INTERNET MEASUREMENT: INFRASTRUCTURE (II)

Gaia Maselli

maselli@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

How to measure infrastructure properties

Tools

- *Active measurement*: adding traffic to the network for the purpose of measurement
- *Passive measurement*: capturing traffic that is generated by other users and applications
- *Fused measurement*: combination of active and passive
- *Bandwidth measurement*
- *Latency measurement and estimation*
- *Geolocation*

Bandwidth measurement

Size-delay method (`pathchar`)

Goal: to measure **link capacity**

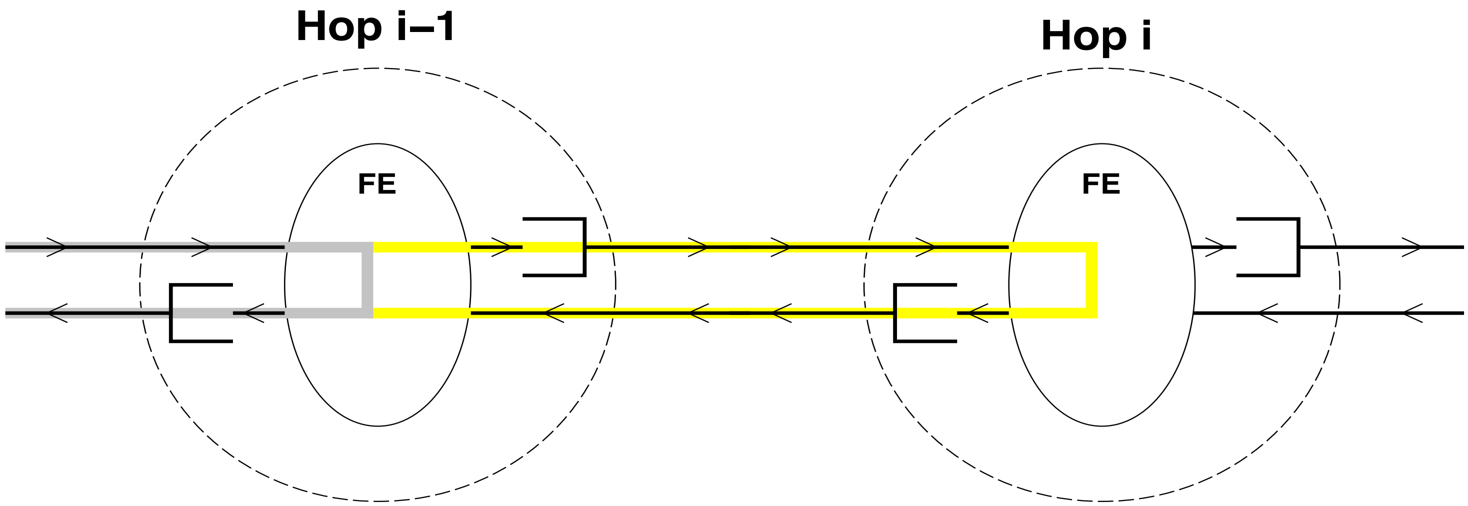
Idea: to extract information on capacity from the transmission time of a single packet

- In absence of cross traffic, the delay experienced as a packet passes over a link is affected by the packet’s size and the link’s capacity
- By varying packet size one can observe the effect on delay and infer the link’s capacity

Size-delay method

- The key element of the technique is to measure the RTT from the source to each hop of the path as a function of the probing size
- Mechanism similar to `traceroute`
  - TTL field of the IP header is used to force probing packets to expire at a particular hop
  - The router at that hop discards the probing packets, returning ICMP “Time exceeded” error messages back to the source
  - The source uses the received ICMP packets to measure the RTT to that hop

