End-to-end Service Quality Measurement Using Source-routed Probes

Summer intern project presentation

Fei Li

lifei@cs.columbia.edu

Mentor: Dr. Marina Thottan

Outline

- Why – motivation?
- What – abstract model?
- How – algorithms?
- How good – simulation results?
- Summary

Motivation

➢ Real-timely monitoring the network performance and service availability requires measurement techniques

  Measure end-to-end delay, packet loss, and the impact on service quality

  - Service-specific *probes* are active probes that closely mimic the service traffic such that they receive the same treatment from the network as the actual service traffic
  - Evaluating their impact of network impairments on service can be performed by end-to-end probes

Measurement Methods

- ➢ Previous/related work

  1. SNMP-based or link-level measurements cannot be used to model network services – for link failure or availability only
  2. Measured data should be correlated with topology information: `traceroute`
     - One-packet (pathchar): estimate link bandwidths
     - Packet-pair (ICMP): estimate available bandwidths and the bottleneck link rate
  3. IPMP: measure one-way delay using the path-record field in the IPMP packet; differently treated from normal service traffic

Source-routed Probes

- ➢ Source-routed probes mimic different network services

  1. Complete knowledge of the network topology
  2. Combined with the miscreant-link detection algorithm (`Parthasarathy, Rastogi, & Thottan, Bell Labs Technical Memo, 2005`) (to isolate the links contributing to the performance degradation)
  3. Source-routed probe mechanism avoids the correlation problem
  4. Network support for source-routing mechanism, such as MPLS.

Design Source-routed Probes

➢ In designing a set of probes, our goals are:

1. To minimize the cost of the probe traffic, while obtaining the maximum (resp. full) coverage of all (resp. interesting) links

   ⇒ Optimizing the total cost of the probe traffic

   ⇒ Optimizing the maximal-cost of a probe

2. To minimize probe installation costs and maintenance costs

   ⇒ Optimizing the number of probes

   ★ We do not consider minimizing the number of *terminals* in the context of this talk

An Example

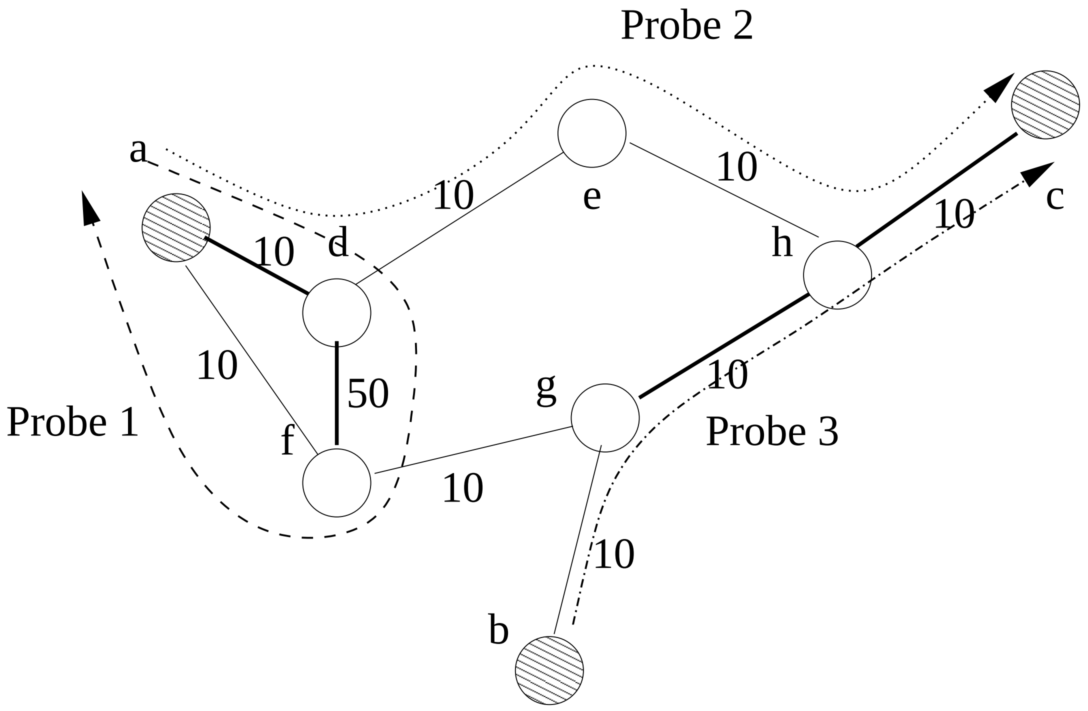


Figure 1: An example: 3 probes.

