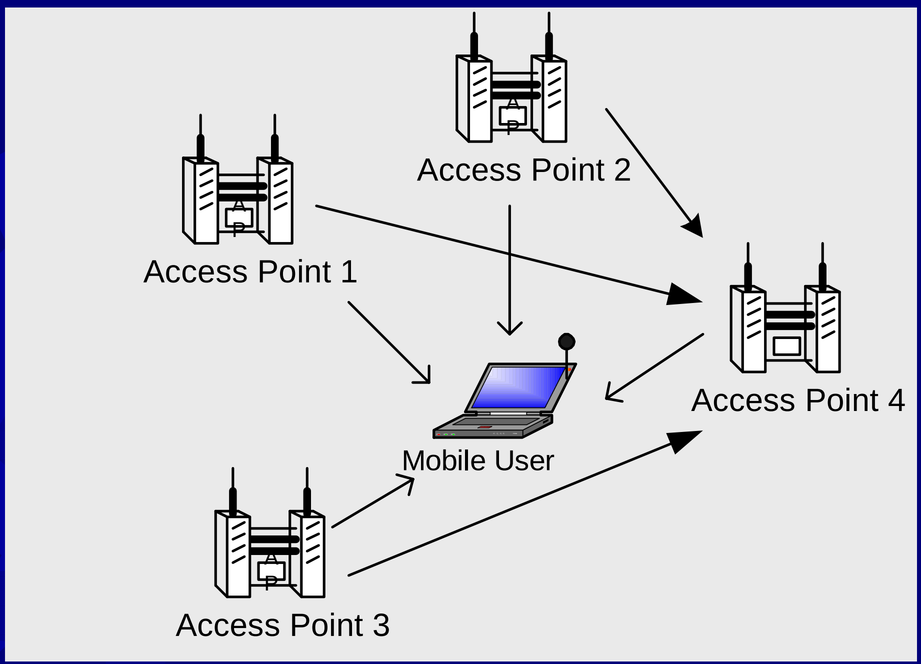
- History-based model that captures physical constraints
- Find the lowest cost path (à la Viterbi algorithm)
- Addresses the problem of *signal strength aliasing*

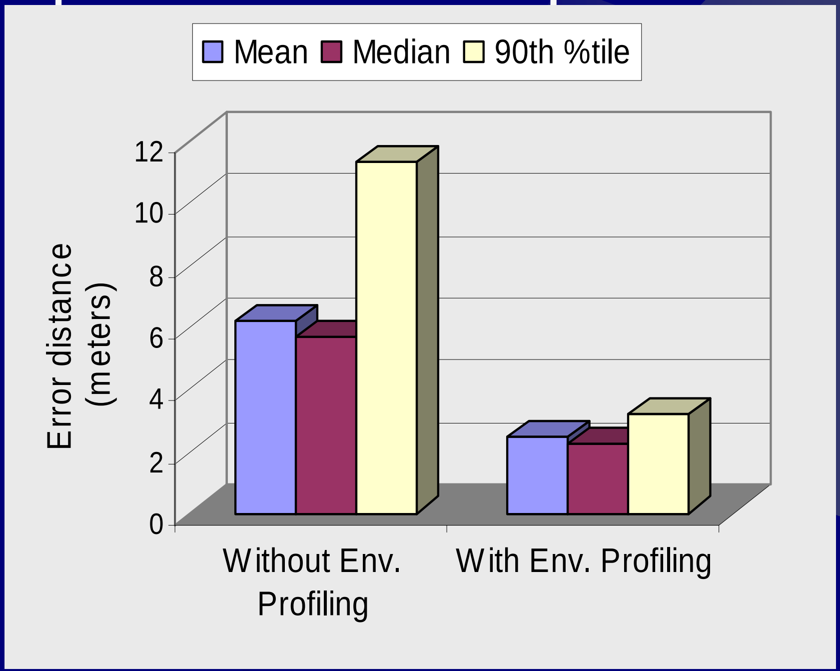


# Environment Profiling

- Addresses problem of changing RF environment
- System maintains multiple radio maps
- Maps indexed by environment profiles created by APs
- APs probe the environment and pick the best map



# Summary of RADAR

- RADAR: a software approach to user positioning
  - leverages existing wireless LAN infrastructure ⇒ low cost
  - enables autonomous operation ⇒ user privacy maintained
- Base system
  - radio map constructed either empirically or mathematically
  - NNSS algorithm matches signal strength against the radio map
- Enhanced system
  - continuous user tracking
  - environment profiling
- Median error: ~2 meters
- Publications:
  - Base system: INFOCOM 2000 paper
  - Enhanced system: Microsoft Technical Report MSR-TR-2000-12

# IP2Geo

(Joint work with L. Subramanian)

# Motivation

- Much focus on location-aware services in wireless and mobile contexts
- Such services are relevant in the Internet context too
  - targeted advertising
  - event notification
  - territorial rights management
  - network diagnostics
- Locating the user or host is a prerequisite
- But this is a challenging problem
  - IP address does not inherently contain an indication of location

# Existing Approaches

- User input
  - burdensome, error-prone
- User registration/cookies: e.g., Hotmail
  - better, but many services do not require the user to log in
  - cookie information may not be always available
  - registered location may be incorrect or stale
- *Whois* database: e.g., NetGeo
  - registered location may correspond to headquarters
  - manual updates, inconsistent databases
- Proprietary technology
  - Traceware (Digital Island), EdgeScape (Akamai)
  - country/state resolution
  - exhaustive tabulation of IP address space exploiting view from within ISP networks?