Size-delay method

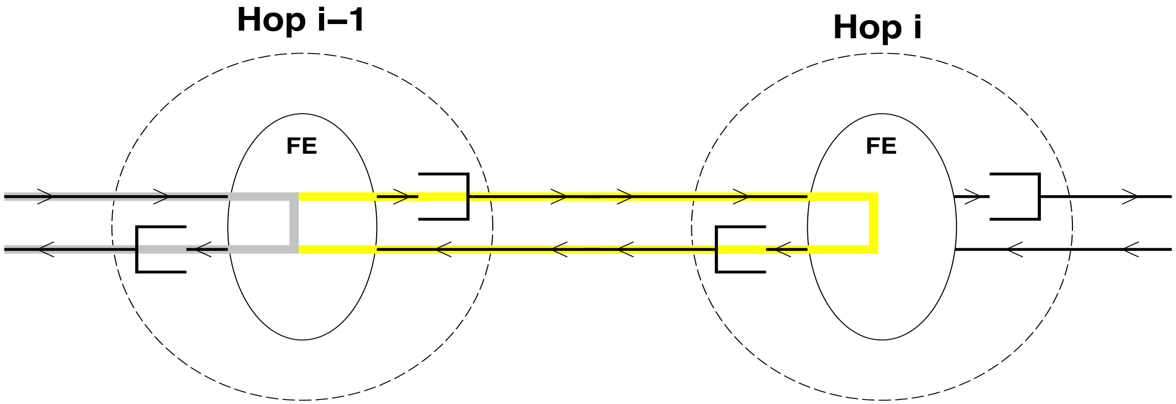


- At each hop the principal delays experienced by a probe packet are
  1. Queuing delay (dependent on packets ahead)
  2. Transmission delay (L/C)
  3. Propagation delay (d/v)

Dependent on packet size

Sapienza Università di Roma

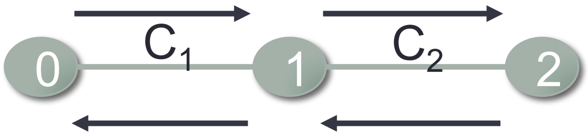
Size-delay method



- Round Trip Time at each hop consists of 3 delay components in the forward and reverse paths

$$\mathrm{RTT}\ (L) = \underbrace{R_{queue} + L/C_i + d/v}_{\text{forward}} + \underbrace{R_{queue} + ErrPkt/C_i + d/v}_{\text{reverse}}$$

Size-delay method



Over a path of two hops

$$RTT_2(L) = \cancel{R_q} + R_t + R_p + \cancel{R_q} + R_t + R_p + \cancel{R_q} + R_t + R_p + \cancel{R_q} + R_t + R_p$$

if there is no cross traffic

$$RTT_2(L) = R_t + R_p + R_t + R_p + R_t + R_p + R_t + R_p$$

$$= \boxed{2R_{p1} + 2R_{p2} + \frac{ICMP_{pkt}}{C_1} + \frac{ICMP_{pkt}}{C_2}} + \frac{L}{C_1} + \frac{L}{C_2}$$

$$RTT_2(L) = \alpha_2 + \beta_2 \cdot L$$ dove $$\beta_2 = \frac{1}{C_1} + \frac{1}{C_2} = \sum_{k=1}^{2} \frac{1}{C_k}$$

