eecs589

ADVANCED COMPUTER NETWORKS

Padmanabhan, V.N. and Subramanian, L., "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proc. of ACM SIGCOMM '01*, 31(4):173-185, Oct. 2001

Gummadi, K.P., Saroiu, S., and Gribble S.D., "King: Estimating Latency Between Arbitrary Internet End Hosts," *Proc. of Internet Measurement Workshop '02*, pp. 5-18, 2002

Where in the World is `141.218.5.5`?

IP addresses don't encode geographic locations

Uses of location info:

- targeted advertising
- geographic market analysis
- geographic DMA rights management

3 Techniques to Infer Geo Location

GeoTrack: based on DNS names of target host or of routers along the path to target host

GeoPing: by triangulating rtt measurements from multiple probing hosts

GeoCluster: based on AP clustering by registered location information

Median error: 28 km (Ann Arbor - Chelsea) to several hundred km (Ann Arbor – Chicago or farther)

Limitations

Require registration:

- burdensome
- mobility
- inaccurate (possibly deliberate)

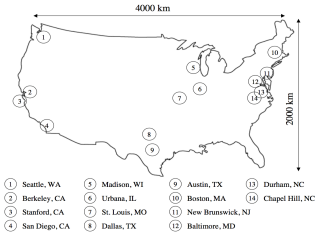
Proxies, NAT boxes, firewalls:

- may not necessarily be co-located (AOL)

Experiment Setup

14 probing hosts used for GeoTrack and GeoPing

265 academic web server as target hosts



Registered location sources:

- hotmail: registered states of 417,721 users
- bCentral: zip codes from HTTP cookies of 181,246 unique IP addresses
- FooTV: associating zipcode in queries of 142,807 unique IP addresses giving 336,181 (IP, zip) pairs

Zipcode can be converted to lat/lon [US Census Bureau]

GeoTrack

Geographic info encoded in router names: city codes, airport codes, country codes

Collected using `traceroute`, extracted by string matching

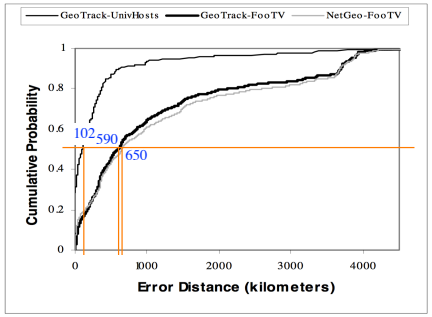
ISP-specific location codes

ISP-specific parsing rules (different positions)

- Sprint: `sl-bb10-`**`sea`**`-9.0.sprintlink.net`
- AlterNet: `192.atm4-0.sr1.`**`atl`**`5.alter.net`

GeoTrack: Performance

Metric: error distance = geographic distance between estimated and registered locations



102 km: ~ Ann Arbor – Lansing, MI
590 km: ~ Ann Arbor – Champaign, IL
650 km: ~ Ann Arbor – Springfield, IL

Figure 3: CDF of the error distance for GeoTrack and NetGeo.

NetGeo: uses `Whois` registration

GeoPing

Correlate network delay and geographic distance

Potential issues:

- circuitous paths
- congestion: discount queueing delay (MIN(measurements))
- asymmetric paths?

Presumed low correlation, but increased bandwidth and coverage raised correlation(?)

GeoPing

Paths circuitous?

- linearized distance: sum of geographic distance between hops along path (as determined by GeoTrack!)
- linearized/geographic distances $\approx 1 \Rightarrow$ not circuitous
- ping from 3 academic sites to 265 academic web sites

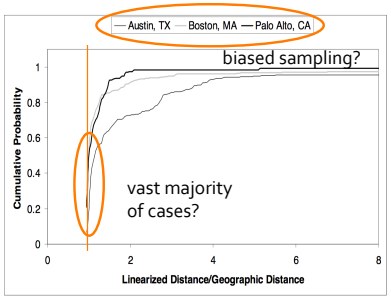


Figure 4: CDF of the ratio of linearized distance to geographic distance for Internet paths originating from three locations.

