Directed Probing for Efficient and Accurate Active Measurements

Robert Beverly Arthur Berger[1]

Naval Postgraduate School
[1]MIT CSAIL
rbeverly@nps.edu, awberger@csail.mit.edu

February 8, 2010

AIMS-2 - Workshop on Active Internet Measurements

Outline

Internet Topology Measurement

The Internet is:

1. Large, and complex
2. Poorly instrumented

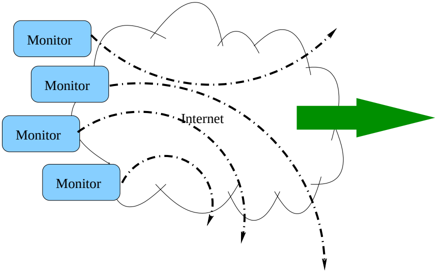
$\Rightarrow$ Poorly understood topology

Internet Topology – why do we care?

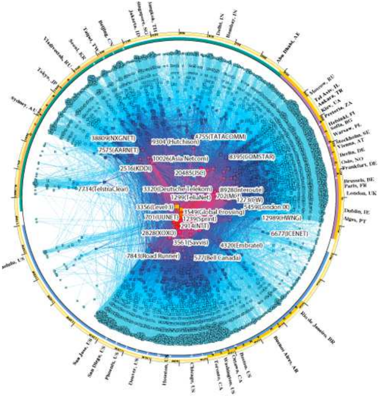
- Critical infrastructure protection
- Network modeling, routing research, protocol validation, etc.
- Future Internet architectures, Internet evolution, etc.

State of the Art

Measure from available vantage points...



Infer structure...



Problem

Internet Topology Measurement

- What we have:
  - Handful of monitoring points from which to run path probes
  - Requires significant time and resources to probe all IPv4 destinations
  - Attempt to balance load vs. measurement cycle time
- What we want:
  - Many vantage points
  - High frequency scanning
  - But, with low-load
  - Coordination between vantage points?

Problem

Hypothesis:

By leveraging network priors (knowledge of routing, structure, etc.) and adaptive sampling (progressively learned knowledge), we can:

- Significantly lower probing load
- Without sacrificing measurement fidelity
- (and perhaps increase fidelity)

Intuition

Scaling:

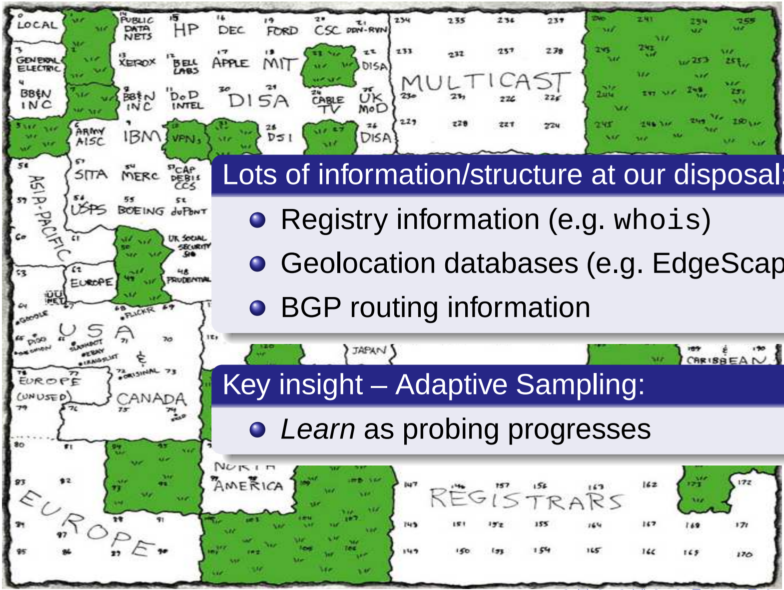
- $\sim 2^{32-1}$ possible destinations (2.9B from Jan 2010 routeviews)
- But, because of hierarchy and aggregation and classful history, practitioners often aggregate measurements into /24's
- 2^{24-1} destinations much more manageable – but, right granularity?

Example:

- Necessary to probe all 2^{16} /24's in `18.0.0.0/8` to ascertain path characteristics or latency?