N.B.: All ICMP replies have the same size

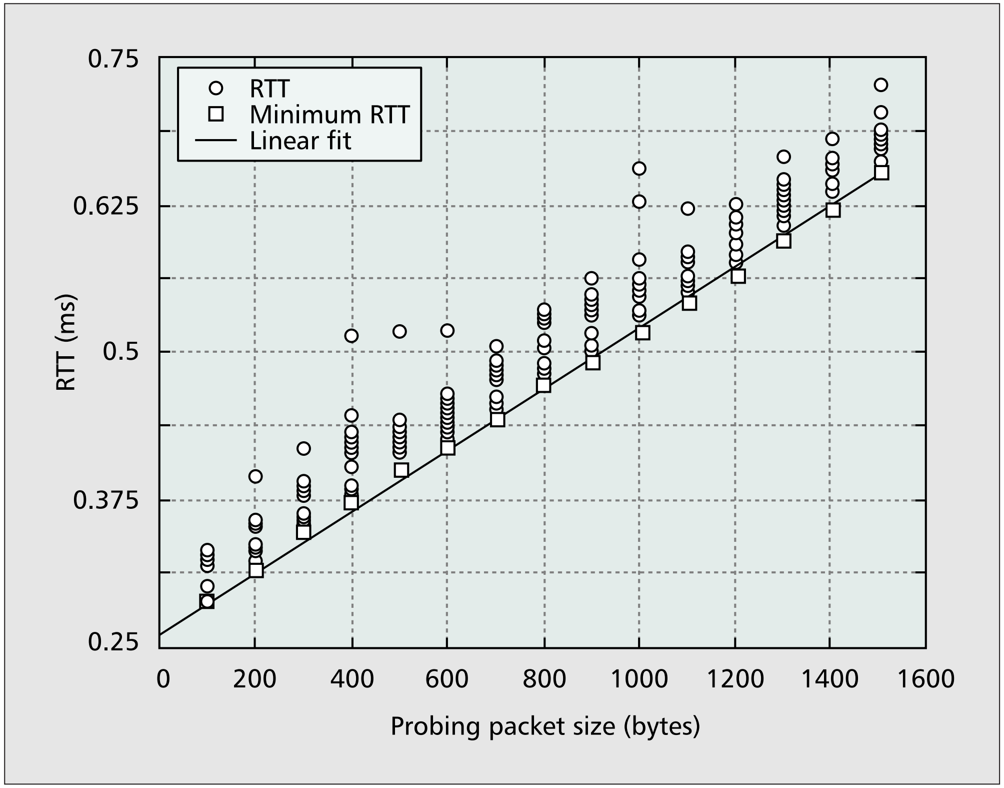
Size-delay method

Over H hops

$$RTT_i(L) = \alpha_i + \underbrace{\beta_i \cdot L} \qquad \text{dove} \qquad \beta_i = \sum_{k=1}^{H} \frac{1}{C_k}$$

Delay up to hop i
Independent of L

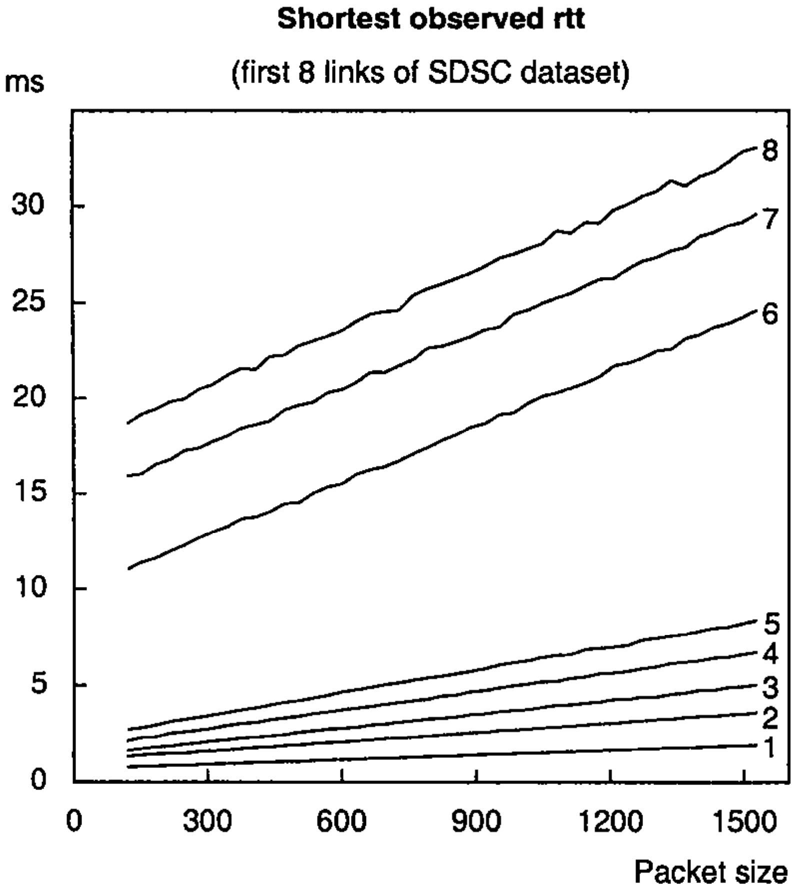
Proportional to
packet size L



The general approach is to send a number of packets with varying size L and estimate β_i (slope) from the resulting measurements

Size-delay method

Example of shortest RTT measured for the first 8 links of a path



Size-delay method

- Repeating the minimum RTT measurement for each hop *i=1,…,H* the capacity estimate at each hop *i* along the forward path is:

$$C_i = \frac{1}{\beta_i - \beta_{i-1}}$$

- In the 2-hop path example:

$$C_2 = \frac{1}{\beta_2 - \beta_1}$$

Open issues

- Accurate on short-paths. The minimum RTT reflects the absence of queuing at any hop. However, as the length of the path being probed grows, this assumption become more suspect, as it becomes harder for a packet to pass through many hops without experiencing queuing at any one of them
- Significant capacity underestimation errors if the measured path includes store-and-forward layer-2 switches, which introduces transmission delays but do not generate ICMP TTL-expired replies because they are visible at the IP layer