Our contributions

➢ Theoretical results:

1. An exact algorithm for minimizing the total probe traffic
2. A 2-approximation algorithm for minimizing the maximal-cost of a probe, in case the number of probes is bounded
   - Getting the exact solution is NP-hard
3. A 2-approximation algorithm for minimizing the number of probes, in case the maximal-cost of a probe is bounded
   - Getting the exact solution is NP-hard

➢ Simulation results:

For most ISP topologies: just 5% of the nodes as *terminals* to cover more than 98% of the edges $\Longrightarrow$ increasing the number of terminals does not help much in minimizing the total probe traffic

Outline

- ➢ Why – motivation?
- ➢ What – abstract model?
- ➢ How – algorithms?
- ➢ How good – simulation results?
- ➢ Summary

Abstract model

- Model the network as an undirected graph $G = (V, E)$

  A set of specific nodes as *terminals* $\mathcal{T} \subseteq V$; a set of interesting edges $S \subseteq E$

  A *path* is a set of concatenated edges between 2 nodes in V; an *elementary path* is a path without loops

  A *probe* is an elementary path from one terminal to another terminal

  Why: eliminating loops is necessary for practical implementation – a path with loops will be rejected by the routers

- A cost function $w_e \in \mathbb{R}^+$ over each edge $e \in E$

  The cost of a probe P is $w(P) := \sum_{e \in P} w_e$

- Our target: find a set of probes, such that . . .

Abstract model

➢ Find a set of probes $\mathcal{P}$, such that $\forall e \in S$, there exists at least one probe $P \in \mathcal{P}$, $e \in P$