This work investigates how we can use network priors to "intelligently" drive probing for more efficient and accurate topology measurements

Network Priors (xkcd insight...)



Lots of information/structure at our disposal:

- Registry information (e.g. `whois`)
- Geolocation databases (e.g. EdgeScape)
- BGP routing information

Key insight – Adaptive Sampling:

- *Learn* as probing progresses

Archipelago

Investigate hypothesis using CAIDA's Ark as case study:

- Distributed "team probing," $\sim$ 41 monitors
- All routed addresses divided into /24's; partitioned across monitors
- From each /24, a single address is selected at random to probe
- Probe == traceroute^{++}; record router interfaces on forward path
- Uses `scamper` (cf. Luckie) for constant load
- A "cycle" == traceroutes to all routed /24's

caida

WIP Caveats

Work in Progress – At this stage:

- Deconstruct probing process of Ark as case study
- Use BGP information from routeviews as decision prior
- Looking at router-level topology, not organization or AS
- Not yet incorporating any alias resolution

Not making claims about topological correctness; investigate ability to reproduce baseline more efficiently

Outline

Data Set

First, let's deconstruct Ark cycle:

- Before developing our new technique (next), understand data
- Start with a *single* vantage point, AMW-US
- Data from this node for a cycle on January 11, 2010
- Represents:
  - 263K traceroutes
  - 55K distinct BGP prefixes
  - $\sim$ 4.4M probe packets

Q: What do we learn?

Edit Distance

Meta-Question: What's the information gain of successive traceroutes?

Q1: *How similar are traceroutes to the same destination BGP prefix?*

- Use Levenshtein "edit" distance DP algorithm
- Determine the minimum number of edits (insert, delete, substitute) to transform one string into another
- e.g. "`robert`" $\rightarrow$ "`robber`" = 2

- We use: $\Sigma = \{0, 1, \ldots, 2^{32} - 1\}$
- Each unsigned 32-bit IP address along traceroute paths $\in \Sigma$

ED=2

```
129.186.6.251 129.186.254.131 192.245.179.52 4.53.34.13
129.186.6.251 192.245.179.52 4.69.145.12
```

Edit Distance

Meta-Question: What's the information gain of successive traceroutes?

Q1: *How similar are traceroutes to the same destination BGP prefix?*

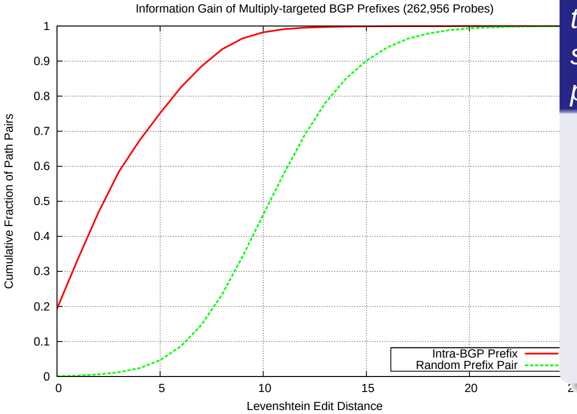
- Use Levenshtein "edit" distance DP algorithm
- Determine the minimum number of edits (insert, delete, substitute) to transform one string into another
- e.g. "`robert`" $\rightarrow$ "`robber`" = 2

- We use: $\Sigma = \{0, 1, \ldots, 2^{32} - 1\}$
- Each unsigned 32-bit IP address along traceroute paths $\in \Sigma$

ED=2

```
129.186.6.251 129.186.254.131 192.245.179.52 4.53.34.13
129.186.6.251 192.245.179.52 4.69.145.12
```

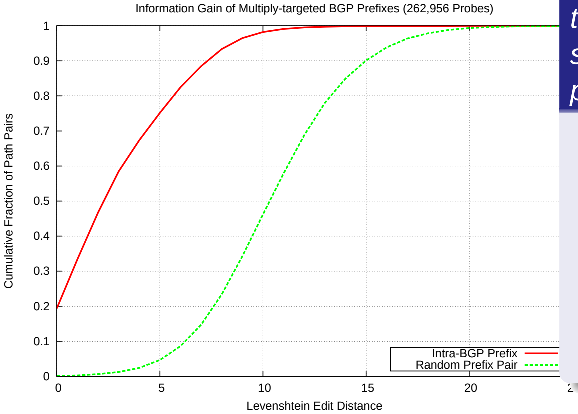
Edit Distance



Q1: *How similar are traceroutes to the same destination BGP prefix?*

- ∼60% of traces to destinations in same BGP prefix have $ED \leq 3$
- Fewer than 50% of random traces have $ED \leq 10$