# IP2Geo

Multi-pronged approach that exploits various "properties" of the Internet

- DNS names of router interfaces often indicate location
- network delay tends to correlate with geographic distance
- hosts that are aggregated for the purposes of Internet routing also tend to be clustered geographically

- *GeoTrack*
  - determine location of closest router with a recognizable DNS name
- *GeoPing*
  - use delay measurements to estimate location
- *GeoCluster*
  - extrapolate partial (and possibly inaccurate) IP-to-location mapping information using BGP prefix clusters

# GeoTrack

- Location info often embedded in router DNS names
  - ngcore1-serial8-0-0-0.Seattle.cw.net, 184.atm6-0.xr2.ewr1.alter.net
- GeoTrack operation
  - do a *traceroute* to the target IP address
  - determine location of last recognizable router along the path
- Key ideas in GeoTrack
  - partitioned city code database to minimize chance of false match
  - ISP-specific parsing rules
  - delay-based correction
- Limitations
  - routers may not respond to *traceroute*
  - DNS name may not contain location information or lookup may fail
  - target host may be behind a proxy or a firewall

# GeoTrack Example

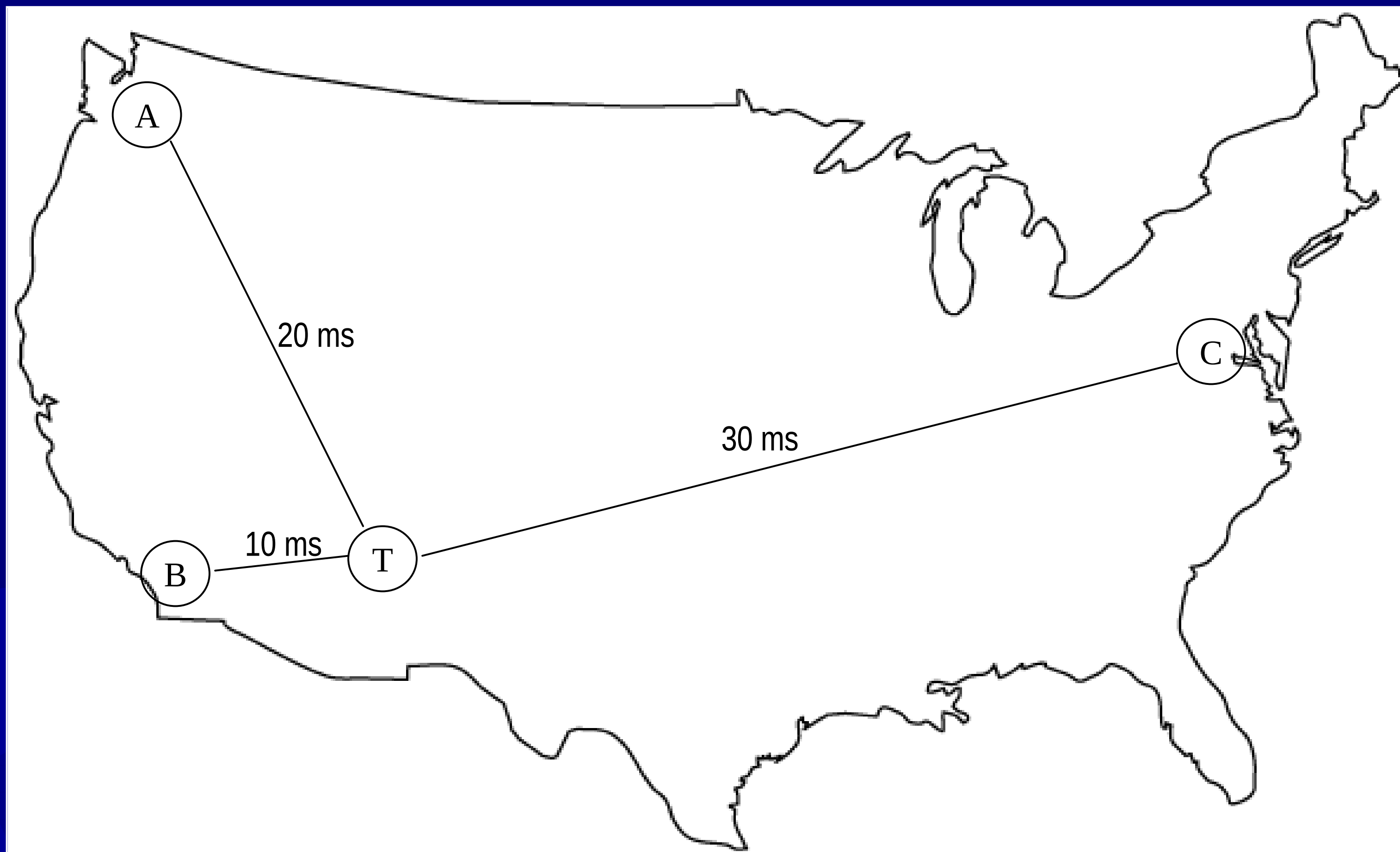
Traceroute from Berkeley to Dartmouth

| | | |
|---|---|---|
| snr46.CS.Berkeley.EDU | Berkeley,CA | UnitedStates |
| gig10-cnr1.EECS.Berkeley.EDU | Berkeley,CA | UnitedStates |
| gigE5-0-0.inr-210-cory.Berkeley.EDU | Berkeley,CA | UnitedStates |
| fast1-0-0.inr-001-eva.Berkeley.EDU | Berkeley,CA | UnitedStates |
| pos0-0.inr-000-eva.Berkeley.EDU | Berkeley,CA | UnitedStates |
| pos3-0.c2-berk-gsr.Berkeley.EDU | Berkeley,CA | UnitedStates |
| SUNV--BERK.POS.calren2.net | Sunnyvale,CA | UnitedStates |
| abilene--QSV.POS.calren2.net | Sunnyvale,CA | UnitedStates |
| dnvr-scrm.abilene.ucaid.edu | Denver,CO | UnitedStates |
| kscy-dnvr.abilene.ucaid.edu | KansasCity,MO | UnitedStates |
| ipls-kscy.abilene.ucaid.edu | Indianapolis,IN | UnitedStates |
| clev-ipls.abilene.ucaid.edu | Cleveland,OH | UnitedStates |
| nycm-clev.abilene.ucaid.edu | NewYork,NY | UnitedStates |
| 192.5.89.101 | | |
| 192.5.89.54 | | |
| bb.berry1-rt.dartmouth.edu | | UnitedStates |
| webster.dartmouth.edu | | UnitedStates |