GeoPing

Delay vs. geographic distance

- cannot be modeled analytically
- short delays (< 10 ms) are within 300 km
- longer delays show cliffs in CDF, but not well correlated

$\Rightarrow$ authors do not attempt to map delay to distance directly

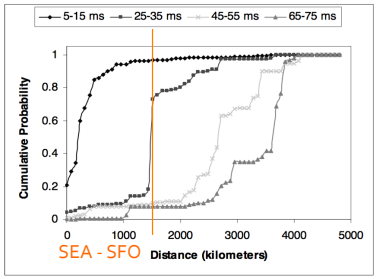


Figure 5: Cumulative Distribution of geographic distance for multiple delay ranges based on data gathered at the Seattle, WA probe location.

Nearest Neighbor in Delay Space

Probe each target from k out of n probing hosts

Each target gets a distance vector: $(d_1, \ldots, d_k)$

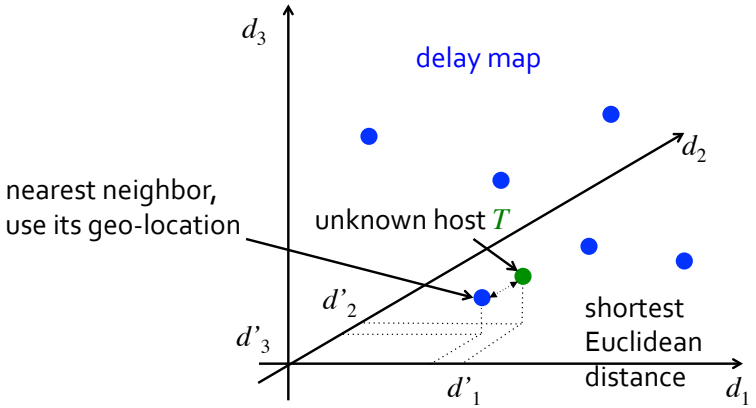
Construct a delay map consisting of (1) hosts with known locations and (2) their distance vectors

To locate an unknown host T:

1. measure its distance vector from the k probe hosts: $(d'_1, \ldots, d'_k)$
2. compute the Euclidean distance between T and all hosts in the delay map: $\sqrt{[(d_1-d'_1)^2+\ldots+(d_k-d'_k)^2]}$
3. assign the location of the host in the delay map with minimum Euclidean distance to T as T's location

Nearest Neighbor in Delay Space

Illustrated with 3 probe hosts:



Nearest Neighbor in Delay Space

How many probing hosts? 7-9

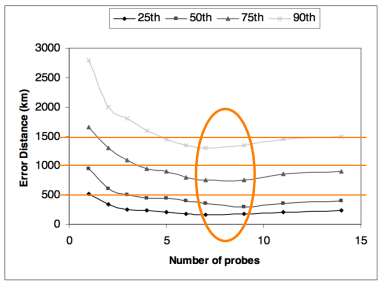


Figure 6: Error distance versus number of probes.

Error distance:
- 150 km at 25%-tile
- 800 km at 75%-tile
- 1300 – 1400 km at 90%-tile
- why the increase with > 9 probes?

For a given number, best to have the probing hosts geographically distributed

GeoCluster

Based on (1) BGP routable Address Prefixes (APs) and(2) registered geographic location of some IPs

AP's geographic location based on consensus IP locations of constituent hosts

If there's no significant consensus, split AP in half and seek consensus for the smaller APs

Significant consensus:

1. state-level info: `cthresh` number of IPs can be geo-located and `fthresh` of them agrees on geo-location
2. zipcodes: `cthresh` number of IPs can be geo-located and lat/lons are not widely dispersed (self-calibrating)

GeoCluster: AP Splitting

ASs aggregate APs in advertisement

An AP associated with multiple locations can be split:

- AP `152.153.0.0/16` is advertised through BGP
- the AP is associated with 3 geo-locations: NYC, DFW, SFO
- split the AP into 2 halves: `152.153.0.0/17` (A) and `152.153.128.0/17` (B)
- 50 IPs located in NYC are in A ⇒ AP A is a geographic cluster
- there's still not sufficient consensus on B, so it's further split into `152.153.128.0/18` ($B1$) and `152.153.192.0/18` ($B2$)
- 30 IPs located in DFW are in $B1$ ⇒ AP $B1$ is a geographic cluster
- 10 IPs located in SFO are in $B2$, 10 is smaller than cthresh ⇒ AP $B2$ is not a geographic cluster

