Tomography-based Overlay Network Monitoring

Yan Chen, David Bindel, and Randy H. Katz

UC Berkeley

Motivation

- Infrastructure ossification led to thrust of overlay and P2P applications
- Such applications flexible on paths and targets, thus can benefit from E2E distance monitoring
  - Overlay routing/location
  - VPN management/provisioning
  - Service redirection/placement …
- Requirements for E2E monitoring system
  - Scalable & efficient: small amount of probing traffic
  - Accurate: capture congestion/failures
  - Incrementally deployable
  - Easy to use

Existing Work

- General Metrics: RON (n^2 measurement)
- Latency Estimation
  - Clustering-based: IDMaps, Internet Isobar, etc.
  - Coordinate-based: GNP, ICS, Virtual Landmarks

- Network tomography
  - Focusing on inferring the characteristics of physical links rather than E2E paths
  - Limited measurements -> under-constrained system, unidentifiable links

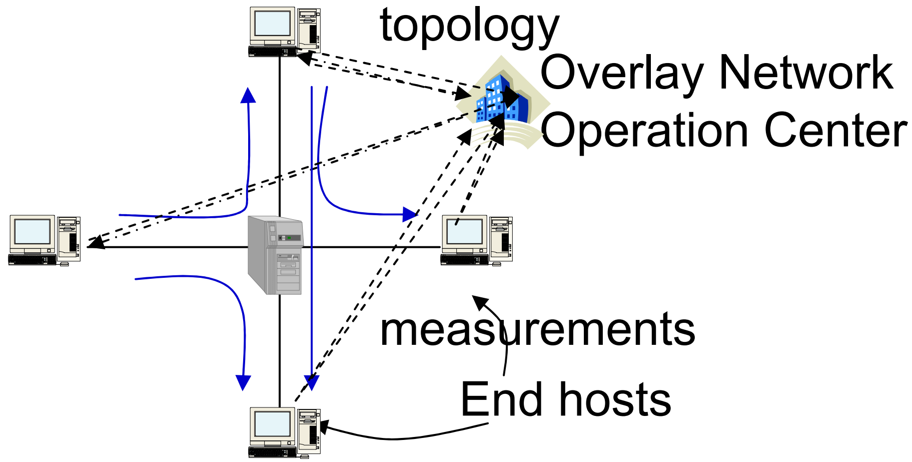
Problem Formulation

Given an overlay of *n* end hosts and O(n^2) paths, how to select a minimal subset of paths to monitor so that the loss rates/latency of all other paths can be inferred.

Assumptions:

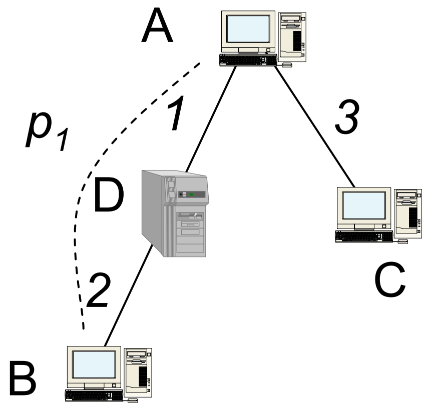
- Topology measurable
- Can only measure the E2E path, **not** the link

Our Approach



Select a basis set of k paths that fully describe $O(n^2)$ paths ($k \ll O(n^2)$)

- Monitor the loss rates of k paths, and infer the loss rates of all other paths
- Applicable for any additive metrics, like latency



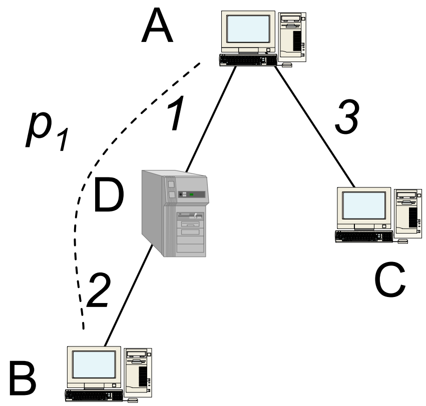
Modeling of Path Space

Path loss rate *p*, link loss rate *l*

$$1 - p_1 = (1 - l_1)(1 - l_2)$$

$$\log(1 - p_1) = \log(1 - l_1) + \log(1 - l_2) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \log(1 - l_1) \\ \log(1 - l_2) \\ \log(1 - l_3) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = b_1$$