# Delay-based Location Estimation

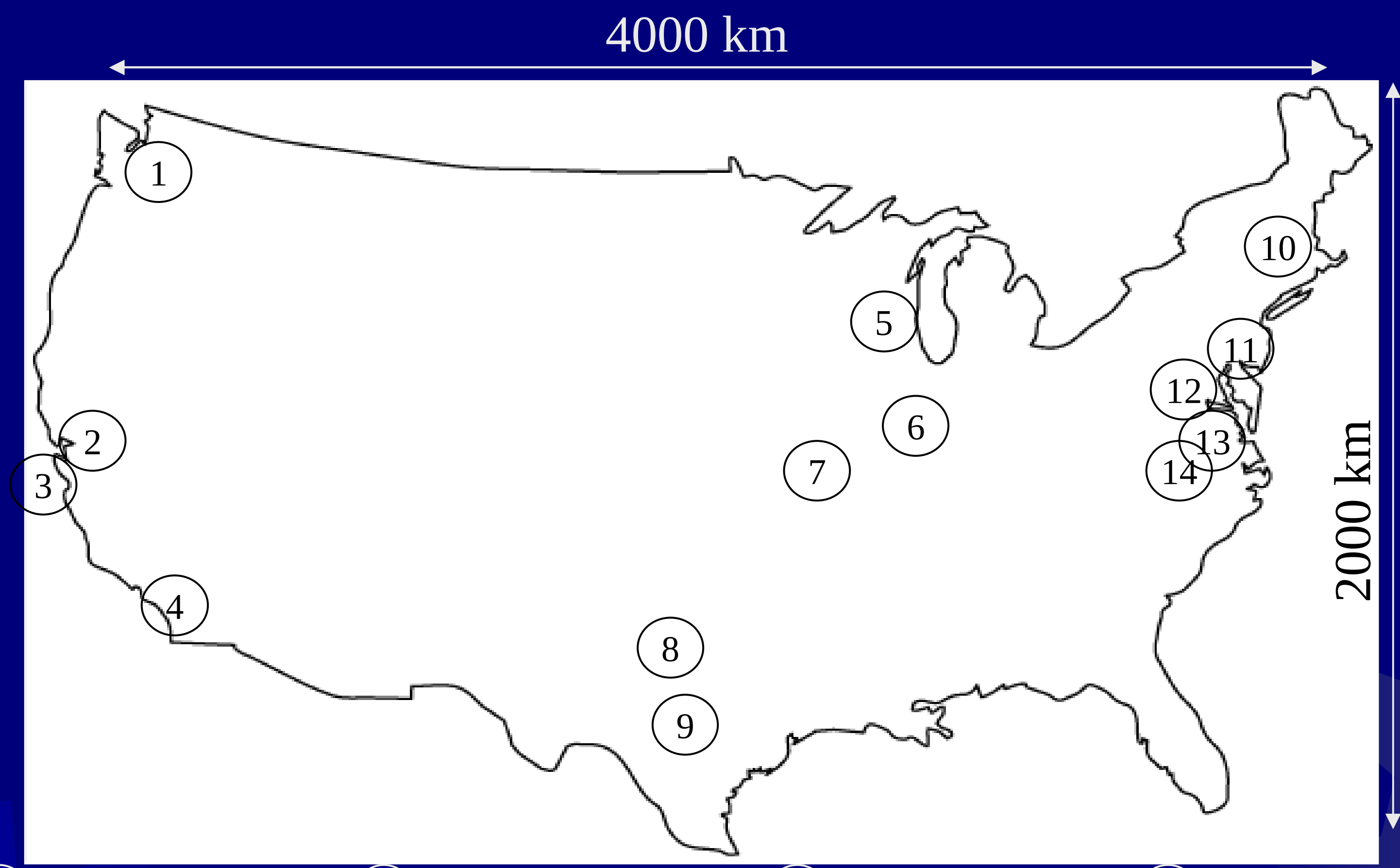
- Delay-based triangulation is conceptually simple
  - delay $\Rightarrow$ distance
  - distance from 3 or more non-collinear points $\Rightarrow$ location

- But there are practical difficulties
  - network path may be circuitous
  - transmission & queuing delays may corrupt delay estimate
  - one-way delay is hard to measure
    - one-way delay $\neq$ round-trip delay/2 because of routing asymmetry


A
20 ms
C
30 ms
10 ms
T
B

# GeoPing

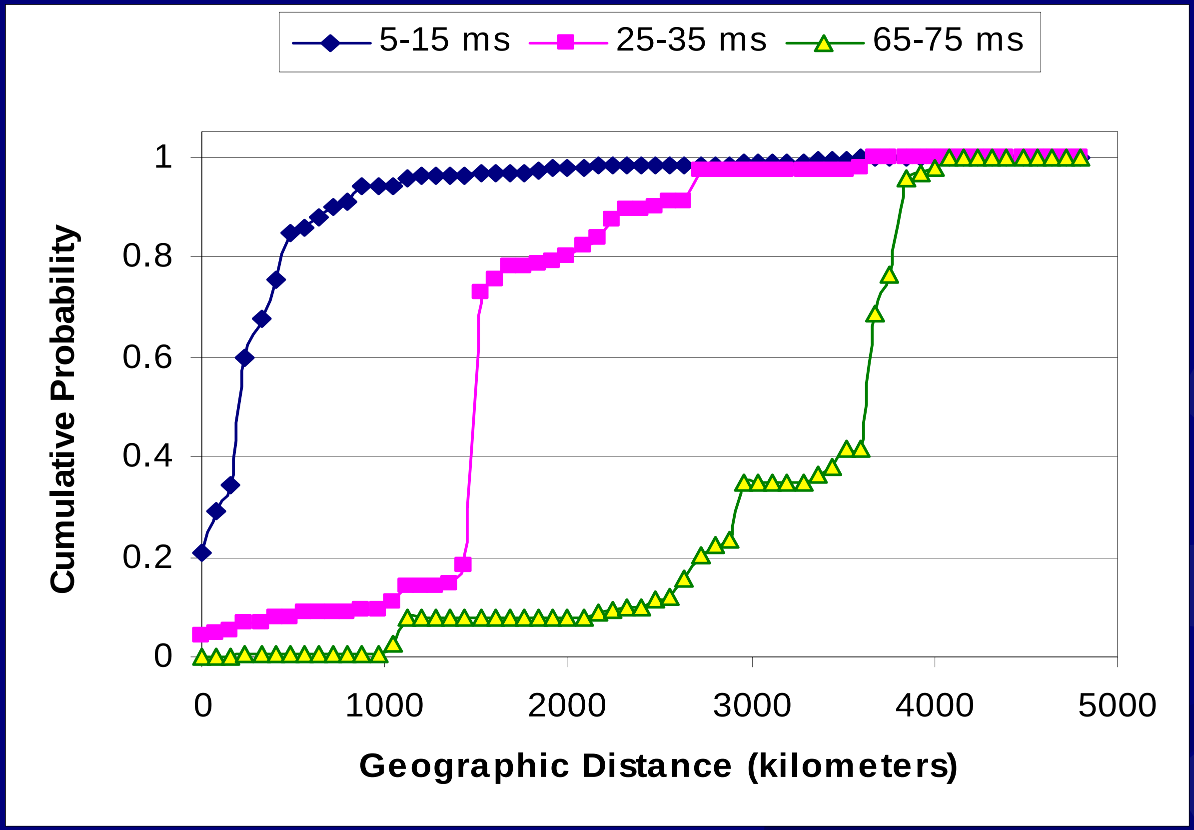
- Measure the network delay to the target host from several geographically distributed *probes*
  - typically more than 3 probes are used
  - round-trip delay measured using *ping* utility
  - small-sized packets $\Rightarrow$ transmission delay is negligible
  - pick minimum among several delay samples
- *Nearest Neighbor in Delay Space* (NNDS)
  - akin to Nearest Neighbor in Signal Space (NNSS) in RADAR
  - construct a *delay map* containing (delay vector,location) tuples
  - given a vector of delay measurements, search through the delay map for the NNDS
  - location of the NNDS is our estimate for the location of the target host
  - More robust that directly trying to map from delay to distance