GeoCluster: Dispersion Metric

Zipcode can be translated into lat/lon

Compute composite location (l_{avg}) by linear averaging lat/lon of locations within a cluster (or AP)

Dispersion quantifies the geographic spread of a cluster

$$\text{dispersion} = \sum_{i \in L} \text{dist}(l, l_{avg}) / |L|$$

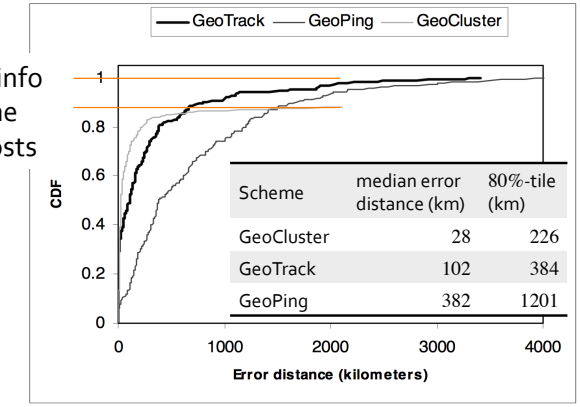
L: set of lat/lon's in cluster

GeoCluster can't find cluster for geographically dispersed clients sharing a remote proxy (AOL case)

> "We believe this is an important property of the sub-clustering algorithm because for many applications a highly inaccurate location estimate may be strictly worse than no location estimate at all."

GeoCluster: Experimental Results

No location info on 12% of the academic hosts



Scheme	median error distance (km)	80%-tile (km)
GeoCluster	28	226
GeoTrack	102	384
GeoPing	382	1201

Figure 7: CDF of the error distance computed over the UnivHosts data set for GeoTrack, GeoPing, and GeoCluster.

GeoCluster: Experimental Results

For bCentral hosts: only 77% could be placed

Median error distance is 685 km, 3,056 km at 75%-tile

bCentral hosts with higher dispersion has worse error distance (last data point was perhaps an anomaly associated with remote dialup)

Scheme	median error distance (km)	80%-tile (km)
GeoCluster	28	226
GeoTrack	102	384
GeoPing	382	1201

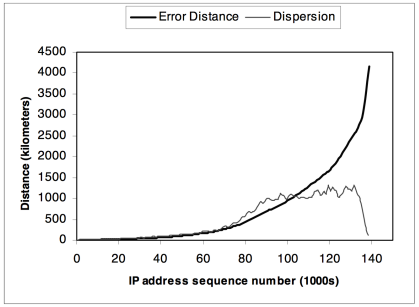


Figure 8: The error distance and the dispersion for hosts in bCentral.

GeoCluster: Experimental Results

Sub-clustering helps accuracy

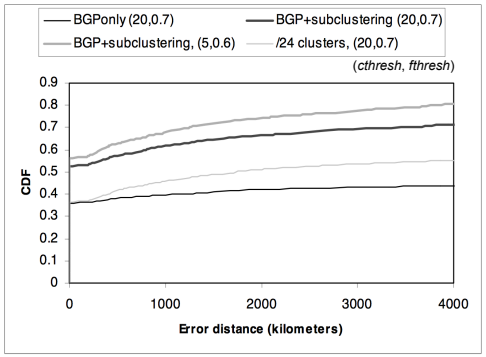


Figure 9: CDF of the error distance (computed at the granularity of states) for the BGPonly and BGP+subclustering variants of GeoCluster, and for the /24-clusters method.