Latency measurement

Network latency

- An indicator of the performance that a network path may support

Metrics

- *Minimum RTT* :the most common metric for network latency is
    - It changes on relatively long timescale (only when topology and routing changes)
- *Instantaneous RTT*, which dynamically varies due to congestion
- *One-way delay* (little work has been done in this direction)

Challenges

- Immediate if `ping` is possible. Otherwise:
- **Proxy-based methods**
  - Neither of the path endpoints can participate in the measurement process
  - To estimate instantaneous RTT between endpoints, methods using additional *proxy hosts* have been developed. Proxies are capable of making measurements to nodes and to other proxies
- **Embedding-based methods**
  - The hosts involved are capable of making measurements but one does not want to measure each path of interest directly
  - Min RTT is the metric of interest
  - To avoid multiple measurements, each node is given a set of coordinates that are used to estimate latency between nodes
- In both cases we seek to estimate latency between a pair of nodes in the network without sending a probe between them

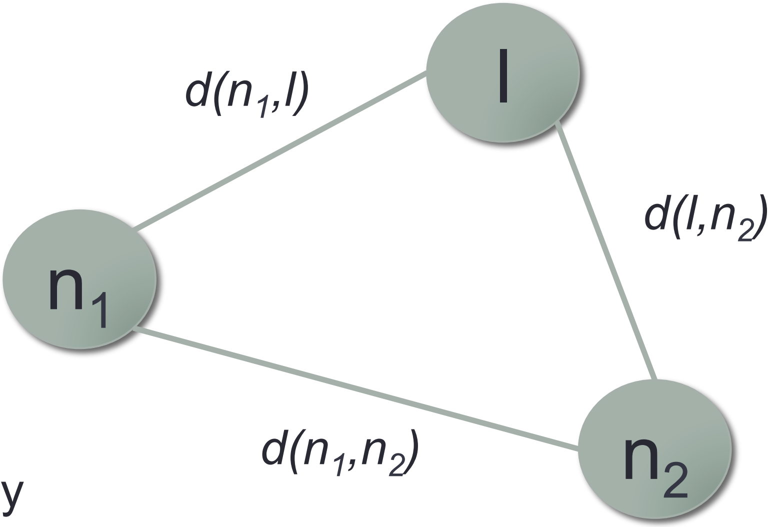
Proxy-based methods for estimation of current RTT

SAPIENZA
UNIVERSITÀ DI ROMA

Triangle inequality

- Assumption: shortest path routing -> validity of triangle inequality

$$d(n_1, n_2) \leq d(n_1, l) + d(l, n_2)$$



Distance function: latency

Method based on triangle inequality

- A set of proxies *{l_i}* is selected
- For two nodes n_1 and n_2, the triangle inequality requires that $d(n_1,n_2)$ is **bounded below** by

$$L = \max_i \left| d(n_1,l_i) - d(n_2,l_i) \right|$$

- And is **bounded above** by

$$U = \min_i \left| d(n_1,l_i) + d(n_2,l_i) \right|$$

- **Weighted averages of L and U** can be used as estimates of $d(n_1,n_2)$

IDMaps

- IDMaps assumes the availability of particular proxies, called tracers
- The latency between nodes n_1 and n_2 is estimated as the latency between n_1 and its nearest tracer, plus n_2 and its nearest tracer, plus the measured latency between the two tracers
- The system also uses a collection of servers that respond to client queries and return network latency estimates
- Accuracy: limited when one or both nodes are far from the nearest tracers

King

- Tool that addresses some drawbacks of IDMaps by exploiting the DNS system
- Rather than relying on specially deployed tracers, King uses a node’s local DNS server as its measurement proxy
- Accurate: King is capable of generating estimates that are very close to the true path latencies
- Its estimates are based on direct, online measurement
- The measured end hosts do not need to cooperate
- Fast and lightweight: King requires the generation of only a few packets to produce an estimate
- King makes use of the existing DNS infrastructure in a novel manner