1 Redmond, WA
2 Berkeley, CA
3 Stanford, CA
4 San Diego, CA
5 Madison, WI
6 Urbana, IL
7 St. Louis, MO
8 Dallas, TX
9 Austin, TX
10 Boston, MA
11 New Brunswick, NJ
12 Baltimore, MD
13 Durham, NC
14 Chapel Hill, NC

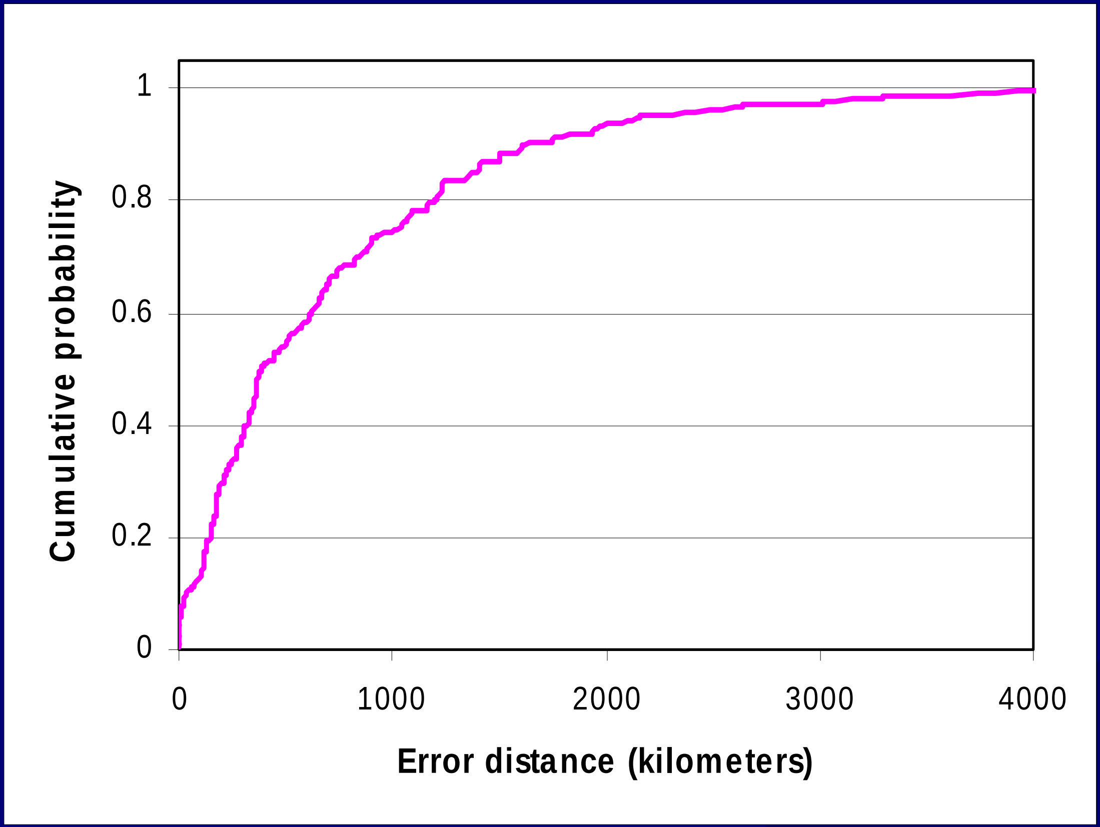
Delay map constructed using measured delays to 265 hosts on university campuses