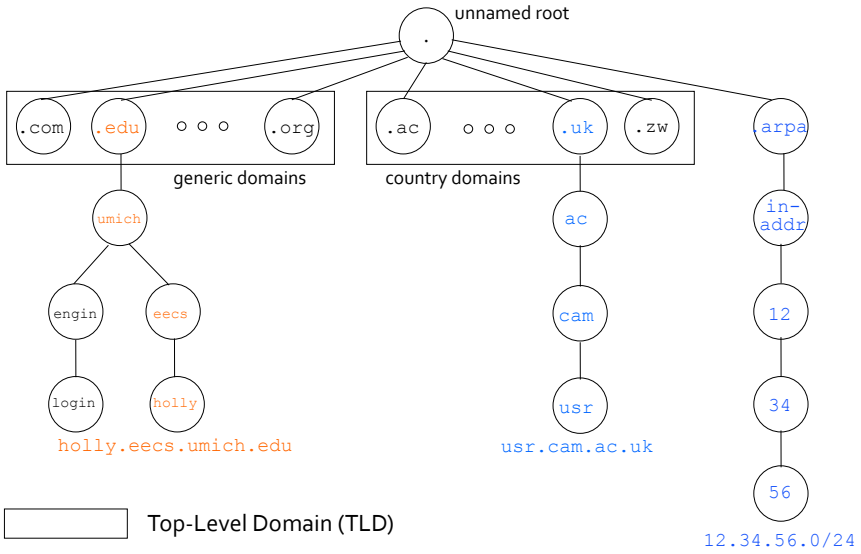
eecs589

ADVANCED COMPUTER NETWORKS

Padmanabhan, V.N. and Subramanian, L., "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proc. of ACM SIGCOMM '01*, 31(4):173-185, Oct. 2001

Gummadi, K.P., Saroiu, S., and Gribble S.D., "King: Estimating Latency Between Arbitrary Internet End Hosts," *Proc. of Internet Measurement Workshop '02*, pp. 5-18, 2002

DNS Hierarchical Name Space



DNS Name Servers

DNS database is partitioned into zones

A zone holds one or more domains, analogy:

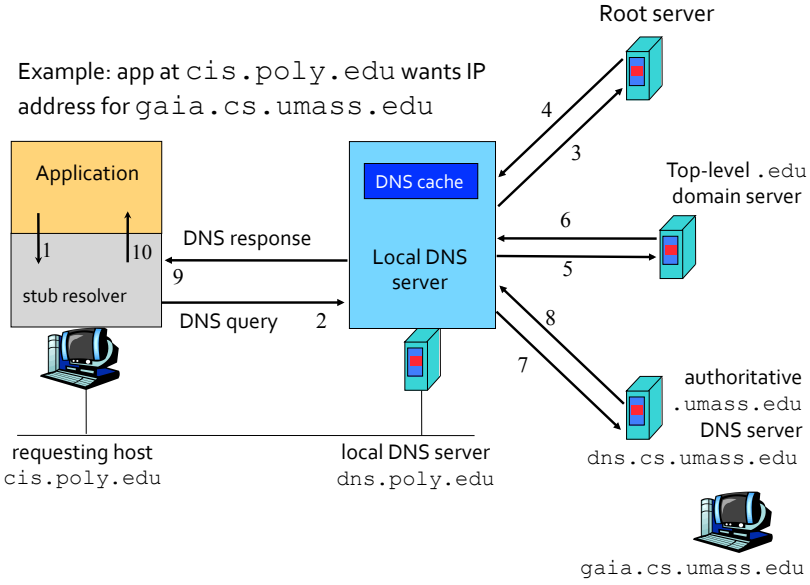
DNS	File System
domains	folders
zones	volumes

Name server: a process that manages a zone

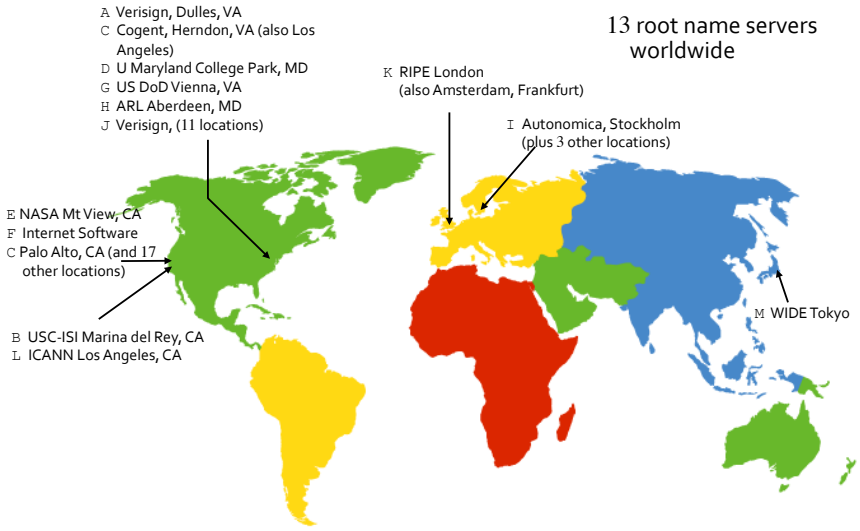
Authoritative or primary name server:
the "owner" of a zone