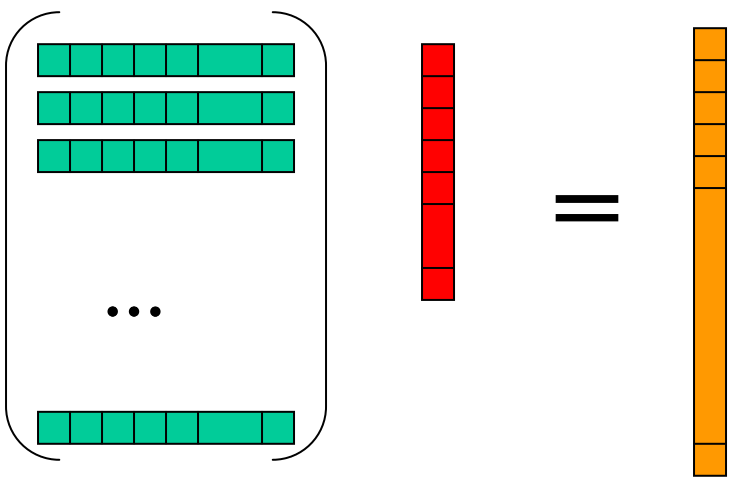


Putting All Paths Together

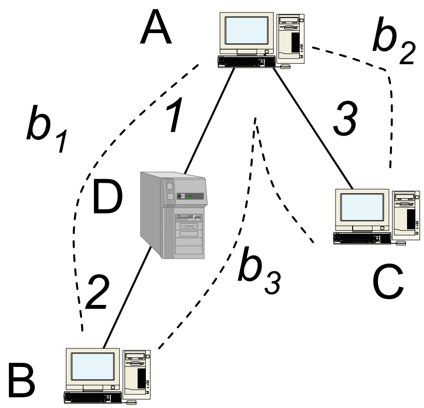
Totally $r = O(n^2)$ paths, s links, $s \ll r$

$$Gx = b, \text{ where path matrix } G \in \{0 \mid 1\}^{r \times s}$$

link loss rate vector $x \in \Re^{s \times 1}$, path loss rate vector $b \in \Re^{r \times 1}$

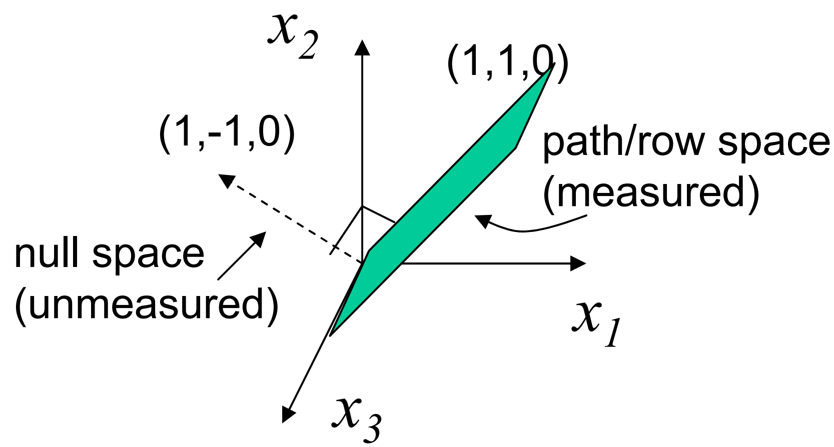


Sample Path Matrix



$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



- $x_1 - x_2$ unknown => cannot compute x_1, x_2
- Set of vectors $\alpha[1 \quad -1 \quad 0]T$ form null space
- To separate identifiable vs. unidentifiable components: $x = x_G + x_N$

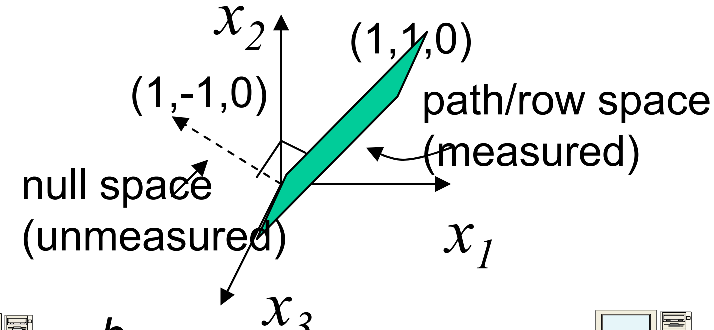
$$x_G = \frac{(x_1 + x_2)}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b_1/2 \\ b_1/2 \\ b_2 \end{bmatrix}$$