# Validation of Delay-based Approach



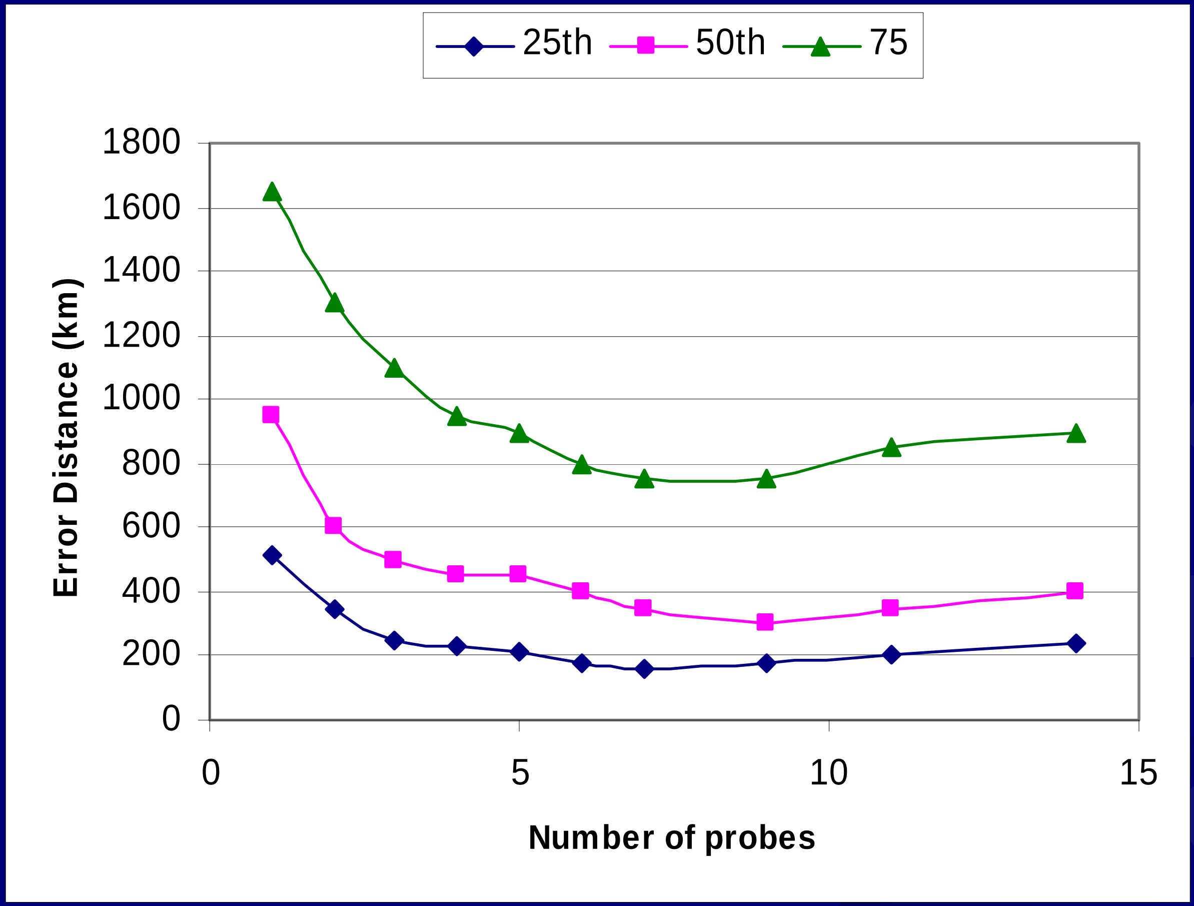

Delay tends to increase with geographic distance

# Performance of GeoPing




9 probes used. Error distance: 177 km (25$^{th}$), 382 km (50$^{th}$), 1009 km (75$^{th}$)

# Performance of GeoPing




Highest accuracy when 7-9 probes are used

# GeoCluster

- A passive technique unlike GeoTrack and GeoPing
- Basic idea:
  - divide up the space of IP addresses into *clusters*
  - extrapolate *partial* IP-to-location mapping information to assign a location to each cluster
  - given a target IP address, first find the matching cluster using longest-prefix match.
  - location of matching cluster is our estimate of host location
- Example:
  - consider the cluster 128.95.0.0/16 (containing 65536 IP addresses)
  - suppose we know that the location corresponding to a few IP addresses in this cluster is Seattle
  - then given a new address, say 128.95.4.5, we deduce that it is likely to be in Seattle too

# Clustering IP addresses

- Exploit the hierarchical nature of Internet routing
  - we use the approach proposed by Krishnamurthy & Wang (SIGCOMM 2000)
  - inter-domain routing in the Internet uses the *Border Gateway Protocol* (BGP)
  - BGP operates on address aggregates
  - we treat these aggregates as clusters
  - in all we had about 100,000 clusters of different sizes