- providing authoritative mappings for organization's server names (e.g., web and mail)
  - can be maintained by an organization or its service provider

DNS Name Resolution



DNS Root Name Servers



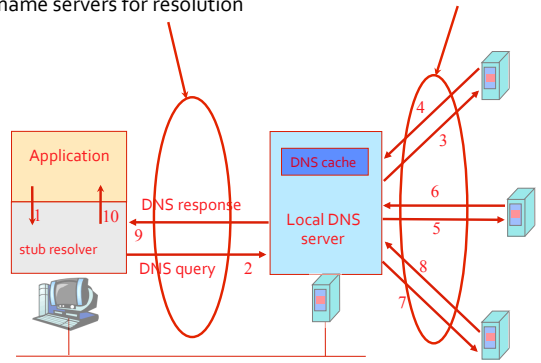
Recursive vs. Iterative Query

Recursive query:

- local name server must resolve the name (or return "not found"); if necessary, by asking other name servers for resolution

Iterative query:

- contacted server replies with the name of server address of sub-domain
  - "I don't know this name, but ask this other name server"
- requesting name server visits each name server referred to



Why not always do recursive resolution?

- puts burden of name resolution on contacted name server

DNS Caching

Once a (any) name server learns of a mapping, it caches the mapping

- to reduce latency in DNS translation

Cache entries timeout (disappear) after some time-to-live (TTL)

- TTL is assigned by the authoritative server (owner of the host name)

Local name servers typically also cache

- TLD name servers cache to reduce visits to root name servers
- all other name servers cache referrals
- cache both positive and negative results

King

Goal: to estimate network latencies between arbitrary Internet end hosts using DNS infrastructure

Uses of latency info:

- to investigate routing path inefficiencies
- to construct topologically-sensitive overlay networks
- closest server selection

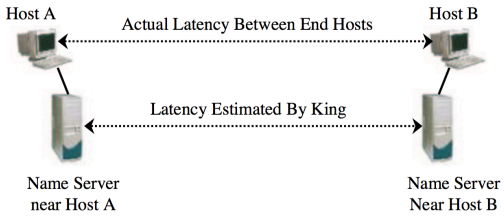
Advantages over existing approaches:

- vs. IDMaps: doesn't require deployment of additional infrastructure and doesn't require probing of end hosts
- vs. GNP: doesn't require well-known reference points

King: Approach

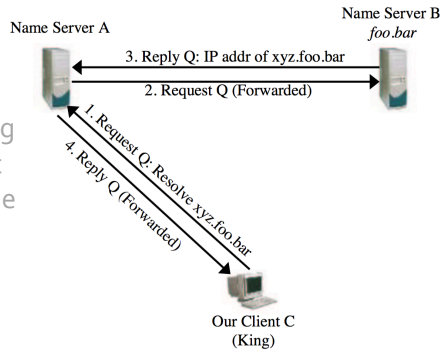
"To use existing protocols in unanticipated ways to obtain results that were previously intractable"

Estimate latency between two end hosts as the latency between their "closest" authoritative name servers



King: Approach

1. Ask A to recursively resolve for `foo.bar` of B
2. Measure latency to A (by ping or by resolving A's name)
3. Subtract the latter from the former
4. Query for multiple *`random_number`*`.foo.bar` to "prime" A with B and to obtain multiple measurements
5. Use DNS cache poisoning to force A to go to B, but must account for multiple retries [not used]



Evaluation

vs. IDMaps, a popular technique to estimate latencies

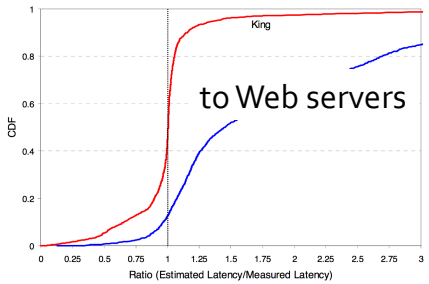
Disclaimer: IDMaps was a project supervised by yours truly

