Routing Through Teranode Networks with Topology Aggregation

Baruch Awerbuch *Yi Du* *Bilal Khan* *Yuval Shavitt**
{baruch, yidu, bkhan, shavitt}@cs.jhu.edu
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218

Abstract

The future global teranode networks will consist of many subnetworks called domains. For reasons of both scalability and security, domains will not release details of their internal structure to nodes outside of the domain . Rather, they will release a summary, or aggregation, of the internal structure, e.g., as required by the ATM PNNI standard.

This work compares by simulation the effect of different aggregation schemes on the network throughput, and on the network control measured in the number of crank-backs per realized connection. It shows, that using stars for aggregation results in bad performance when compared with other simply methods.

keyword: PNNI, topology aggregation, routing, teranode networks, hierarchical network structure

Areas of interest: internetworking, network signaling and control

*Corresponding author. Fax: (410) 516 6134, Tel: (410) 516 5036

1 Introduction

The future global teranode networks will consist of many subnetworks called domains. Typically these domains will correspond to portions of the network that reside in a particular geographical region or that belong to a particular organization. For reasons of both scalability and security, domains will not release details of their internal structure to nodes outside of the domain [tc96, CCS96]. Rather, they will release a summary, or aggregation, of the internal structure. The need for aggregation purely for the sake of scalability is clear from the simple observation that the complexity of routing protocols, both for source routing and link-by-link routing, grows at least linearly with the number of links in the network. We note that very large networks have a hierarchical domain structure.

The most compact representation of topology information that does not compromise precision is to use a complete graph between the externally visible *border* nodes of a domain. This approach requires a large quadratic representation, and thus is of little practical use. When restrictions such as communication cost, topology database storage space, and route calculation time require to trade precision with representation size there are many ways the internal structure of a domain can be aggregated. The most promising approach is to find a new aggregate topology which is simpler than the real topology, yet still captures its structure.

This is the approach taken by the Private Network-to-Network Interface (PNNI) group of the ATM Forum. The PNNI standard [tc96] suggests to replace internal network structure by a star with a virtual center and spokes to the border nodes. Other, simple aggregation schemes are to use minimum (or random) spanning trees. Unfortunately, stars and trees can cause error that is linear in the ratio of the maximum to minimum cost of a path in the complete graph representation.

Interestingly, the effect of topology aggregation on the performance of networks was rarely studied in the past. There is a large body of theoretical research that studies compact graph representation [Bar96, PU88, PS89, ADD$^+$93] but the emphasis in the theoretical research is on the maximum cost distortion between any pair of nodes in the aggregated graph. It is not clear that this is an important evaluation criterion of an aggregation scheme, and as our study shows it is certainly not the most important or the only one.

In this work, we simulate several aggregation schemes and compare their performance. We concentrated on stars, spanning trees, and spanners [PU88, PS89, ADD$^+$93], and compare their performance to the case where no aggregation is performed. We show that the differences in the throughput and, to a larger extent, in the network control overhead can be hundreds of percents.

We also study the effect of link cost functions, and the re-aggregation policy. We show that the performance when exponential link cost function is used is much better than the performance

of min-hop routing. Re-aggregation policy determines when the topology aggregation needs to be re-computed. We show that one can save half of the aggregation computation overhead by re-aggregating only when the residual bandwidth is crossing boundaries between bandwidth intervals whose width is exponentially growing.

The rest of the paper is organized as follows. In the next section, we describe the simulation environment. Section 3 contain the results from the simulation we conducted, and the last section is a compilation of our conclusions and some future research directions.

2 The simulation environment

We build an event simulator that emulates PNNI routing over hierarchical networks. The simulator has a modular structure that enables easy plugging of most of its features. In particular, we used for this work several link cost functions for routing, various topology aggregation methods, and numerous topologies. We plan to use CAC in our future studies. For increased performance, the aggregations are done by a separate *aggregation server* that accepts a topology and an aggregation method code and returns the resulted aggregated metric. A server can serve multiple requests simultaneously. A graphical display shows the progress of the reservation process in time. The simulator is available at `http://www.cnds.jhu.edu/aggregation`.