Edit Distance



Q1: *How similar are traceroutes to the same destination BGP prefix?*

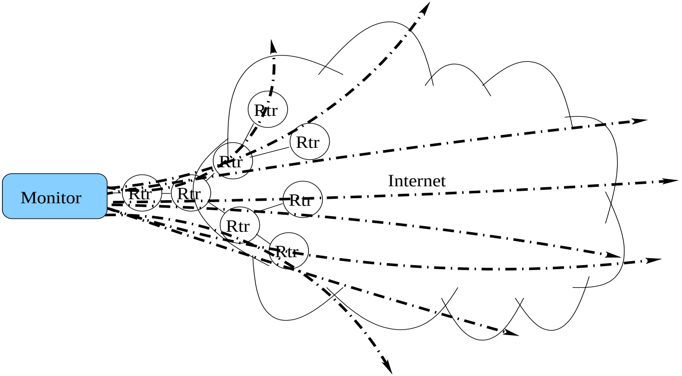
- $\sim$60% of traces to destinations in same BGP prefix have $ED \leq 3$
- Fewer than 50% of random traces have $ED \leq 10$

Confirms our intuition

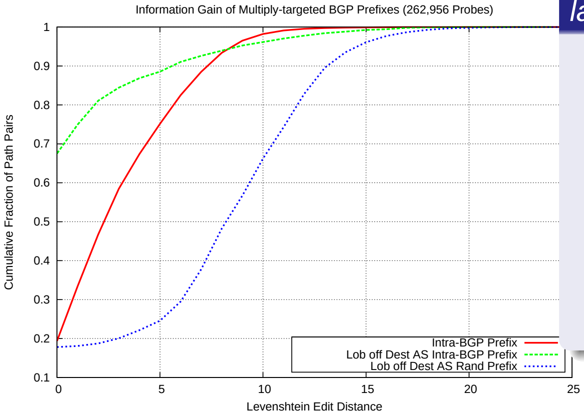
Edit Distance

Q2: *How much path variance is due to the last-hop AS?*

- Intuitively, number of potential paths exponential in the depth
- More information gain at the end of the traceroute?



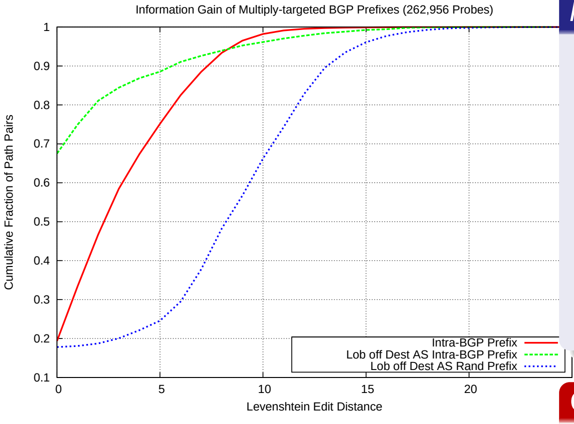
Edit Distance



Q2: *How much path variance is due to the last-hop AS?*

- Lob off last AS
- Answer: lots!
- For $\sim$ 70% of probes to same prefix, we get no additional information beyond leaf AS

Edit Distance



Q2: *How much path variance is due to the last-hop AS?*

- Lob off last AS
- Answer: lots!
- For ∼ 70% of probes to same prefix, we get no additional information beyond leaf AS

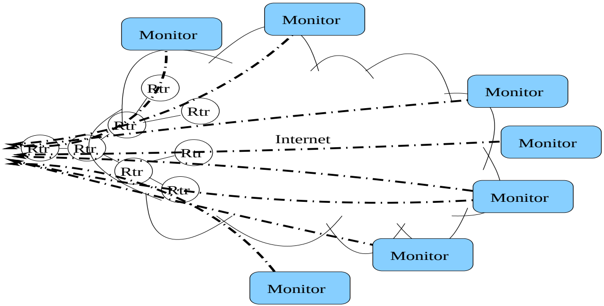
Conclusion 1:

Significant *packet* savings possible

Multiple Vantage Points

Q3: *How much information gain do multiple vantage points yield?*

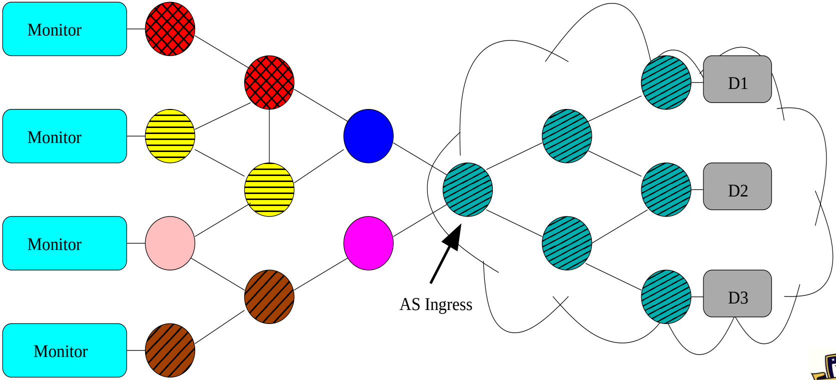
- Intuitively, expect traceroute "tail" to be similar
- Majority of information gain in first half of trace?



Multiple Vantage Points

Q3: *How much information gain do multiple vantage points yield?*

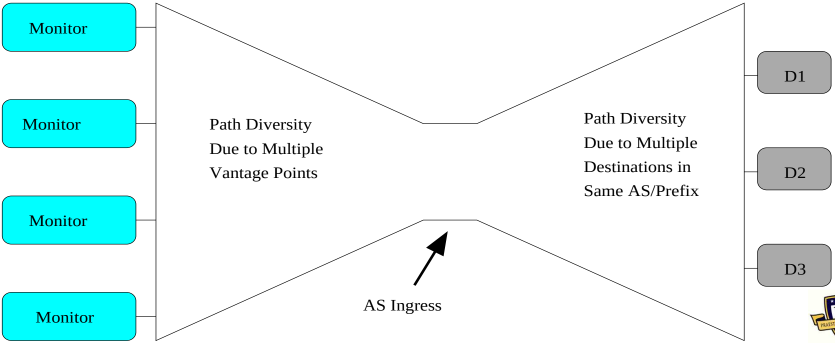
- Information gain is at *both* tails



Multiple Vantage Points

Q3: *How much information gain do multiple vantage points yield?*

- Information gain is at *both* tails
- The "hourglass effect" – what's the commonality of the "narrow waist?"



Multiple Vantage Points

Q3: *How much information gain do multiple vantage points yield?*

- Want to understand “waist commonality”
- Exclude end of the tail (per previous results)
- Reverse align (tail commonality)
- Measure reverse longest common subsequence (and ED)

For example...

Waist Commonality (ex. 1)

Two vantage points, different dsts in same prefix, WC=10

[tr: 0] [dst: 44.148.217.39][asn: 7377] 129.186.6.251 129.186.254.131 192.245.179.52 164.113.238.213 164.113.238.193 64.57.28.57 64.57.28.44 137.164.26.145 137.164.26.246 137.164.46.103 137.164.46.7 137.164.24.178 132.239.255.129 132.239.255.84 132.239.255.42 169.228.66.251

[tr: 1][dst: 44.107.75.47][asn: 7377] 84.88.81.121 84.88.19.149 130.206.202.29 130.206.250.25 130.206.250.2 62.40.124.53 62.40.112.25 62.40.112.22 62.40.125.18 64.57.28.6 64.57.28.43 64.57.28.44 137.164.26.145 137.164.26.246 137.164.46.103 137.164.46.7 137.164.24.178 132.239.255.129 132.239.255.84 132.239.255.42 169.228.66.251

Waist Commonality (ex. 2)

Two vantage points, different dsts in same prefix, WC=2

```
[tr:  0] [dst:  114.182.222.103][asn:  4713]
129.186.6.251 129.186.254.131 192.245.179.52 4.53.34.13
4.69.135.233 4.69.135.230 4.69.145.12 4.68.63.226
129.250.2.173 129.250.4.25 129.250.5.82 129.250.11.54
122.28.104.181 118.23.146.50 218.43.251.130
219.167.250.62 118.21.197.34 118.21.194.43
```