`Link-covering problem (`**LCP**`) :` $\min \sum_{P \in \mathcal{P}} w(P)$

`Primal link-cover problem (`**PLP**`) :` $\min(\max_{P \in \mathcal{P}} w(P))$

`subject to:` $|\mathcal{P}| \leq k$

`Dual link-cover problem (`**DLP**`) :` $\min k$

`subject to:` $|\mathcal{P}| = k$

$w(P) \leq l_{max}, \forall P \in \mathcal{P}$

An Example

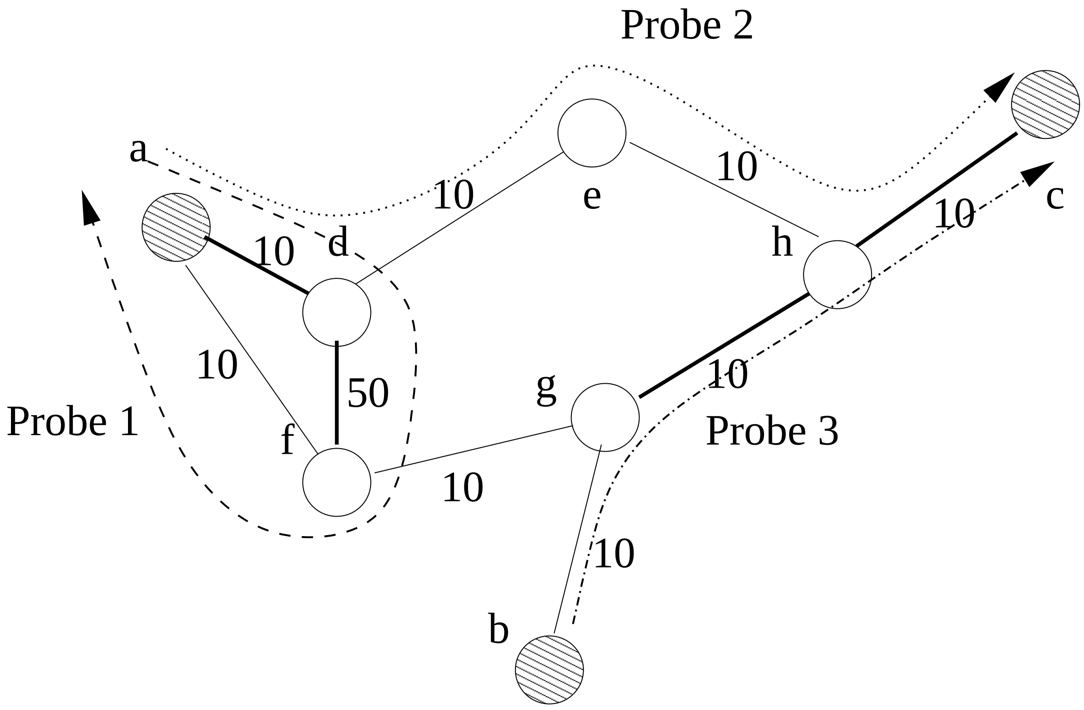


Figure 2: 3 probes, the maximal-cost of a probe is 70; the total cost of the probes is 140; probe 2 is unnecessary.

Outline

- ➢ Why – motivation?
- ➢ What – abstract model?
- ➢ How – algorithms?
  1. **LCP**
  2. **PLP** and its hardness
  3. **DLP** and its hardness
- ➢ How good – simulation results?
- ➢ Summary

An Exact Algorithm for **LCP**

$\min \sum_{P \in \mathcal{P}} w(P)$

- High-level ideas:

  Divide-and-conquer

  Optimal for both

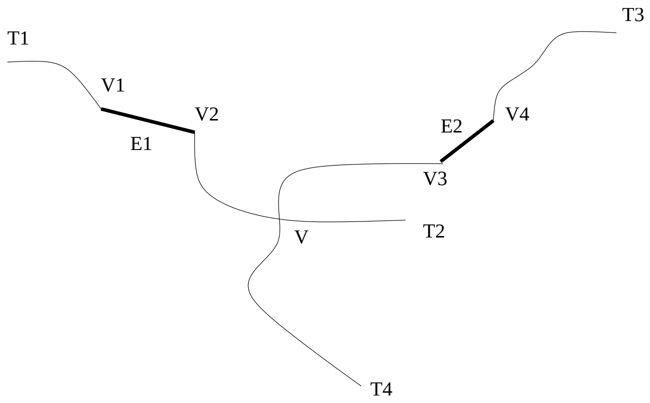
- Approaches:
  1. For each edge $e \in S$, find its minimal-cost probe
  2. Remove redundancy

- Techniques used:
  1. Indexing all terminals
  2. Shortest-path from one node to another node
  3. Case analysis for redundancy
  4. *Virtual terminals* on a probe

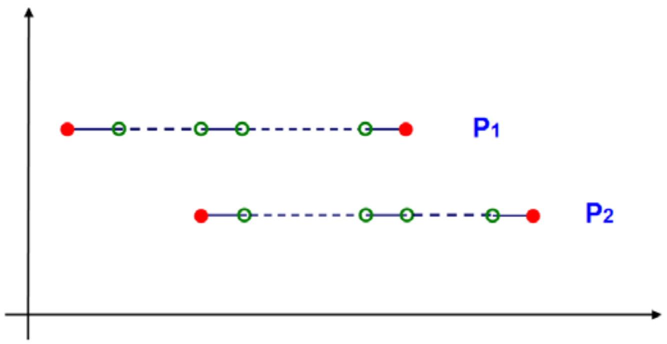
```
Indexing all terminals;
for each edge $e$ do
  for each terminal do
    Find a shortest path from one end of $e$ to one terminal in $G$, ties broken;
    Remove all intermediate nodes and associated edges;
    On the remaining graph, find the shortest path from the other end of $e$ to one terminal;
  end for
end for
Choose the minimal-cost probe $P_e$ for edge $e$;
Remove all interesting edges $\in P_e$; mark $e$ and them as $Y$ to $P_e$ if edges $\notin P_i$, $\forall i \neq e$; else mark as $N$ to $P_e$;
for each probe $P_i$ do
  if $\forall f \in P_i$, $f$ is marked $N$ to $P_i$ or shared edge(s) are marked $N$ to $P_i$ then
    Remove probe $P_i$ or concatenate unshared part at the joint point;
    Update edge status;
  end if
end for
```

Optimality Proof

1. Optimal for divide (single edge) and conquer (combine)
2. Contradiction method used to prove for the single edge case
3. Any 2 probes have no shared nodes (crossing points), but (possibly) shared edges
4. No cross-link terminals $\Rightarrow$ end nodes of end edges act as terminals with associated *gain* $\Rightarrow$ Disjoint probes can be combined if cost is reduced