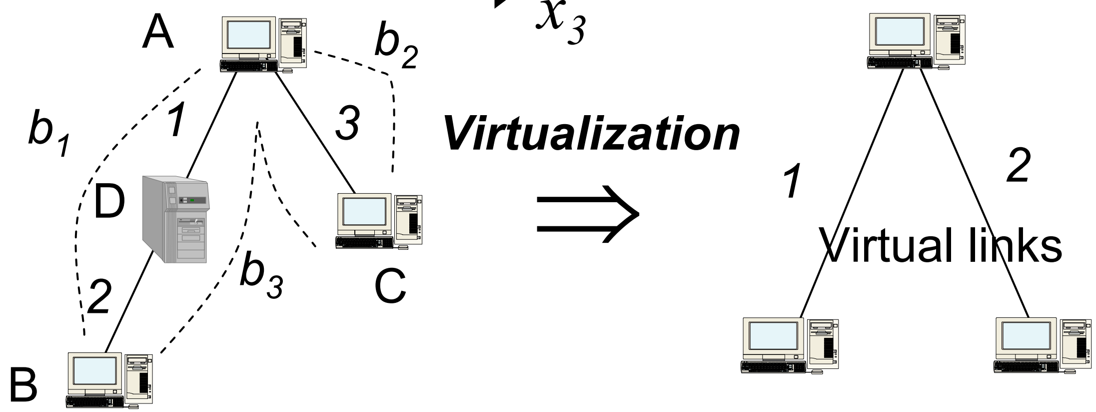
$$x_N = \frac{(x_1 - x_2)}{2} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Intuition through Topology Virtualization

Virtual links:

- Minimal path segments whose loss rates uniquely identified
- Can fully describe all paths
- x_G is composed of virtual links





$$x_G = \frac{(x_1+x_2)}{2}\begin{bmatrix}1\\1\\0\end{bmatrix} + x_3\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}b_1/2\\b_1/2\\b_2\end{bmatrix}$$

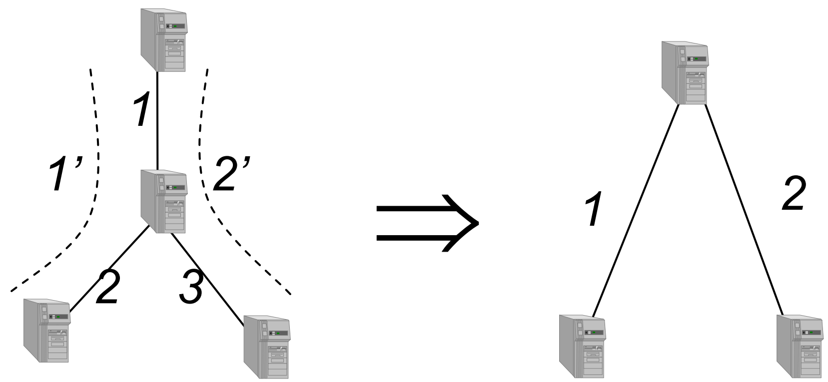
All E2E paths are in path space, *i.e.*, $Gx_N = 0$

$$b = Gx = Gx_G + Gx_N = Gx_G$$

More Examples

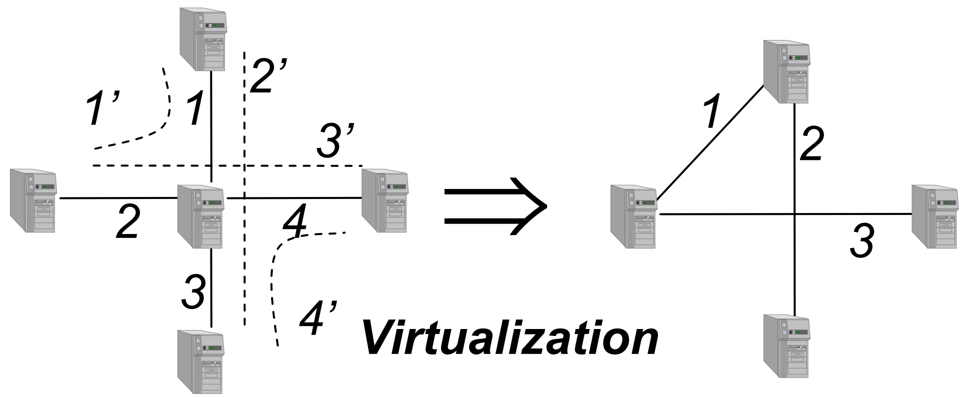
$$G = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Rank(G)=2