King

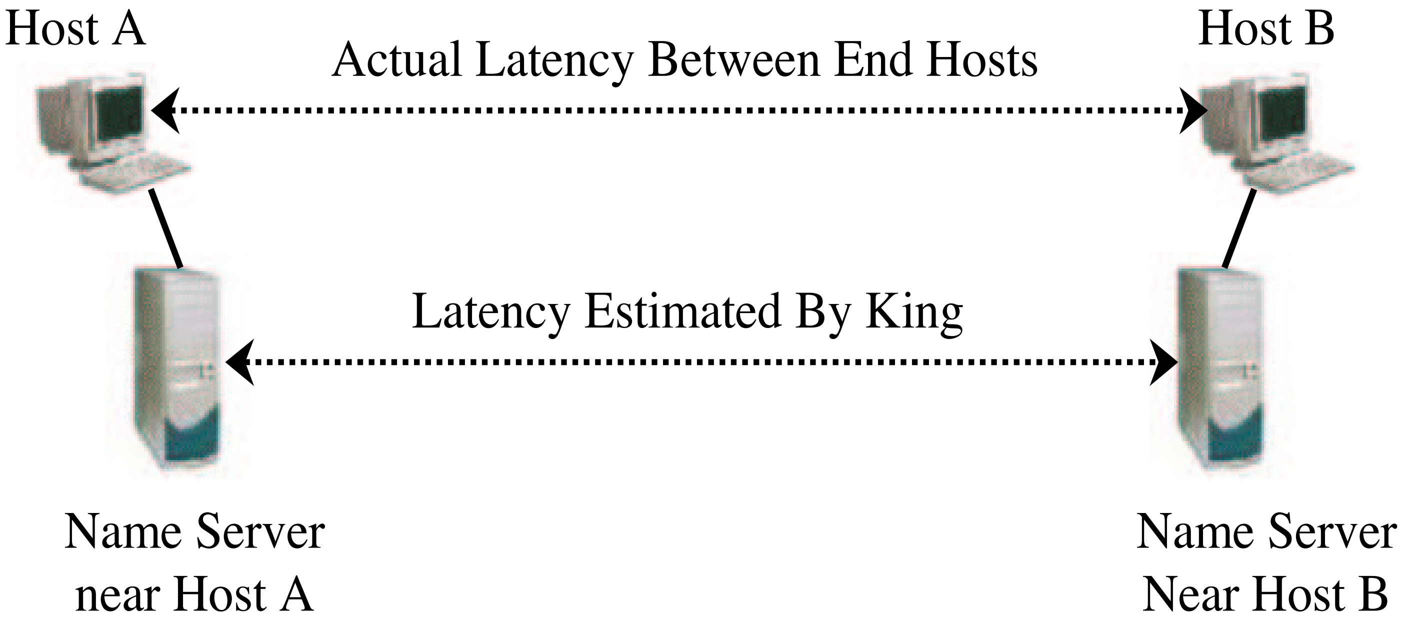
The method used in King is based on two **observations**

1. Given a pair of end hosts to be measured, in most cases it is possible for King to find DNS name server that are topologically close to the end hosts
2. Given a pair of DNS name servers, King can accurately estimate the latency between them using recursive DNS queries

King is able to use the measured latency between name servers as an estimate of the latency between end hosts

King

King estimates the latency between two end hosts by measuring the latency between nearby DNS name servers



King

- Observations:

1. Most end hosts in the Internet are located close to their DNS name servers
2. Recursive queries can be used to measure the latency between pairs of DNS servers

- Procedure:

1. Locating nearby name servers
2. Measuring the latency between them

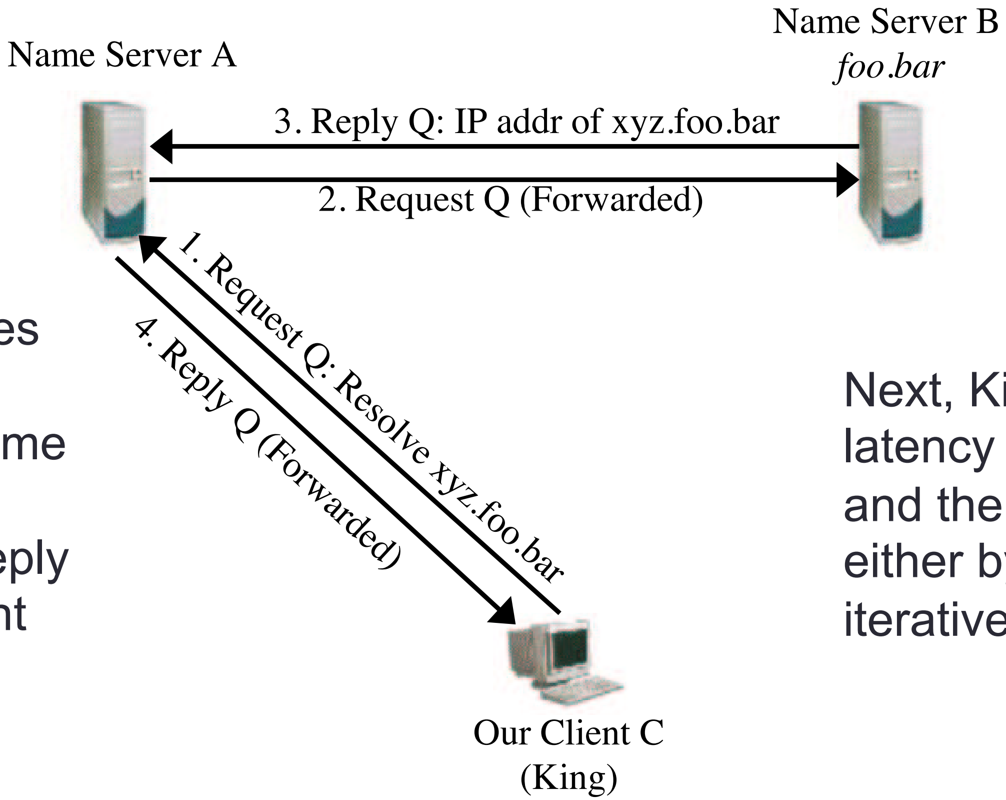
King: locating nearby name servers

- It is a fairly common practice in the Internet to collocate authoritative name server for a domain close to the hosts in that domain
- Authoritative name servers for a domain can be found by querying the DNS system for name server records associated with the domain name

  Es. (`nslookup -type=NS uniroma1.it`)
- N.B. there might be multiple authoritative name servers for a domain close the host in that domain (some may be far away)

King: measuring latency

To measure the latency between two name serves, King issues a recursive DNS query to one name server, requesting it to resolve a name belonging to a domain for which the other server is authoritative



Name Server A

Name Server B
foo.bar

3. Reply Q: IP addr of xyz.foo.bar

2. Request Q (Forwarded)

1. Request Q: Resolve xyz.foo.bar

4. Reply Q (Forwarded)

Our Client C
(King)

Server A resolves the query by interrogating name server B, and forwarding its reply back to the client

B is authoritative for the domain foo.bar

Next, King measures the latency between the client and the name server A, either by Ping or issuing an iterative DNS query

King: measuring latency

- Assumption: name server A directly contacts name server B instead of having to traverse the DNS hierarchy
- To ensure this: before measurement name server A issues a recursive query for *foo.bar* so that A caches the fact that B is authoritative for *foo.bar*
- For accuracy: King can be configured to measure latency multiple times
  - as after the first query A caches the reply form B, each query has the form *randomnumber .foo.bar*, where *randomnumber* is a large random number
  - It is very likely that this query is not a valid name in the *foo.bar* domain and in such a case, name server B would reply with a “NXDOMAIN” message

Embedding-based methods for estimation of minimum RTT

Embedding-based methods

- Embedding approach assigns to each node a location in an abstract, high dimensional Euclidean space.
- A node’s location can be fixed using a set of measurements to landmarks
- Only the landmarks need to perform all-pairs latency measurements

GNP

- At initialization , the *N* landmarks $l_1, l_2, \ldots, l_N$ performs all pair latency measurements yielding the set of measured latencies $d(l_i, l_j)$
- Each landmark l_i is then assigned a coordinate vector $\vec{x}_i \in \Re^r$
- This assignment is obtained by minimization of the objective function

$$f_l(\vec{x}_1, \ldots, \vec{x}_N) = \sum_{i,j \in 1,\ldots,N} err(d(l_i, l_j), \left\| \vec{x}_i - \vec{x}_j \right\|_2)$$

- Where *err(a,b)* is typically the simple squared error

$$err(a,b) = (a-b)^2$$

GNP

- After initialization, each landmark has a coordinate vector
- Each node then measures latencies to all N landmarks
- Node *n*$_i$ find its own coordinate vector *x*$_i$ by minimizing a similar objective function

$$f_n(\vec{x}_i) = \sum_{j \in 1,\ldots,N} err(d(n_i, l_j), \left\| x_i - x_j \right\|_2)$$