(a)



(b)

An Example

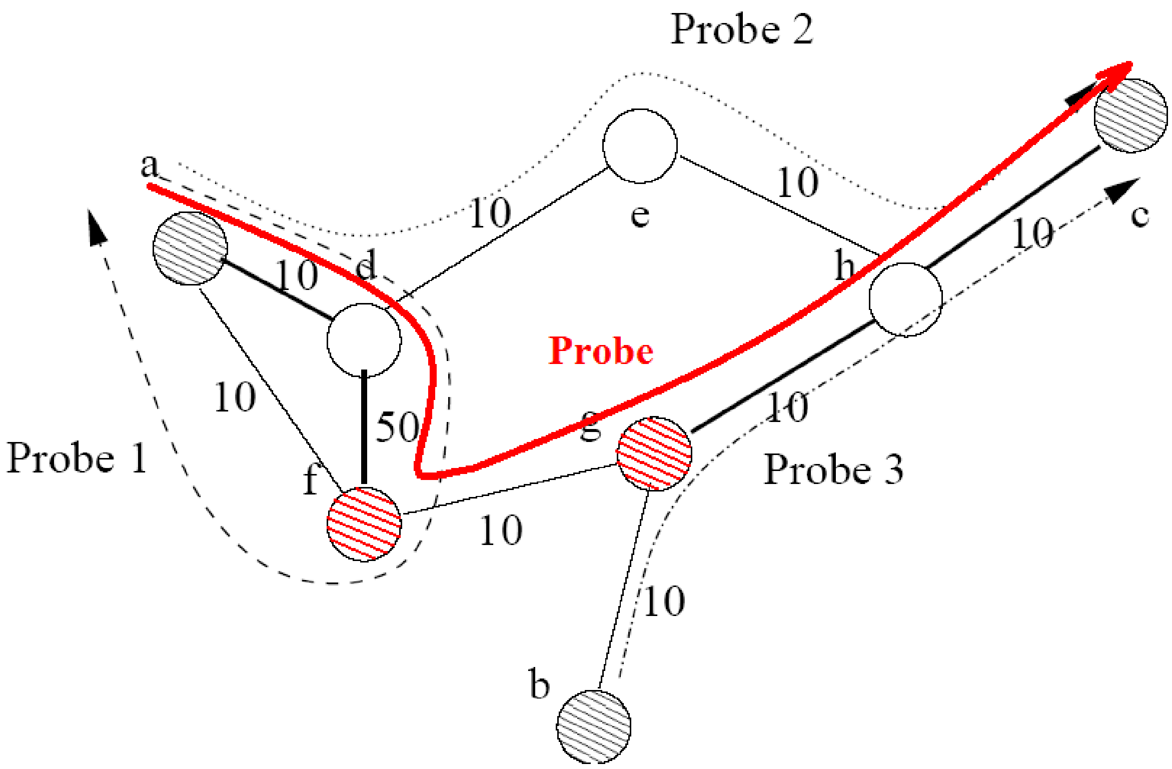


Figure 3: 1 probe, the maximal-cost and the total cost of the probe is 90.

NP-hardness for **PLP** and **DLP**

$\min(\max_{P \in \mathcal{P}} w(P))$, subject to: $|\mathcal{P}| \leq k$

$\min k$, subject to: $|\mathcal{P}| = k$, and $w(P) \leq l_{max}, \forall P \in \mathcal{P}$