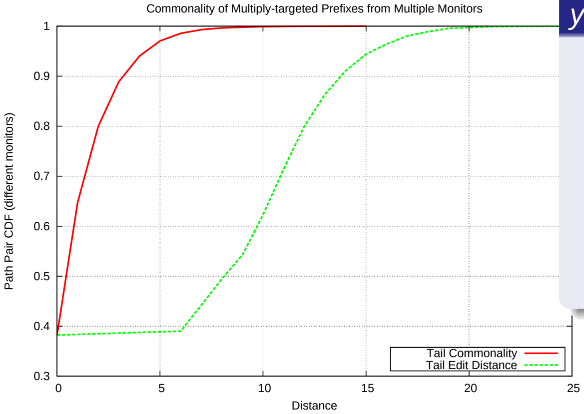
```
[tr:  1] [dst:  114.166.196.77][asn:  4713] 84.88.81.121
84.88.19.149 130.206.202.29 130.206.250.25 162.97.119.17
208.50.13.146 129.250.5.237 129.250.5.35 129.250.4.209
129.250.3.210 129.250.11.54 122.28.104.181 118.23.168.13
122.28.168.42 118.23.96.18 118.23.99.71
```

Multiple Vantage Points

Q3: *How much information gain do multiple vantage points yield?*

- Add new Ark vantage point, BCN-ES into the analysis...

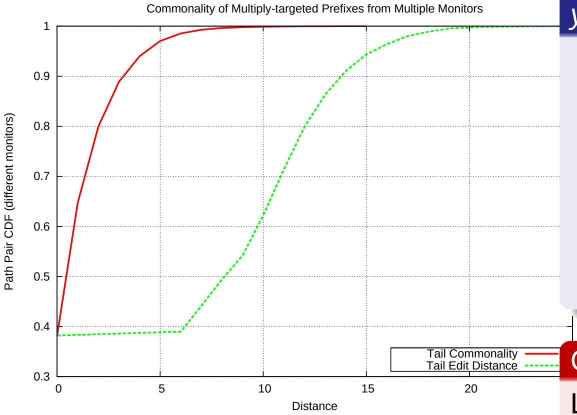
Multiple Vantage Points



Q3: *How much gain do multiple vantage points yield?*

- In ~ 30% of the cases, <u>all new</u> information
- Only ~ 10% of probes yield more than 4 duplicate hops

Multiple Vantage Points



Q3: *How much gain do multiple vantage points yield?*

- In ∼ 30% of the cases, <u>all new</u> information
- Only ∼ 10% of probes yield more than 4 duplicate hops

Conclusion 2:

Lots of information gained from multiple vantage points

Outline

Simulation-Driven Probing

Based on results from data analysis...

Strategy:

- Similar idea to adaptive sampling methods
  - e.g. sequential analysis for rare events (oil ground samples)
  - Active learning
- Given samples thus far,
  - How many to sample next?
  - Which ones to sample next?
- $P(s|\hat{y})$ for $\hat{y}$ already observed

Simulation-driven Probing

Methodology:

- We simulate adaptive sampling by selectively withholding points in the Ark traces given traces observed thus far
- Compare topology resulting from complete Ark traceroute cycle against a simulated cycle
- Evaluate metrics:
  1. Probing cost (packets, traces, etc)
  2. Model fidelity (graph theoretic properties)

Model Metrics

Simple Metrics to Compare G, G':

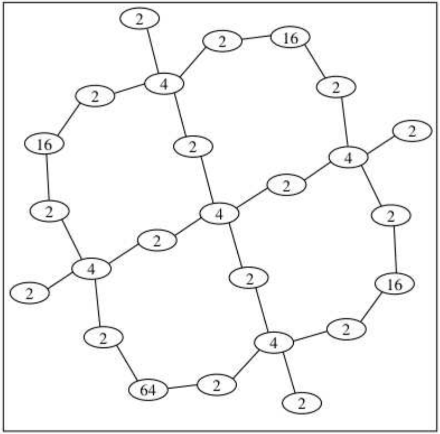
- Number of vertices, edges
- Graph diameter
- Degree distribution
- But, what topology / process generated this degree distribution?