$$G = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Rank(G)=3

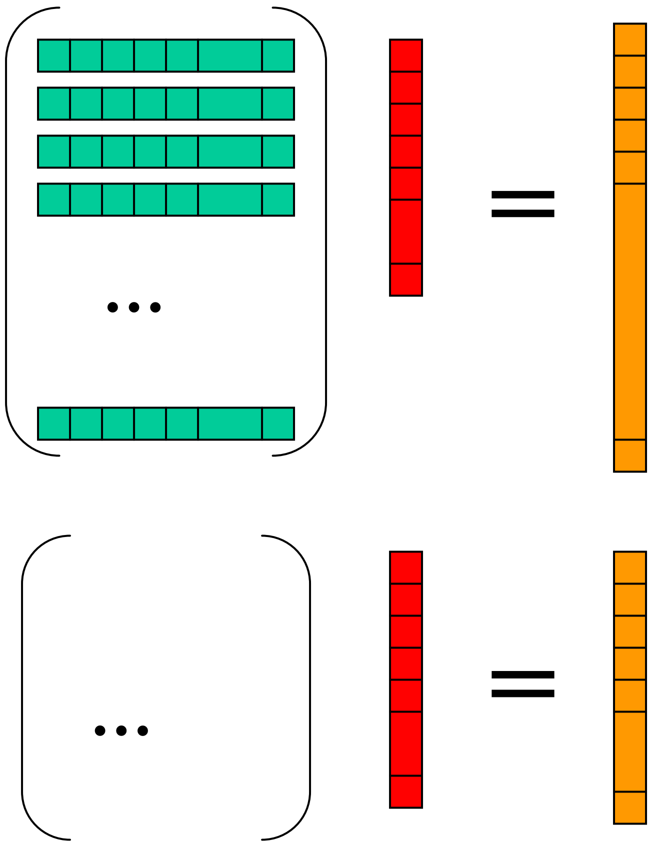


Real links (solid) and all of the overlay paths (dotted) traversing them

Virtual links

Algorithms

$$Gx_G = b$$



- Select k = rank(G) linearly independent paths to monitor
  - Use QR decomposition
  - Leverage sparse matrix: time $O(rk^2)$ and memory $O(k^2)$
    - E.g., 10 minutes for n = 350 (r = 61075) and k = 2958
- Compute the loss rates of other paths
  - Time $O(k^2)$ and memory $O(k^2)$

How many measurements saved ?

$k \ll O(n^2)$?

For a power-law Internet topology

- When the majority of end hosts are on the overlay

  $\mathbf{k = O(n)}$ (with proof)

- When a small portion of end hosts are on overlay
  - If Internet a pure hierarchical structure (tree): $k = O(n)$
  - If Internet no hierarchy at all (worst case, clique): $k = O(n^2)$
  - Internet has moderate hierarchical structure [TGJ+02]

For reasonably large n, (e.g., 100), $k = O(n\log n)$

(extensive linear regression tests on both synthetic and real topologies)

Practical Issues

- Topology measurement errors tolerance
- Measurement load balancing on end hosts
  - Randomized algorithm
- Adaptive to topology changes
  - Add/remove end hosts and routing changes
  - Efficient algorithms for incrementally update of selected paths

Evaluation

- Extensive Simulations
- Experiments on PlanetLab
  - 51 hosts, each from different organizations
  - 51 × 50 = 2,550 paths
  - On average $k = 872$
- Results Highlight
  - Avg real loss rate: 0.023
  - Absolute error mean: 0.0027 90% < 0.014
  - Relative error mean: 1.1 90% < 2.0
  - On average 248 out of 2550 paths have no or incomplete routing information
  - No router aliases resolved

Areas and Domains			**# of hosts**
US (40)	.edu		33
	.org		3
	.net		2
	.gov		1
	.us		1
International (11)	Europe (6)	France	1
		Sweden	1
		Denmark	1
		Germany	1
		UK	2
	Asia (2)	Taiwan	1
		Hong Kong	1
	Canada		2
	Australia		1

Conclusions

- A tomography-based overlay network monitoring system
  - Given n end hosts, characterize $O(n^2)$ paths with a basis set of $O(n \log n)$ paths
  - Selectively monitor the basis set for their loss rates, then infer the loss rates of all other paths
- Both simulation and PlanetLab experiments show promising results