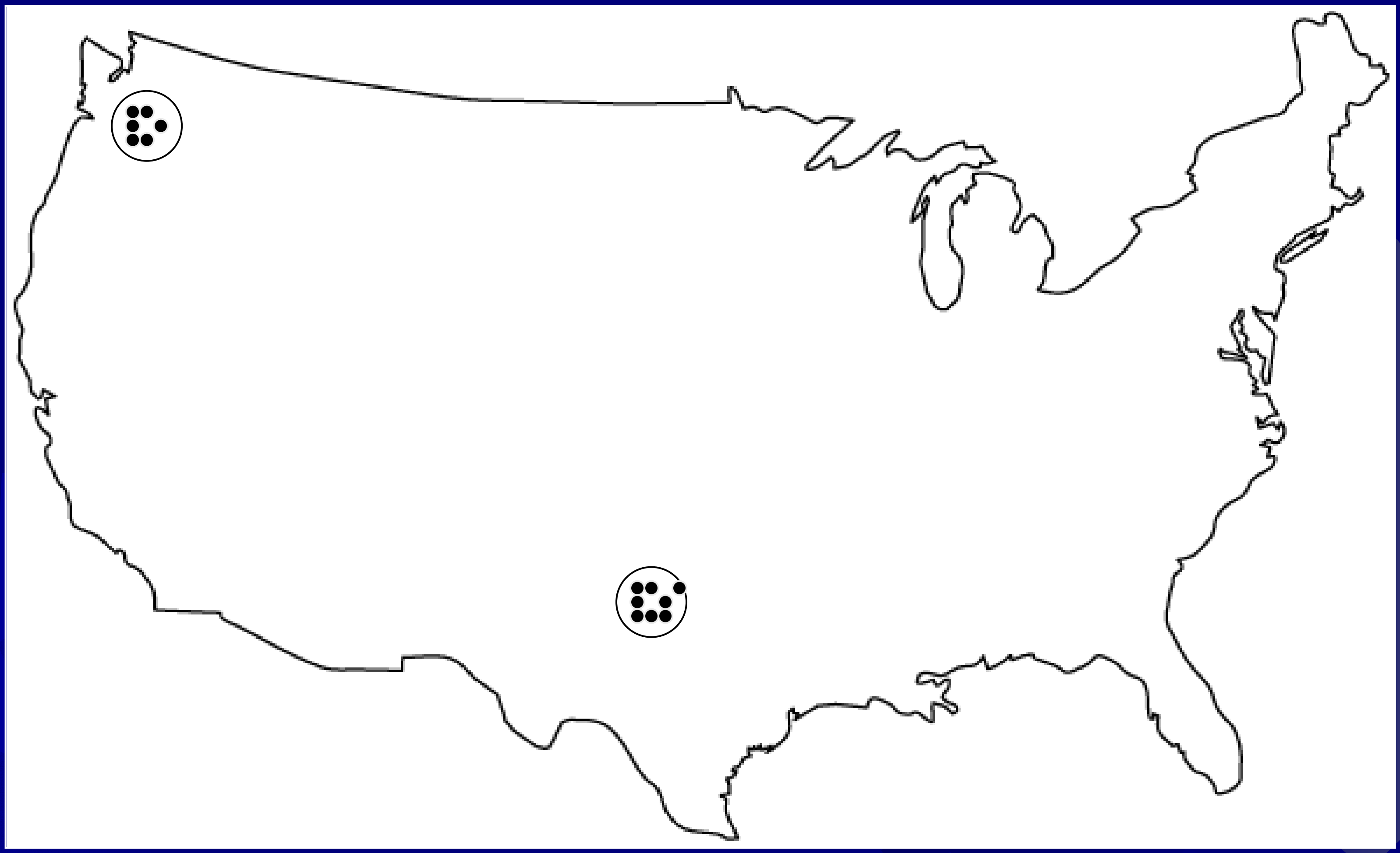
# IP-to-location Mapping

- IP-to-location mapping information
  - *partial* information (i.e., only for a small subset of addresses)
  - possibly *inaccurate* (e.g., manual input from user)
- We obtained mapping information from a variety of sources
  - *Hotmail*: combined anonymized user registration information with client IP address
  - *Online TV guide*: combined zip code submitted in user query with client IP address
  - *bCentral*: derived location information from cookies
- How would this information be obtained in general?
  - likely location (not necessarily accurate) may be inferred from user queries (e.g., TV guide)
  - location information from small number of registered users could be extrapolated to a much larger number of casual users

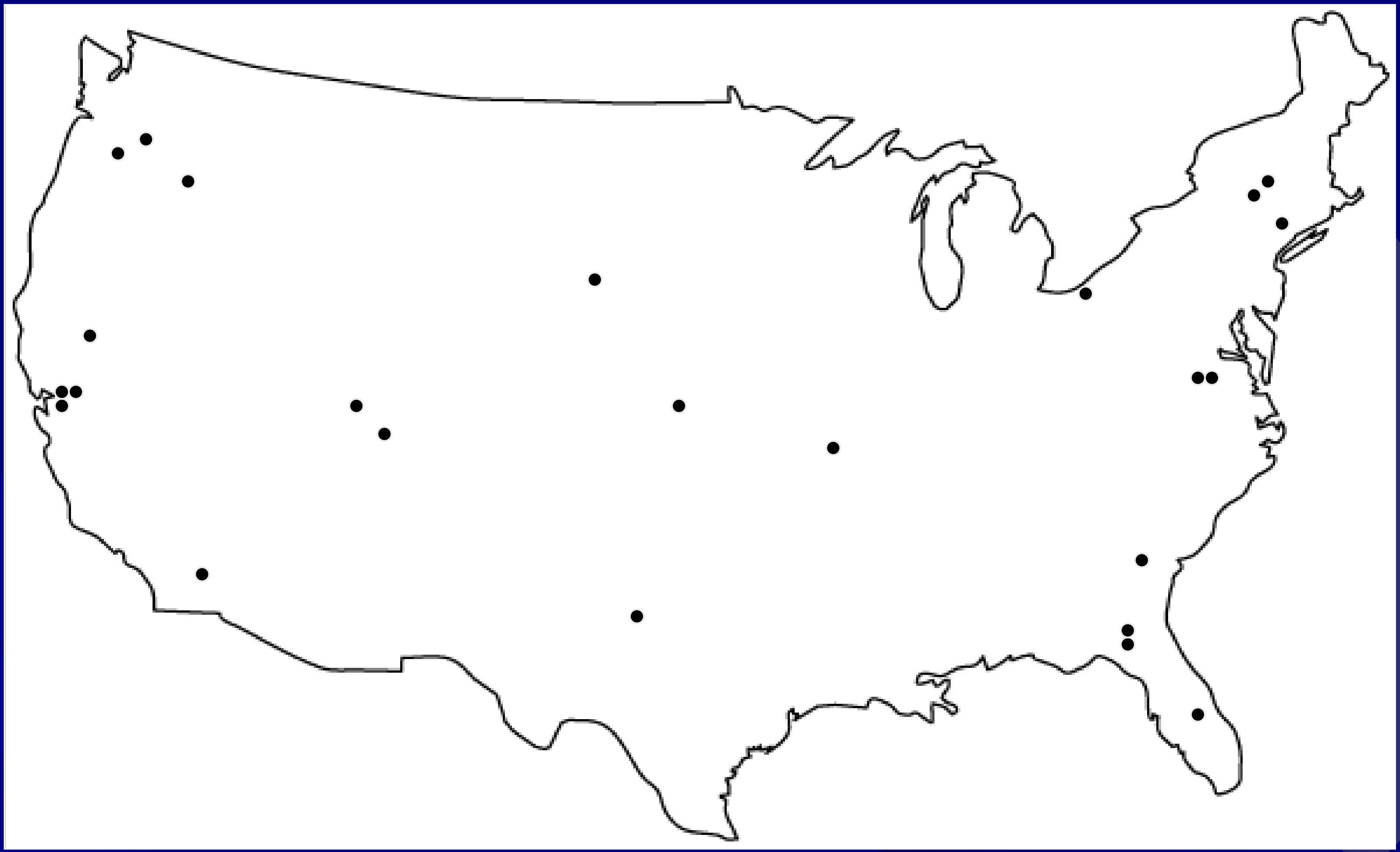
# Extrapolating IP-to-location Mapping

- Determine location most likely to correspond to a cluster
  - majority polling
  - "average" location
  - *dispersion* is an indicator of our confidence in the location estimate
- What if there is a large geographic spread in locations?
  - some clusters correspond to large ISPs and the internal subdivisions are not visible at the BGP level
  - *sub-clustering algorithm*: keep sub-dividing clusters until there is sufficient consensus in the individual sub-clusters
  - some clients connect via proxies or firewalls (*e.g.*, AOL clients)
    - sub-clustering may help if there are local or regional proxies
    - otherwise large dispersion $\Rightarrow$ no location estimate made
    - many tools fail in this regard