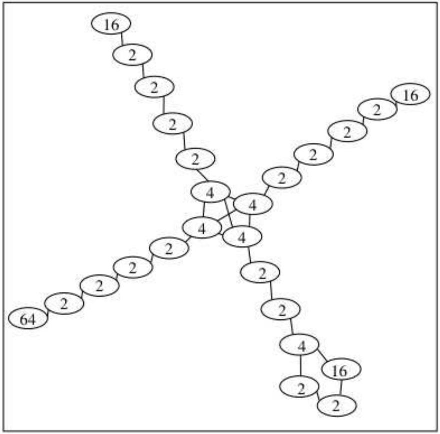
Typically not enough to understand graph.

Understanding Graphs

David Alderson (NPS OR):

Two graphs with same degree distribution:

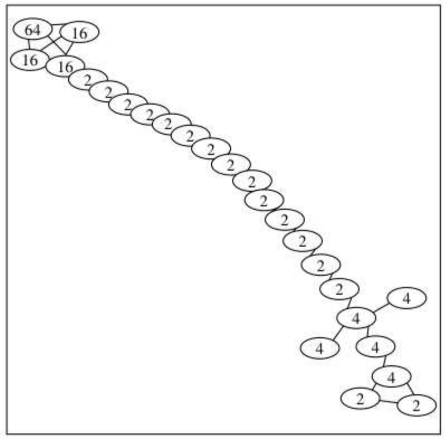


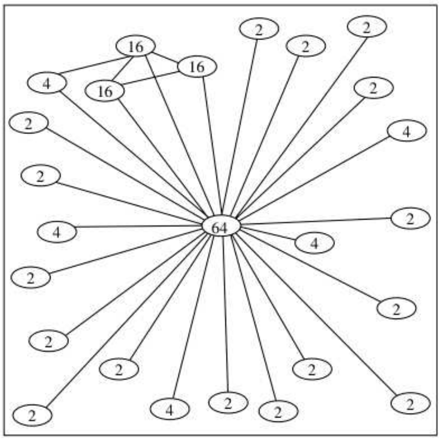


Understanding Graphs

David Alderson (NPS OR):

And two more, same degree distribution:





Model Metrics

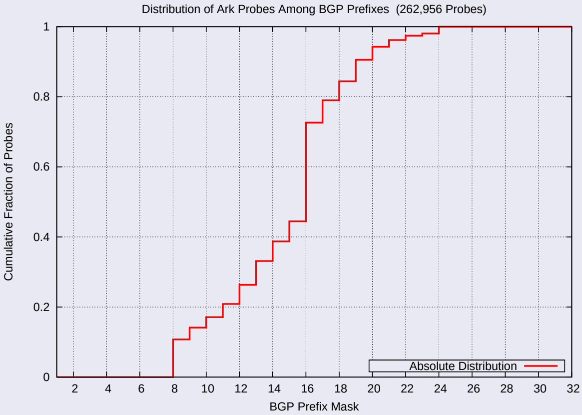
Metrics to Compare G, G':

- ***Expansion:*** $E(h)$ = avg fraction of nodes in G that fall within a radius h (reachable set)
- ***Resilience:*** Minimum number of cuts to achieve bi-partition (NP-hard)
- ***Distortion:*** For the SPT on G, distance between vertices sharing an edge if forced to use the SPT
- ***Spectral Properties:*** e.g. eigendecomposition, random walk
- ***Likelihood:*** High-degree nodes connected to high-degree nodes (scale-free, hub-like)?

$$L(g) = \sum_{(i,j)\in E(g)} \omega_i \omega_j$$

Adaptive Sampling

- Distribution of Ark traceroute probes to the size of the BGP prefix of the traceroute destination

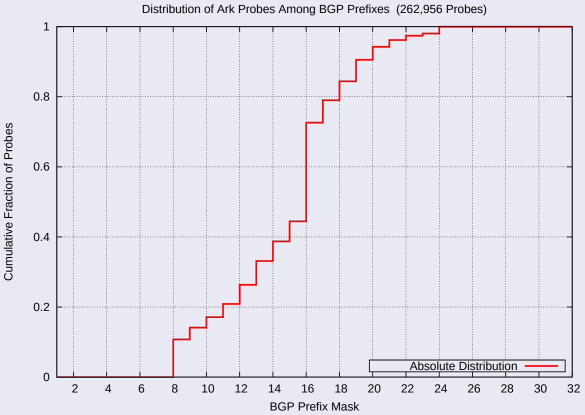


naïve Strategy:

- Litmus test, how well do we do by probing only *one* point in each BGP prefix?
- Significant reduction in probing load
- Model fidelity?