Two performance measures are used to compare the performance of the algorithms:

Throughput measured by the cumulative length of the realized connections, and

Control overhead measured by the average number of backtracks[1] per connection.

The control overhead is an indicator to the efficiency of the routing under a certain aggregation scheme.

There are many parameters and environment attributes that can influence the performance. The main one of interest is the aggregation schemes that are described in section 2.1. However, other parameters might enhance or suppress the differences among the aggregation schemes. The metric used by the routing algorithm in calculating the shortest path was checked. The comparison between the minimum hop metric and the exponential cost function, as described in section 2.2, is an important result of this paper. Other parameters under which the different aggregation schemes where checked are: Link delay (normalized to the connection average holding time), network load, and link capacities.

[1] Crank-backs and backtracks are used interchangeably

aggregation scheme	representation size	calculation complexity
Complete	$b(b-1)$	-
DIA/AVE	1	$O(b^2)$
MST	$b-1$	$O(b^2)$
RST	$b-1$	$O(b^2)$
Spanner	$O(b^{1+1/t})$	$O(b^4)$

Table 1: A summary of the aggregation schemes simulated in this paper

To gain insight on the effect of the different parameters (especially aggregation schemes) on performance we examine topologies which emphasize the differences between the examined parameters. We also report results for randomly generated topologies that are commonly used in simulation studies [Wax88]. Finally, we test the influence of the aggregation policy, i.e., when we should report about changes and re-advertise.

2.1 Aggregation methods

The following aggregation were simulated:

Complete — No aggregation is done. The full cost matrix between the border nodes is broadcasted.

DIA — A star aggregation where the cost of the star radius is half the cost of the network diameter.

AVE — A star aggregation where the cost of the star radius is half the average cost of the distances between the border nodes.

MST — A minimum spanning tree.

RST — A random spanning tree.

Spanner — a t-spanner. We use $t = 2$ in most of the simulations.

Table 1 summarizes complexities associated with the aggregation schemes as a function of the number of border nodes, b.

Complete is used to check the performance loss due to aggregation. In some cases, especially when the link delay was low, Complete exhibits slightly worse performance than other methods. This is due to the on-line nature of the problem, where 'optimal' decision made at a certain moment may prove to be bad for future connection requests.

2.2 Link cost

Two link cost metrics were used

constant — The link cost is constant regardless of the available bandwidth along it. This metric corresponds to minimum hop routing.

exponential — The link cost is an exponential function of the residual bandwidth. This cost function stems from both queueing theory and opportunity cost functions. If link cost is to be determined by the expected queueing delay of a packet through the link, which is the major factor in the total delay, the delay function is given by $[\mu(c-f)]^{-1}$, where c is the link capacity, f is the current flow through the link, and μ is the service rate. As f approaches c, the link cost increases to infinity. Opportunity cost functions suggest to maximize utilization by charging link usage according to a function that increases exponentially as the residual bandwidth decreases.

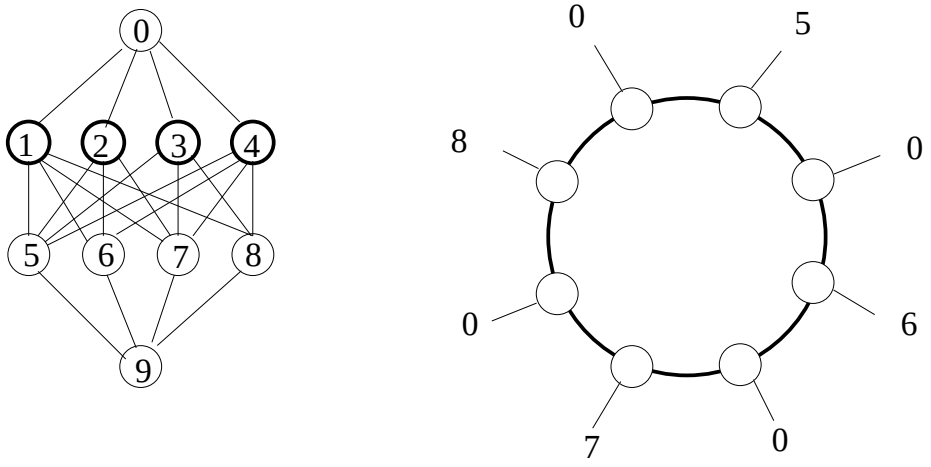
When backtracks are used in both link cost metrics, the entire network bandwidth is bound to be consumed regardless of the aggregation method in use. Thus, throughput is expected to be similar for the various aggregation methods with differences that are only due to interaction between concurrent reservations. In some cases, larger difference may occur due to extremely bad route selection due to inaccurate aggregated information.