Hardly popular, just a research prototype

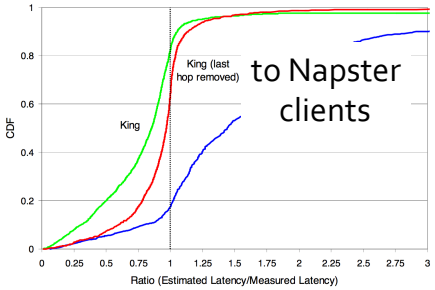
At most active, had only a handful of *tracers* on the Internet
⇒ Bogus comparative study

Evaluation

Estimated: King
Measured: `traceroute`

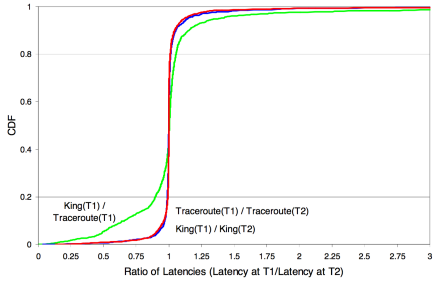


Removing last hop conveniently ignore cases when authoritative NS is not co-located with client?

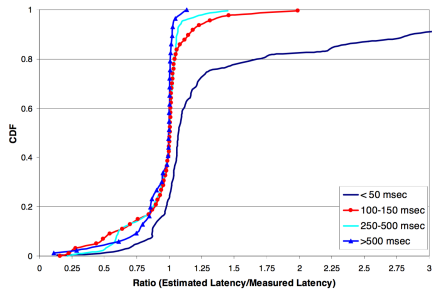


Evaluation

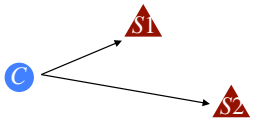
Consistency of estimates across time



Less accurate for longer paths



Evaluation



Rank accuracy:

Does King consistently rank $S1$ closer to C than to $S2$?

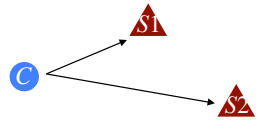
How well correlated are two sets of rankings?

$R_1 = \{r_{1,1}, r_{1,2}, r_{1,3}, \ldots, r_{1,n}\}$, $\bar{r}_1$: averages(?) of traceroute rankings

$R_2 = \{r_{2,1}, r_{2,2}, r_{2,3}, \ldots, r_{2,n}\}$, $\bar{r}_2$: averages(?) of King rankings

$$Rank\ correlation\ coefficient = \frac{\sum_{i=1}^{N}(r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2)}{\sqrt{\sum_{i=1}^{N}(r_{1,i} - \bar{r}_1)^2 (r_{2,i} - \bar{r}_2)^2}}$$

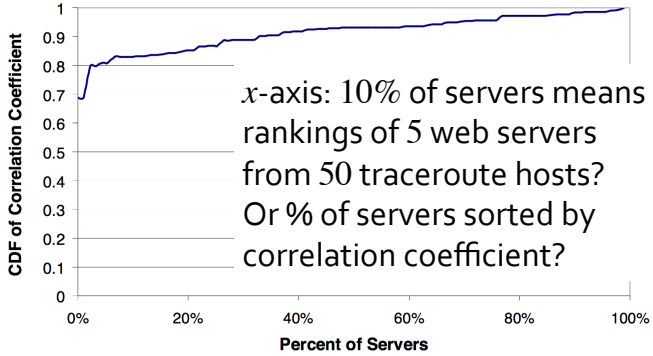
Evaluation



Rank accuracy:

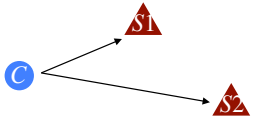
Does King consistently rank $S1$ closer to C than to $S2$?

How well correlated are two sets of rankings?