Adaptive Sampling

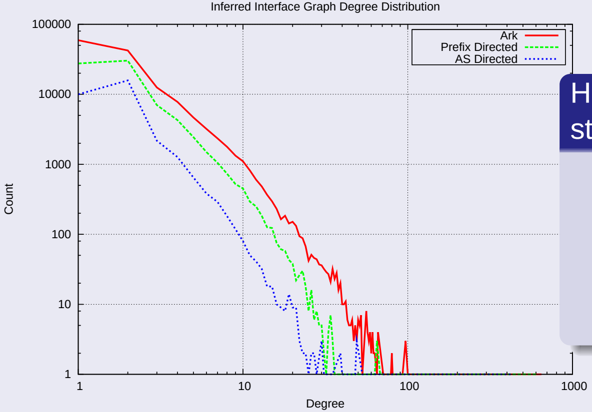
- Distribution of Ark traceroute probes to the size of the BGP prefix of the traceroute destination



naïve Strategy:

- Litmus test, how well do we do by probing only *one* point in each BGP prefix?
- Significant reduction in probing load
- Model fidelity?

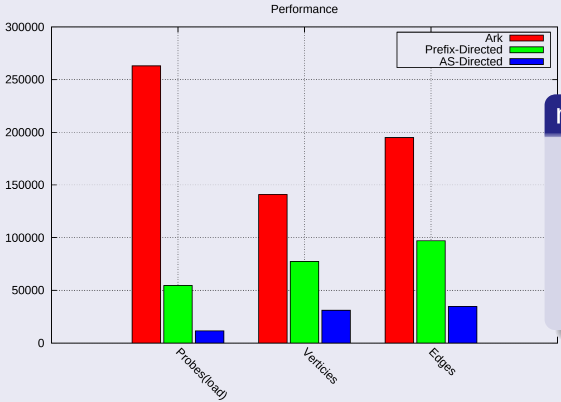
naïve Performance



How well do naïve strategies work?

- Reproduces similar structure
- But, misses significant information

How much load can be saved?

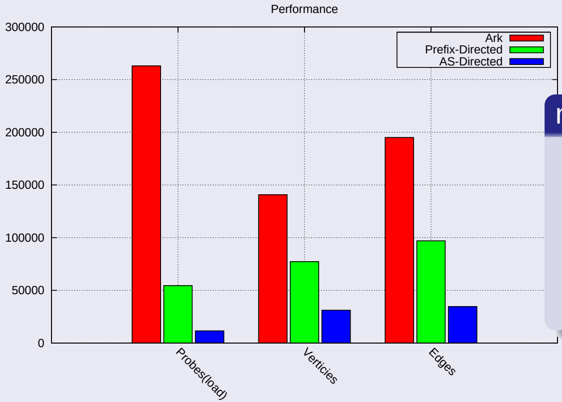


naïve Strategy:

- Huge savings in probing load
- But we’ve missed too many network links & nodes

Reproduce with higher fidelity with moderate increase in load?

How much load can be saved?



naïve Strategy:

- Huge savings in probing load
- But we've missed too many network links & nodes

Reproduce with higher fidelity with moderate increase in load?