3 Simulation results

Theoretical results show that optimal competitive on-line routing algorithms can be obtained by using an exponential cost metric. As a result, we select this metric for our simulations. In section 3.1 we report simulations that confirm this theoretical result. We first study some regular topologies, and then report results from simulations of randomly generated networks.

The regular topologies we simulated are depicted in figures 1 and 2. Figure 1 is a topology of two hierarchical levels where the internal structure of some of the nodes is a ring. Figure 2 is a topology of three hierarchical levels. Each level is a multi-stage graph.

Figures 3, and 4 compare the performance of the aggregation schemes for the topology where rings are connected in stages as depicted in figure 1. We simulated flows from a single entry point (node 0 in figure 1) to a single exit point (node 9 in figure 1). When the link delay is low, their is little difference between the aggregation schemes except for the RST under which the throughput is lower by 10% for medium load, and up to almost 50% for high load. When the link delays are longer, tree aggregation schemes, that were only slightly below the Complete's throughput in low link



A. High-level topology

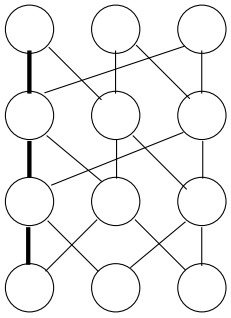
B. Second level topology with connectivity to high-level nodes

Figure 1: A layered ring topology. Node 1-4 (bolded) on the right are structured as rings and connected as indicated by the ring on the left. The number at the end of the edges on the left indicate the node number to which this edge is connected in the higher layer. Link capacities: 5 in the second level ring, 6 elsewhere.

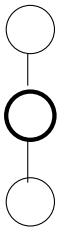
delays, are now 10% and more worse. RST remains the worst aggregation scheme for this topology in terms of both throughput and backtracks. For the number of backtracks the results are illustrated in figure 4. The relative order among the schemes is the same, but with more profound differences. For RST the number of crank-backs is up to 400% more than that for Complete, while for DIA/AVE this number is 100% more when compared to Complete. There is no visible difference among Spanner, MST, and Complete.

Figures 5, and 6 compare the performance of the aggregation schemes for a layered topology that has a hierarchical structure (see figure 2). The highest layer contains three nodes, two of them are physical nodes used as source and destination for the flows. The node in the center is a complex node consisting of two additional layers of multi-stage graph. Under the two star aggregation schemes throughput is almost 10% lower than when the other aggregation schemes are used. The number of backtracks when stars are used to aggregate is, at least, double than this number for the other schemes. There is no significant difference among the other aggregation schemes.

To study the effect of the network diameter on the performance, we generalize the network in figure 1. We create networks that are built from S stages of W rings, each ring is comprised of $2W$ nodes. We call these networks (S, W) Stage-Ring, or simply (S, W)-SR. The network in figure 1 is an (1,4)-SR network, figure 7 depicts an (2,4)-SR network.



A. topology of layers 2 & 3



B. topology of layer 1

Figure 2: A hierarchical layered structure. The middle node in layer 1 (right hand side) is structured as depicted on the left. In layer 2 each of the 12 nodes is structured recursively the same. Link capacities: 5 for internal links in layers 2 and 3, 10 for the bolded internal links in layers 2 and 3, 10 for links between border nodes, 15 for links between border nodes that both have bolded links.