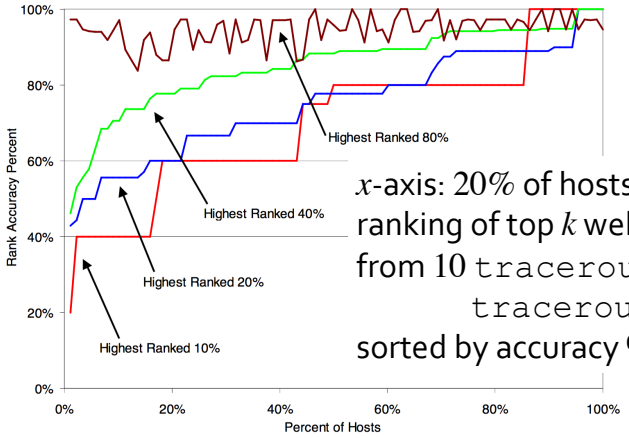
x-axis: 10% of servers means rankings of 5 web servers from 50 traceroute hosts?
Or % of servers sorted by correlation coefficient?

Evaluation



Ranked accuracy percent:

How many hosts are included in the top k% of 2 rankings?

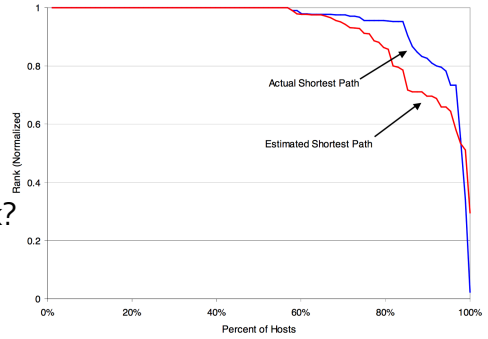


x-axis: 20% of hosts means ranking of top k web servers from 10 `traceroute` hosts?
`traceroute` hosts sorted by accuracy %age?

Evaluation

True rank:

Is King's 1st rank always `traceroute`'s 1st rank?



Yes for 60% of King's selection

(?) 80% of King's selection lies among the closest 20%

(?) 90% of true closest is in 20% of King's closest

Evaluation

How close are name servers to end host?

Hop count is meaningless

Ignoring the last hop to Napster clients from analysis discounted the true distance between end hosts and their name servers!

Self-diagnostic ability is thus suspect

A Comcast Client (2016)

```
% dig c-68-62-19-135.hsd1.mi.comcast.net
comcast.net.            164518  IN      NS      dns102.comcast.net. ; 68.87.85.132
comcast.net.            164518  IN      NS      dns105.comcast.net. ; 68.87.72.244
... ; dns101: 69.252.250.103; dns103: 68.87.76.228; dns104: 68.87.68.244

% traceroute 68.62.19.135
 1  141.212.113.1 (141.212.113.1)  0.946 ms  1.059 ms  1.186 ms
 ...
 6  50-224-111-17-static.hfc.comcastbusiness.net (50.224.111.17)  0.589 ms  0.594 ms  0.604 ms
 7  162.151.127.49 (162.151.127.49)  2.251 ms  2.452 ms  3.241 ms
 8  te-9-3-ur02.nannarbor.mi.michigan.comcast.net (68.85.222.246)  3.312 ms  3.413 ms  3.554 ms
 9  te-6-1-acr01.nannarbor.mi.michigan.comcast.net (68.86.120.170)  2.898 ms  2.974 ms  2.827 ms
10  * * *

% traceroute dns102.comcast.net
 1  141.212.113.1 (141.212.113.1)  0.899 ms  1.048 ms  1.191 ms
 ...
 6  50-224-111-17-static.hfc.comcastbusiness.net (50.224.111.17)  0.556 ms  0.594 ms  0.560 ms
 7  162.151.127.49 (162.151.127.49)  2.466 ms  2.252 ms  2.323 ms
 8  xe-0-0-0-0-sur01.macomb.mi.michigan.comcast.net (68.86.122.158)  5.257 ms 3.606 ms  3.685 ms
 9  be-33668-cr02.350ecermak.il.ibone.comcast.net (68.86.90.45)  11.173 ms  11.178 ms  12.620 ms
10 be-10517-cr02.denver.co.ibone.comcast.net (68.86.85.170)  35.007 ms  34.985 ms  34.961 ms
 ...
14  po5-sw303a-d.cmc.co.ndcwest.comcast.net (162.151.85.194)  34.352 ms  45.465 ms  35.081 ms
15 dns102.comcast.net (68.87.85.132)  33.936 ms  34.687 ms  34.881 ms

% traceroute dns105.comcast.net ;
 1  141.212.113.1 (141.212.113.1)  0.852 ms  0.984 ms  1.155 ms
 ...
 6  50-224-111-17-static.hfc.comcastbusiness.net (50.224.111.17)  1.047 ms  0.751 ms  0.736 ms
 7  162.151.127.49 (162.151.127.49)  2.305 ms  2.464 ms  2.331 ms
 8  xe-0-0-0-0-sur02.macomb.mi.michigan.comcast.net (68.86.122.162)  3.613 ms  3.681 ms 5.036 ms
 9  be-33668-cr02.350ecermak.il.ibone.comcast.net (68.86.90.45)  10.472 ms  12.514 ms  12.794 ms
10  be-7922-ar01.area4.il.chicago.comcast.net (68.86.91.166)  12.452 ms  11.544 ms  11.532 ms
11  te-8-4-ur05-d.area4.il.chicago.comcast.net (68.87.210.6)  11.244 ms  11.383 ms  10.680 ms
12  dns105.comcast.net (68.87.72.244)  9.900 ms  9.803 ms  11.483 ms
```