Reduced from Minimal Makespan Problem

Reduced from Bin-packing Problem

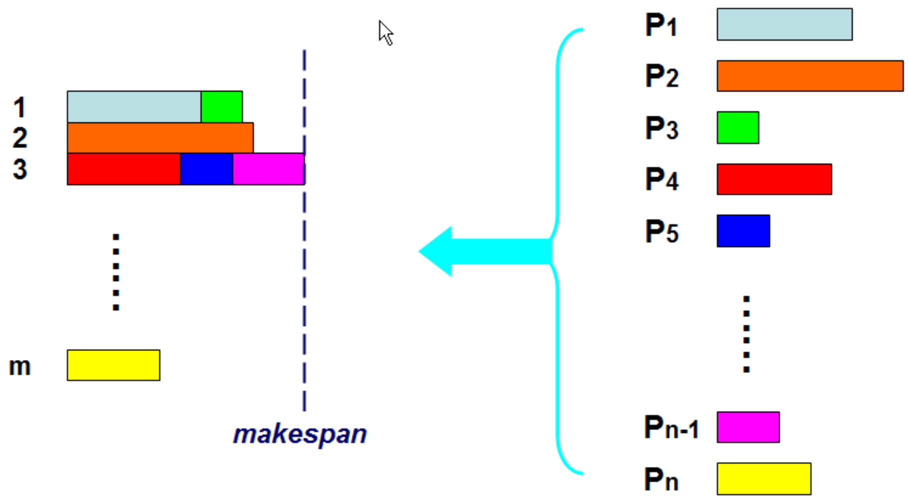


Figure 4: Minimal Makespan problem is NP-hard.

NP-hardness for **PLP** and **DLP**

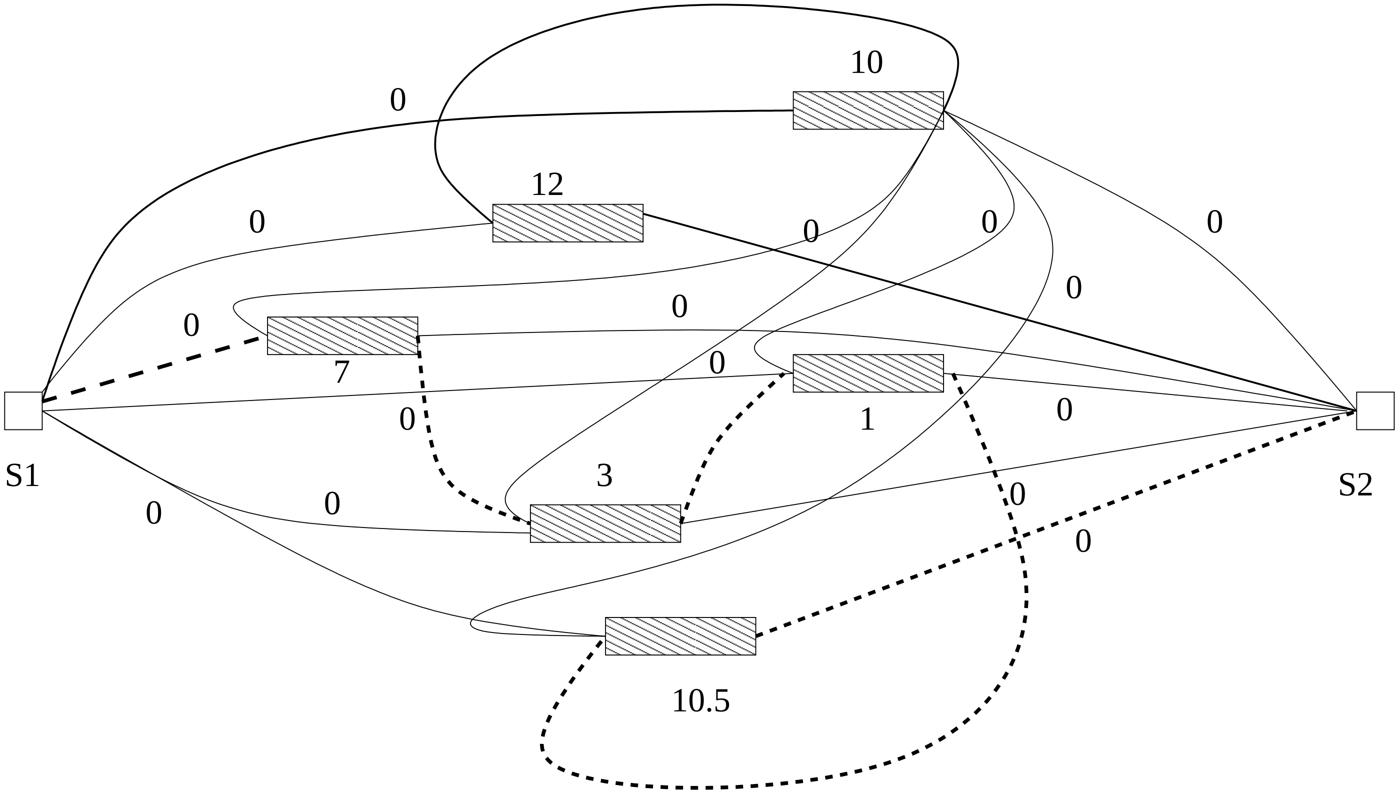


Figure 5: An example illustrating the reduction. Not all links are shown.