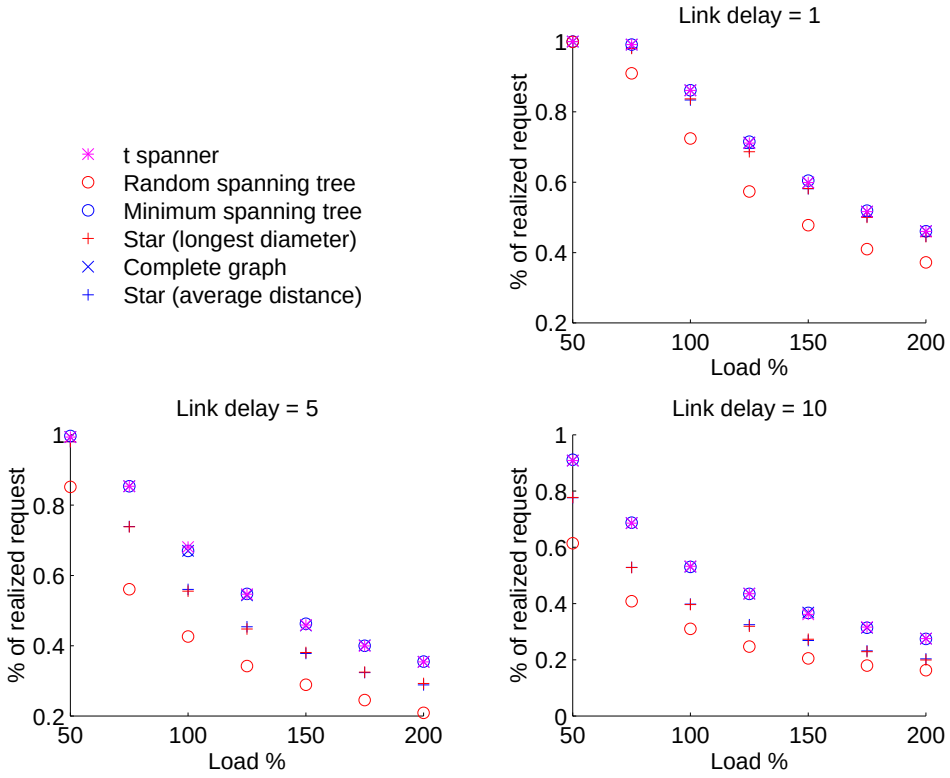


Figure 3: Throughput in the layered ring topology of figure 1.