- After each node has a coordinate vector, the latency between node *n*$_i$ and *n*$_j$ can be estimated without any additional measurement
- The estimate is:

$$d(n_i, n_j) \approx \left\| \vec{x}_i - \vec{x}j \right\|_2$$

Geolocation

- Finding the geographic location of network elements can be useful for a wide variety of social, economic, and engineering purposes
- Geolocation problem: given the network address od a target host, what is the host's geographic location?
- Approaches:
  - Name-based geolocation
  - Delay-based geolocation

State of the art

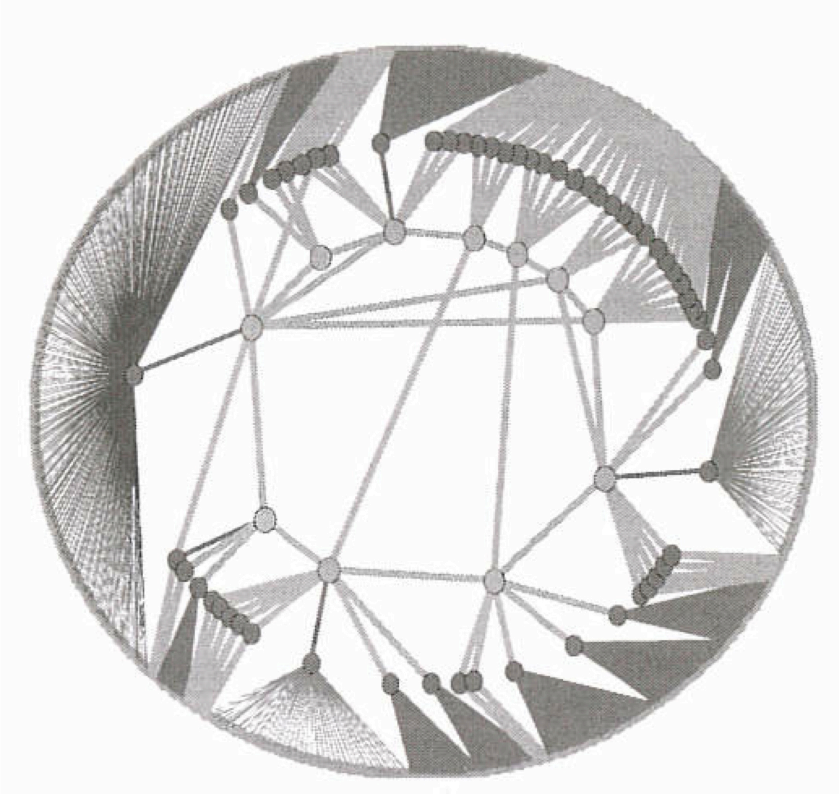
Equipment properties

- Routing delays: the delay a packet experiences as it passes through a router is very small
  - Minimum delay can be 20 microseconds (same interface)
  - Tens of milliseconds (different interfaces)
  - If router heavily loaded, queuing delays can be of the order of tens of milliseconds
- Processing (how routers produce and consume traffic)
  - OSPF link state announcements require on the order of 100 microseconds to be processed (much of the time due to data copying within the router)
- Middleboxes (NATs and firewalls) introduces delays ranging form one milliseconds to hundreds of milliseconds when forwarding packets

Topology properties

AS graph show high variability in degree distribution

Router graph also show high variability in degree distribution: the total bandwidth of a router generally declines as the number of links connected increases. In the core high bandwidth is essential and routers tend to low degree. At network edges the need to serve many users with relatively low bandwidth connections leads to routers with higher degree



Example synthetic graph meeting technological constraint for routers

Topology properties

Path properties:

- typical paths through the router graph tend to be short. Measures of the number of IP hops between nodes in the Internet show average values around 16: paths of more than 30 hops are rare
- Despite the fact that path lengths in the Internet are relatively short, measurements show that they are often longer than necessary (longer than shortest path in terms of number of IP hops), due mainly to AS-AS peering policies and inter-domain policies

Interaction of traffic and network

Packet delay distributions show high variability

- It is useful to divide the various sources of packet delay into two types:

1. Deterministic (includes transmission and propagation delay)
   - Transmission delay in Internet is only significant on slow access links
     - Backbone: 1.2 microseconds
     - 56Kbps dialup: over 200 microseconds
   - Propagation delay depends on geographic distance
2. Stochastic (queuing delay)
   - Varies from hundreds of milliseconds to tens of seconds