Approximation Algorithms for **PLP** and **DLP**

$\min(\max_{P \in \mathcal{P}} w(P))$, subject to: $|\mathcal{P}| \leq k$

- Two-stages for **PLP**
  1. Find a probe for each edge; end nodes of end edge act as terminals
  2. Merge 2 probes with minimal-cost between 2 terminals
- Binary search for the solution to **DLP**
- Analysis

  Feasibility: merging still results elementary probe when no shared edges (proved in the paper)

  Performance: 2-approximation, similar to the bin-packing algorithm's proof (see the paper for details)

Outline

- Why – motivation?
- What – abstract model?
- How – algorithms?
- How good – simulation results?
- Summary

Simulation Set-up

1. ISP topologies from RocketFuel project
2. The largest 5 topologies: Telstra (Australia), Sprintlink (US), Verio (US), Level3 (US), and AT&T (US)
3. The terminals are chosen from the backbone nodes: 5%, 10%, and 15% of $|V|$
4. The interesting edges are randomly selected: 25%, 50%, 75%, and 100% of $|E|$

Name	$\lvert V \rvert$	$\lvert E \rvert$	used T (as %)	covered E (as %)	probe cost total	probe cost average	probe cost maximal	# of probes
Telestra (Australia)	351	784	17 (5%)	392 (50%)	802	3.46	8	232
				769 (98.1%)	1436	3.25	9	442
			52 (15%)	392 (50%)	635	2.56	5	248
				769 (98.1%)	1262	2.58	5	490
Sprintlink (US)	604	2279	30 (5%)	1139 (50%)	3290	3.91	10	842
				2277 (99.9%)	6323	3.85	10	1643
			90 (15%)	1139 (50%)	2495	2.77	5	902
				2277 (99.9%)	4852	2.77	5	1751
Verio (US)	972	2839	48 (5%)	1419 (50%)	3671	3.72	12	987
				2839 (100%)	6782	3.88	19	1749
			145 (15%)	1419 (50%)	2979	2.70	8	1103
				2839 (100%)	5240	2.66	8	1967
Level3 (US)	624	5301	31 (5%)	2650 (50%)	7588	3.29	8	2304
				5301 (100%)	15124	3.27	8	4621
			93 (15%)	2650 (50%)	6460	2.72	11	2378
				5301 (100%)	12951	2.72	11	4753
AT&T (US)	631	2078	31 (5%)	1039 (50%)	2889	3.93	11	736
				2078 (100%)	5356	3.85	11	1392
			94 (15%)	1039 (50%)	2281	2.94	8	776
				2078 (100%)	4432	2.89	8	1534

Simulation Results on Telestra

used T (as $\%$)	covered E (as $\%$)	probe cost total	probe cost average	probe cost maximal	# of probes
17 (5%)	196 (25%)	558	3.44	7	162
	392 (50%)	802	3.46	8	232
	588 (75%)	1435	3.30	7	435
	769 (98.1%)	1436	3.25	9	442
38 (10%)	196 (25%)	519	3.00	5	173
	392 (50%)	911	2.99	5	305
	588 (75%)	1492	3.33	5	452
	769 (98.1%)	1490	3.13	5	476
52 (15%)	196 (25%)	457	2.77	5	165
	392 (50%)	630	2.50	5	248
	588 (75%)	1390	3.00	5	463
	769 (98.1%)	1262	2.58	5	490
88 (25%)	196 (25%)	337	2.88	3	117
	392 (50%)	946	2.92	4	324
	588 (75%)	1371	2.77	4	495
	769 (98.1%)	1520	2.87	4	530

Simulation Results on Approximation Algorithms

Simulation of **PLP** algorithm using 15% of nodes as terminals, and covering all edges, and k is $1/2$ of the probes of **LCP**.