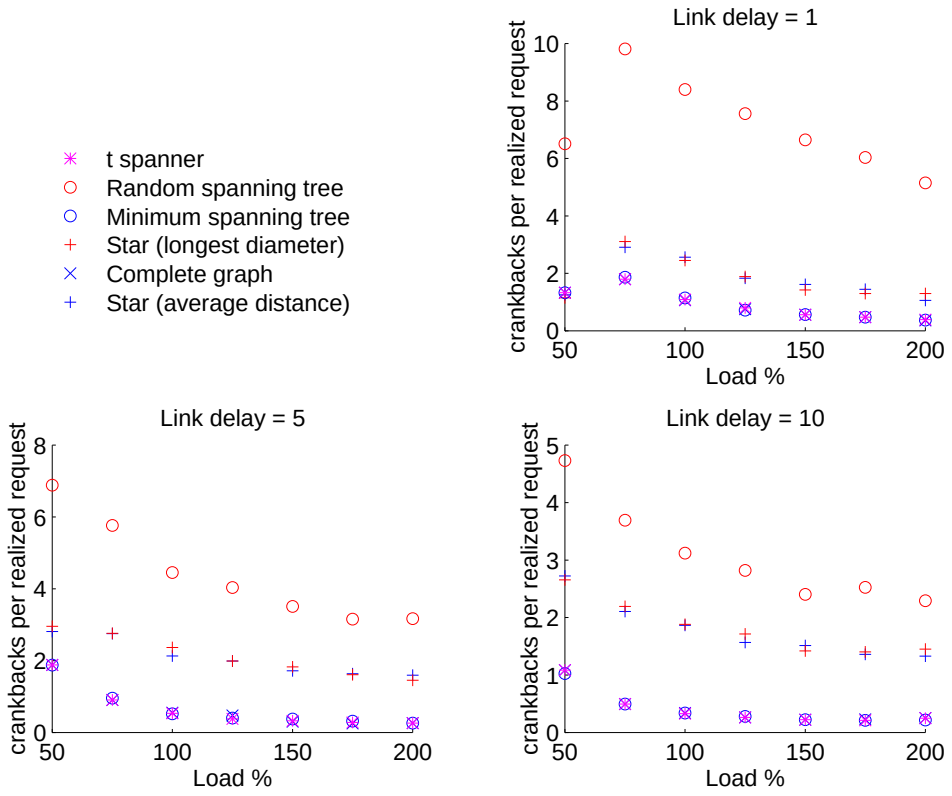


Figure 4: Backtracks in the layered ring topology of figure 1.

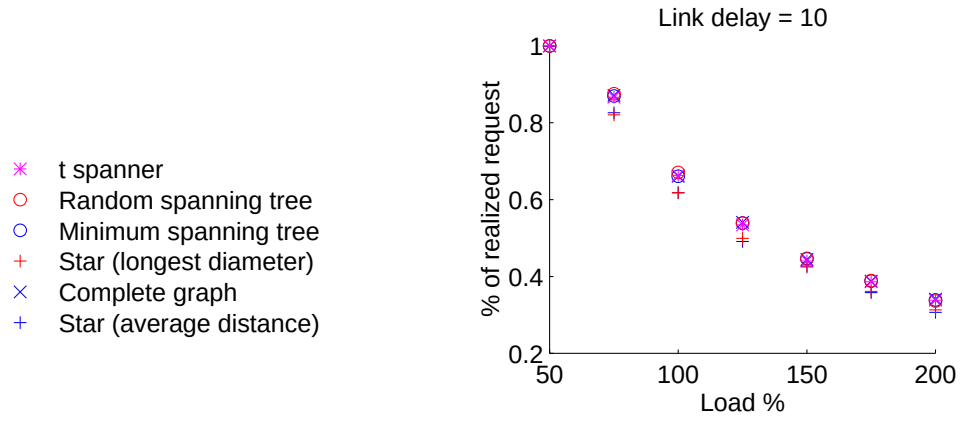


Figure 5: Throughput in the layered graph topology of figure 2.

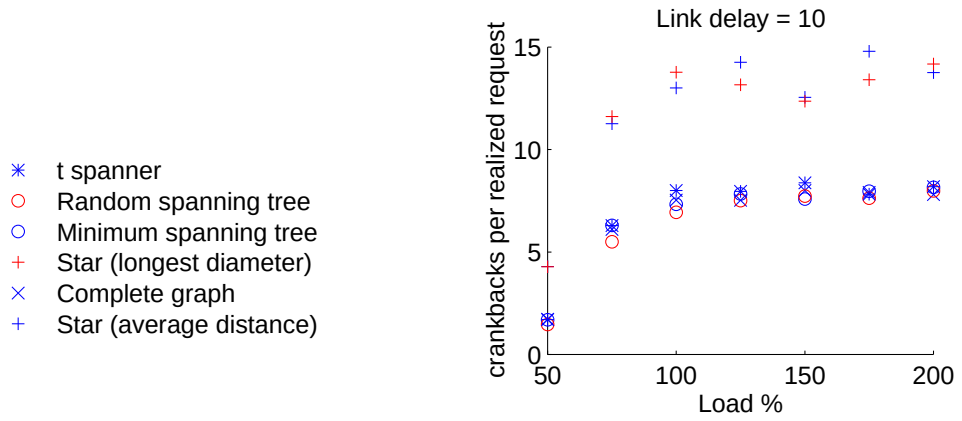
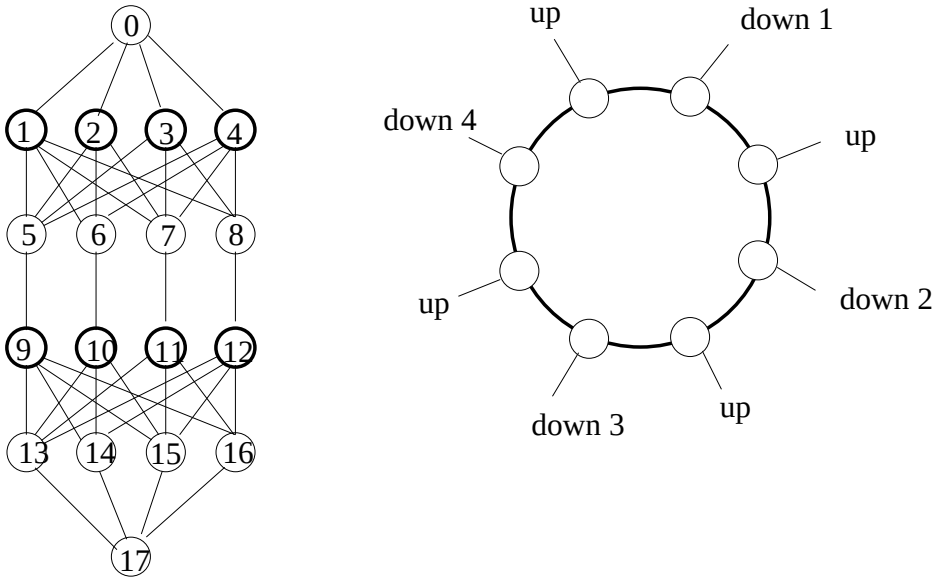