A T-Mobile Client (2016)

```
% dig 163.20.56.172.in-addr.arpa ns
172.in-addr.arpa.       10158   IN      SOA     z.arin.net. ; 199.212.0.63

% traceroute 172.56.20.163
 1  141.212.113.1 (141.212.113.1)  0.946 ms  1.059 ms  1.186 ms
 ...
 6  ae4.anar-cor-cath.merit.edu (192.12.80.33)  0.457 ms  0.479 ms  0.501 ms
 7  aelx22.sfld-cor-123net.mich.net (198.108.23.50)  7.406 ms  7.398 ms  1.387 ms
 8  et-10-0-0.1279.rtr.ashb.net.internet2.edu (64.57.29.177)  21.386 ms  21.384 ms  21.361 ms
 9  64.57.20.106 (64.57.20.106)  21.379 ms  21.309 ms *
10  ae7.er1.iad10.us.zip.zayo.com (64.125.25.49)  21.839 ms  21.814 ms  21.821 ms
11  ae6.cr1.dca2.us.zip.zayo.com (64.125.20.117)  14.369 ms  14.450 ms  13.994 ms
12  ae27.cs1.dca2.us.eth.zayo.com (64.125.30.246)  24.792 ms  24.665 ms  24.700 ms
13  ae28.mpr4.atl6.us.zip.zayo.com (64.125.31.169)  33.246 ms  33.159 ms  33.219 ms
14 * * *

% traceroute z.arin.net
 1  141.212.113.1 (141.212.113.1)  0.899 ms  1.048 ms  1.191 ms
 ...
 6  ae4.anar-cor-cath.merit.edu (192.12.80.33)  0.471 ms  0.609 ms  0.573 ms
 7  aelx69.eq-chi2.mich.net (198.108.22.97)  6.312 ms  7.156 ms  7.131 ms
 8  12.250.16.17 (12.250.16.17)  6.385 ms  6.395 ms  6.406 ms
 9  cr1.cgcil.ip.att.net (12.122.133.122)  8.684 ms  8.716 ms  8.693 ms
10  gar8.cgcil.ip.att.net (12.122.133.161)  7.971 ms  7.943 ms  7.930 ms
11  ix-ae-15-0.tcore2.CT8-Chicago.as6453.net (64.86.79.41)  22.550 ms  20.490 ms  20.471 ms
12  if-ae-22-2.tcore1.CT8-Chicago.as6453.net (64.86.79.2)  44.702 ms  44.491 ms  44.895 ms
13  if-ae-26-2.tcore2.NTO-New-York.as6453.net (216.6.81.28)  45.693 ms  45.544 ms  45.616 ms
14  * * *
15  if-ae-11-4.tcore2.AEQ-Ashburn.as6453.net (216.6.87.168)  45.680 ms  44.834 ms if-ae-11-3.tcore2.AEQ-
Ashburn.as6453.net (216.6.87.241)  107.924 ms
16  * * *
17  * * *
18  66.198.9.30 (66.198.9.30)  45.709 ms  46.161 ms  45.936 ms
19  * * *
```