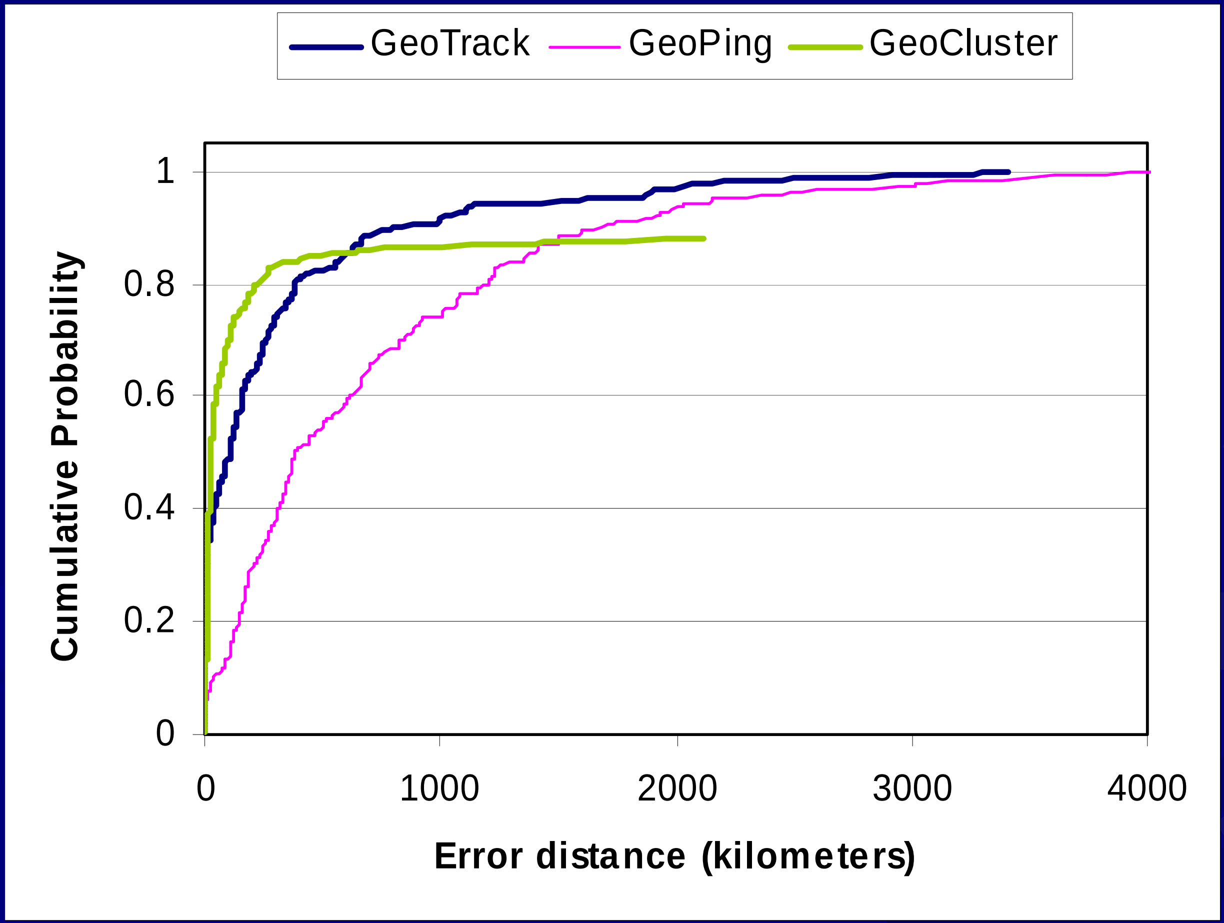
# Geographically Localized Clusters



# Geographically Dispersed Clusters



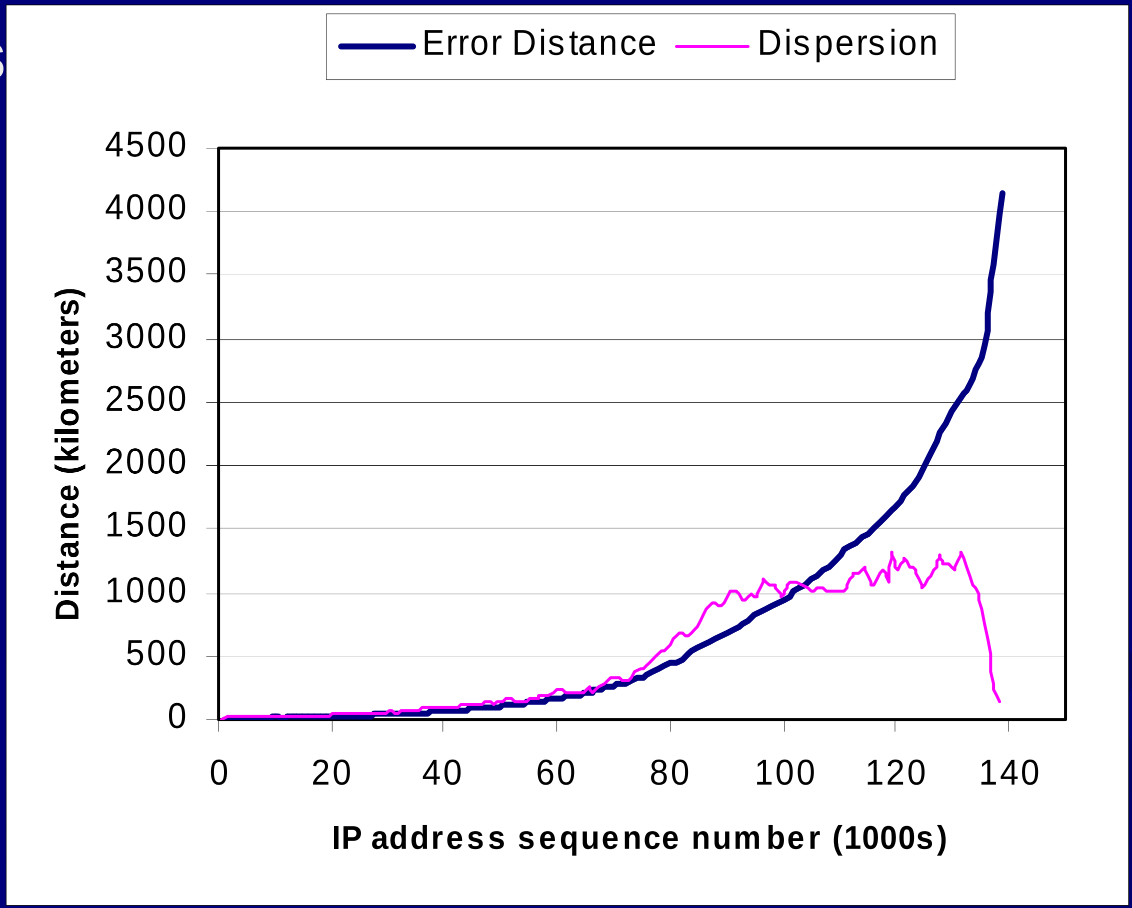
# Performance of GeoCluster




Median error: GeoTrack: 102 km, GeoPing: 382 km, GeoCluster: 28 km

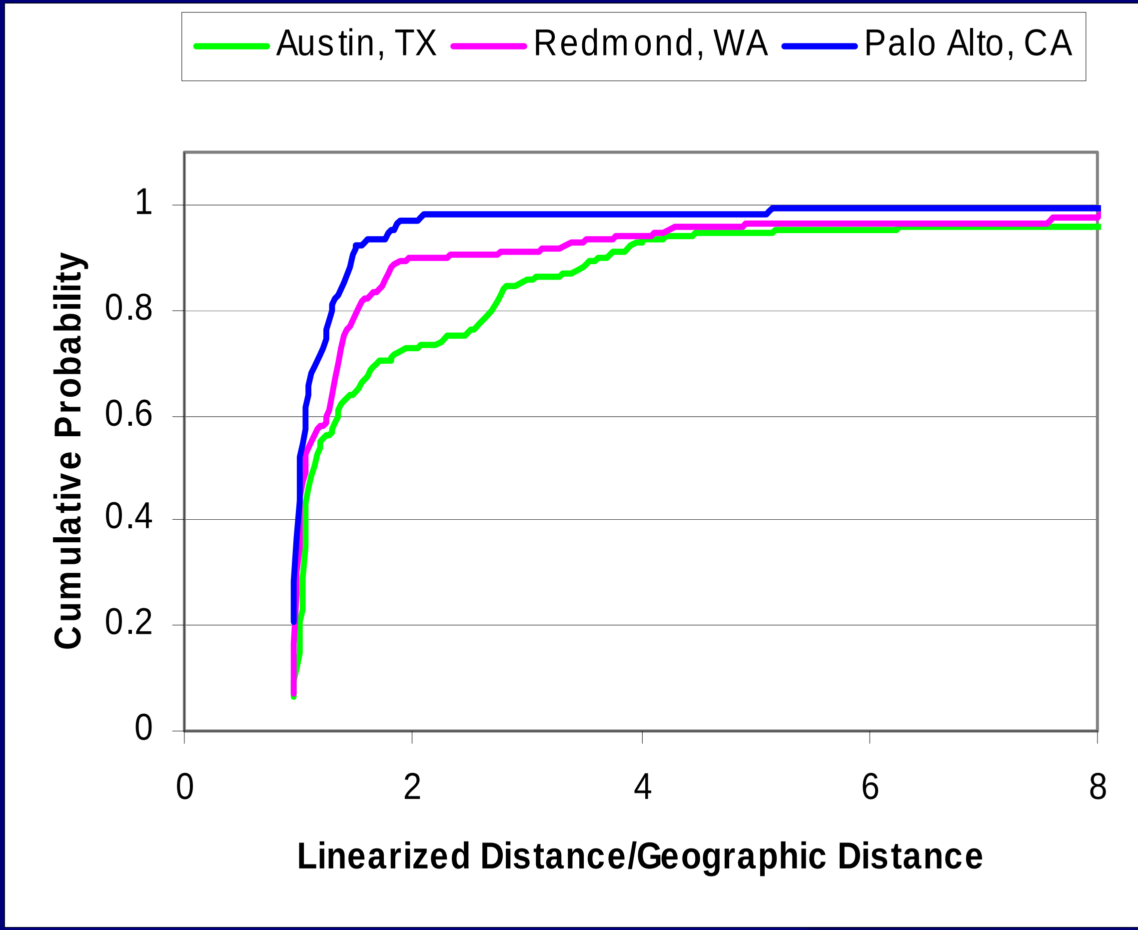
# Performance of GeoCluster

bCentral clients




Dispersion is on average a good indicator of accuracy

# Using IP2Geo to Study Internet connectivity




Path from TX to KY: TX → CA → NJ → IN → KY

# Summary of IP2Geo

- A variety of techniques that depend on different sources of information
  - GeoTrack: DNS names
  - GeoPing: network delay
  - GeoCluster: address aggregates used for routing
- Median error varies 20-400 km
- Even a 30% success rate is useful especially since we can tell when the estimate is likely to be accurate
- Paper to appear in ACM SIGCOMM 2001

# Conclusions

- RADAR and IP2Geo try to solve the same problem in very different contexts
  - wireless versus wireline
  - indoor environment versus global scale
  - accuracy of a few meters versus tens or hundreds of kilometers
- Interesting but challenging problem!
- For more information visit: *http://www.research.microsoft.com/~padmanab/*