Figure 6: Backtracks in the layered graph topology of figure 2.



A. High-level topology

B. Second level topology with connectivity to high-level nodes

Figure 7: An (2,4)-SR network.

link delay	AVE	Complete	DIA	MST	RST	Spanner
1	40.8174	35.1618	38.8821	34.8019	44.1552	35.7323
10	11.9472	11.0179	12.8383	10.4952	14.1115	10.6929

Table 2: Backtrack at load 100% for the random topology of figure 8

link delay	AVE	Complete	DIA	MST	RST	Spanner
1	0.473331	0.476973	0.473570	0.476223	0.473756	0.476714
10	0.155969	0.159922	0.162303	0.155457	0.145881	0.160501

Table 3: Flow at load 100% for the random topology of figure 8

We check how the network diameter effect the performance. In all the SR networks the behavior of the aggregation schemes is similar in the sense that figures look alike only the scale of the Y-axis is different. When the network diameter grows the number of crank-backs increases. Interestingly, the number of crank-backs grows faster than linearly with the number of stages, S. The total flow through the network decreases with S as the mutual disturbances among flows increase with the diameter.

On the other hand, as W grows the number of crank-backs increases sharply, although the network diameter does not change. This increase can be explained by the increase in the cost of selecting the wrong entry point to the ring, and by the increased risk of en route rendezvous of competing reservation requests.

Although the number of crank-backs increases with S, the relative difference between the aggregation schemes decreases with S. For example, for $S = 1$ the ratio between both star aggregations and MST, Spanner, and Complete climbs from 2 at low load to over 5 in high loads. For $S = 2$ this ratio is about 1.5, and for $S = 3$, it is about 1.4. We note that for $S = 1$ the difference increase with the load while for $S > 1$ it stays flat. It can be explained by the fact that the first stage of rings is where the majority of requests fail to pass, and for the rest of the ring stages the flow of requests does not increase with the load. This phenomenon was observed by Cidon et al. [CRS96] for one-way reservation schemes.

We turn to investigate random topologies. We generate random two level networks depicted at figures 8 and 9. At each level the nodes were given random locations on a grid, with 7 (or 8) in the high level and 20 in the low level. Four nodes in each low level sub-network were forced on a