Outline

Adaptive Sampling

naïve Strategy (2):

- Use edit distance on traceroutes to a pair of destinations in prefix
- We would expect two consecutive IP addresses to be more likely to share paths (low ED) than two distant addresses
- Use address distance?
- Doesn't capture structure of how networks are typically subnetted

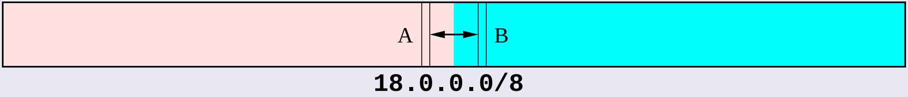
Adaptive Sampling

Current Strategy:

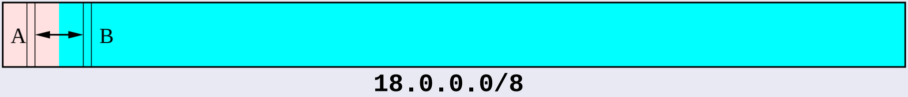
- Use knowledge of how networks are provisioned
- “max-min prefix” principle: maximize size of the minimum prefix induced by assuming two points are in *different* networks

Penalizing Complexity:

Easier to believe A and B in different subnets:



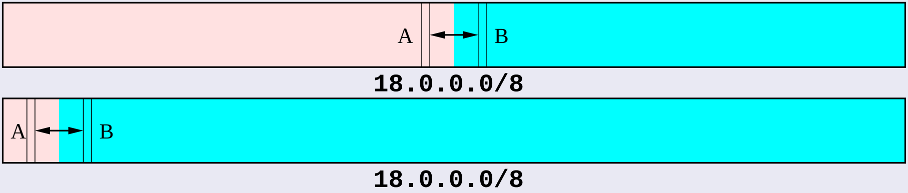
than A' and B' in different subnets:



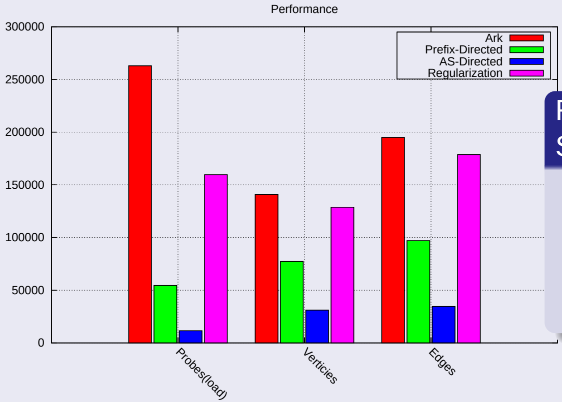
Adaptive Sampling

Max-min prefix:

- Let X be event that IP's A and B do not share path
- $P(X|max - min\ difference)$
- Idea: A high max-min difference implies that, in order for A and B to be in different networks, there is lots of subnetting
- Regularization, penalize more complex explanation (model)
- Find two points with high probability of being in different subnets
- Test their ED, recurse with a threshold



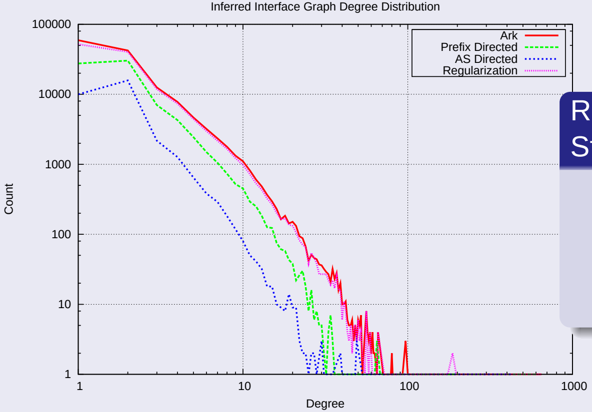
Regularized Model Performance



Regularization Strategy:

- $>$ 92% of vertex and edge fidelity
- $<$ 60% of the probing load

Regularized Model Performance



Regularization Strategy:

- Much better fidelity with baseline!
- Current work: do even better

Outline

Open Questions

1. Understand, quantify, and use information gain from other vantage points
2. Higher accuracy via selectively performing *more* traces to particular prefixes; requires actual deployment on Ark
3. Stability of topologies between probing cycles
4. Different edit distance metrics, for instance bit-level alphabet to capture similar, but different, IPs in path
5. Alias resolution using ED?
6. Lots more work to do ☺

Summary

Take-Aways:

- Deconstructed Ark topology tracing as case study
- Without sacrificing topological fidelity:
  - Large *packet* savings possible with single monitor
  - Significant *trace* savings possible with single monitor
  - $\Rightarrow$ more efficient, higher-frequency topology measurement
- Lots possible with multiple vantage point coordination

Thanks!

Questions?