Name	$\lvert V \rvert$	average degree	$\lvert \mathcal{T} \rvert$	# maximal-cost before merge	# maximal-cost after merge
Telstra	351	2.336	52	5	9
Sprintlink	604	3.77	90	5	10
Verio	972	2.92	145	8	10
Level3	625	8.41	93	11	11
AT&T	631	3.29	94	8	10

Simulation

1. Not all edges randomly selected can be covered by a probe
2. The number of hops accounts for the cost of a probe
3. Terminals resides in backbone nodes

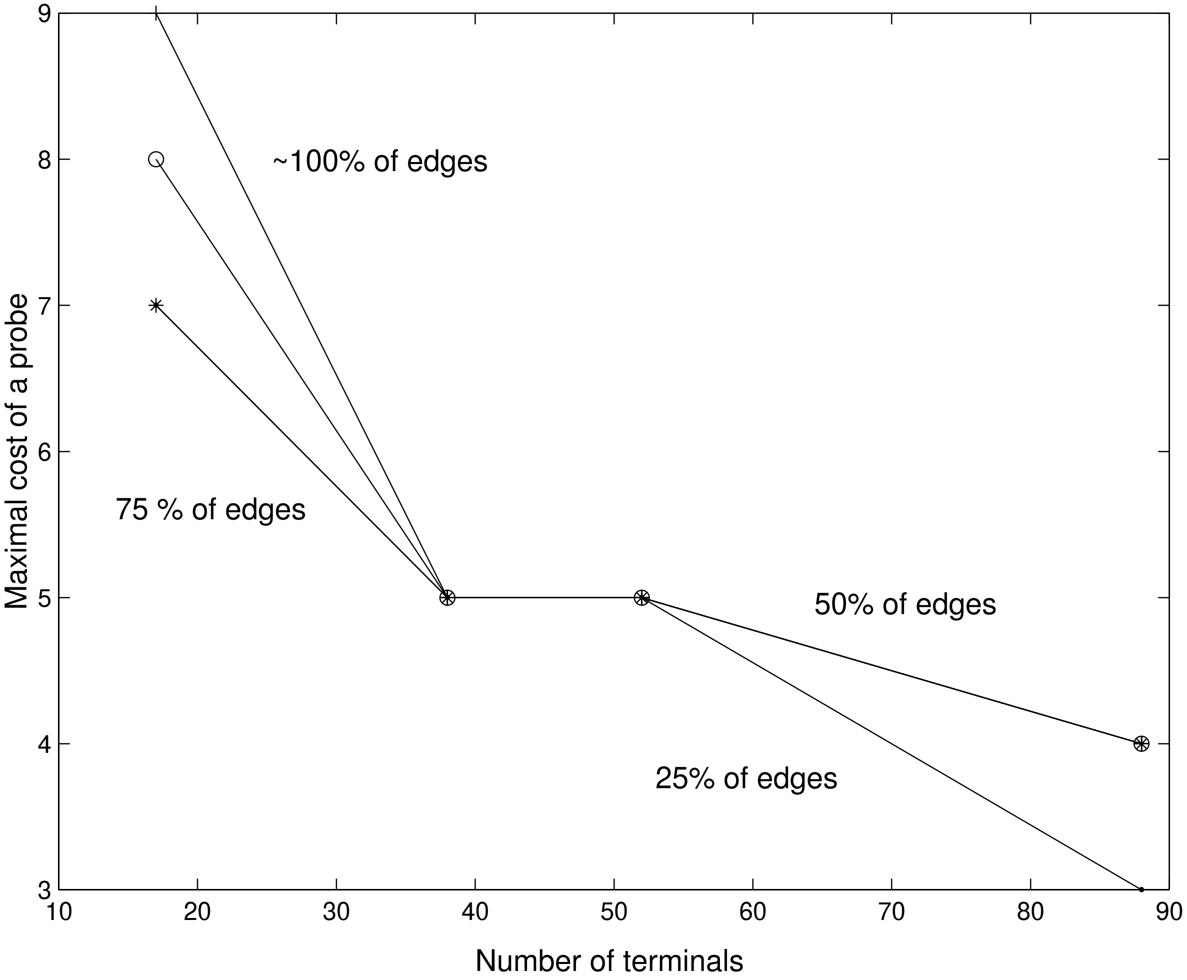


Figure 6: Maximal-cost of a probe v.s. the number of terminals for the Telestra topology.

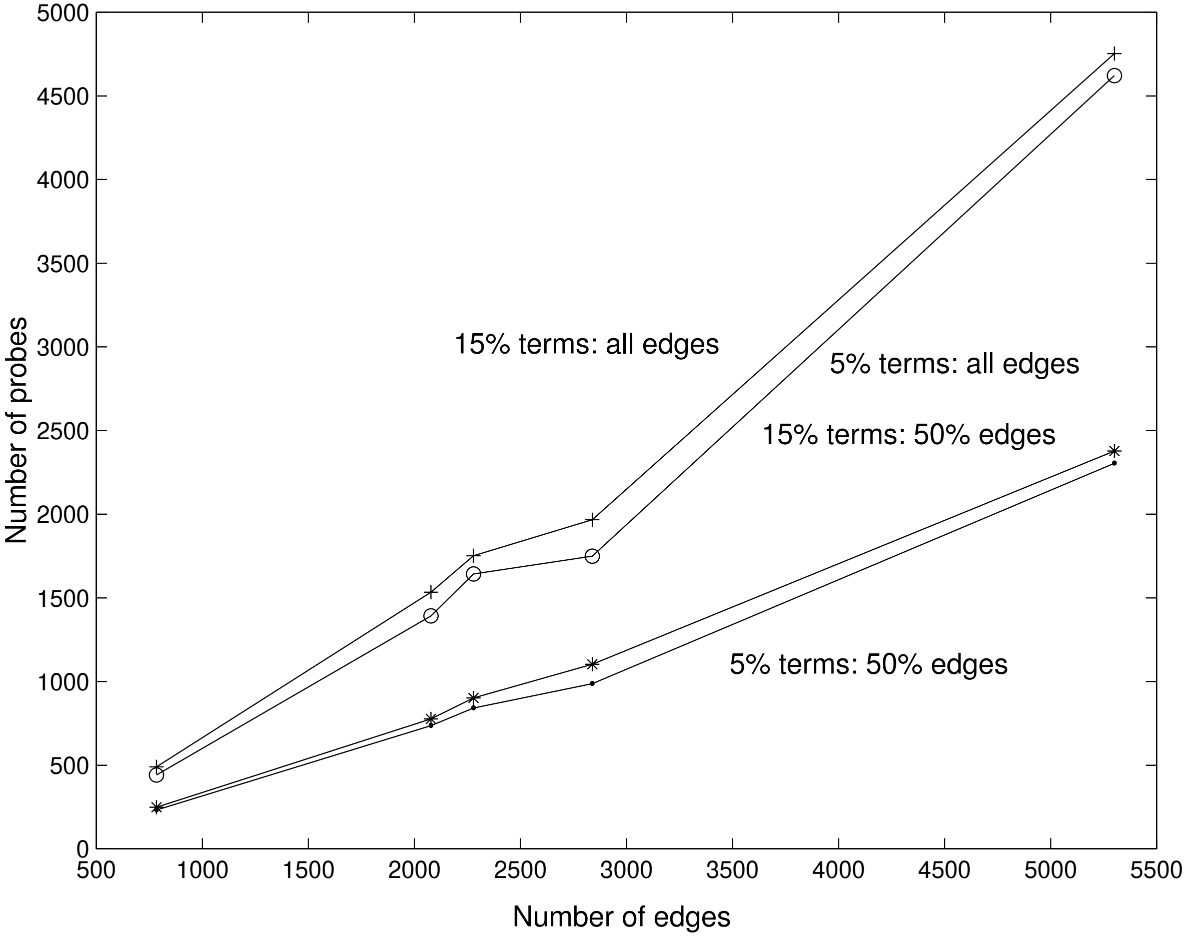


Figure 7: The total number of edges in the network v.s. the total number of probes.

Outline

- Why – motivation?
- What – abstract model?
- How – algorithms?
- How good – simulation results?
- Summary

Future Work

1. Consider the location and/or number of terminals, see related work (`Bejerano & Rastogi INFOCOM'03`)
2. $(1+\epsilon)$-approximation algorithms for **PLP** and **DLP**, based on the PTAS solutions to the Minimal Makespan Problem & Bin-packing Problem ?
3. 2-criteria optimization problem (probe traffic, $\#$ of terminals), the Pareto optimality?
4. Topological issues should be taken into account
5. Online version of this topic