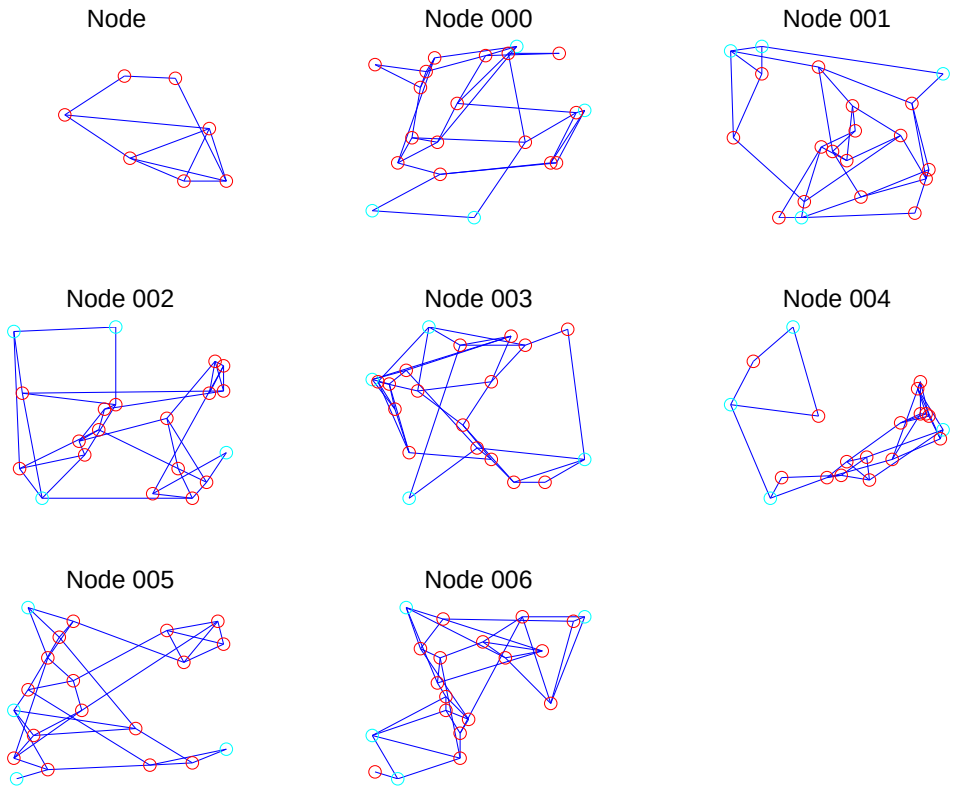


Figure 8: A random topology with seven subnetworks each is comprised of 20 nodes.

link delay	AVE	Complete	DIA	MST	RST	Spanner
1	36.0455	34.8815	37.3631	35.2874	38.8096	34.0514
10	11.4899	10.4486	12.0092	10.0924	12.1570	10.1349

Table 4: Backtrack at load 100% for the random topology of figure 9

link delay	AVE	Complete	DIA	MST	RST	Spanner
1	0.438416	0.443879	0.440972	0.442210	0.431519	0.443361
10	0.104207	0.112880	0.108013	0.111831	0.113848	0.114851

Table 5: Flow at load 100% for the random topology of figure 9

random location on the grid edge and were marked as potential border nodes (light circles in figure 8 and 9). Links are added by a random process that generates a random node-pair and adds a link between them with probability that decreases exponentially with the distance between the two nodes [Wax88]. Links that increase the node degree (external links are not counted) above four are rejected. The process terminates when all nodes degree is at least two. High level links are assigned arbitrary border nodes in the corresponding sub-networks. Note that at least in one case (subnetwork 002 in figure 9), the algorithm failed to increase one on the border nodes degree to two (and report the problem). Flows were randomly generated between nodes that reside at different domains.

Tables 2 and 3 summarize the simulation of the networks depicted at figure 8. The differences in the throughput were generally less than 1.5% between the aggregation schemes with the exception of RST that falls almost 9% behind Complete for delay 10. For delay 10, DIA is better than Complete by 1.4%, this small difference can be attributed to the on-line nature of the problem, where full knowledge of the past and the present does not guarantee optimal decision about future requests.

The differences between crank-backs for the aggregation schemes are more visible. Star aggregations require 8.5-16.5% more crank-backs than Complete to achieve similar throughput. MST performs slightly better than Complete, and RST requires 25% more crank-backs and achieves less throughput.

Tables 4 and 5 summarize the simulation of the random two level network depicted at figure 9. The results for crank-backs are similar to the previous topology, the differences between the number of crank-backs for AVE, DIA, and RST and that for Complete, MST, and Spanner are usually in the double digit percent region (AVE is a little better).

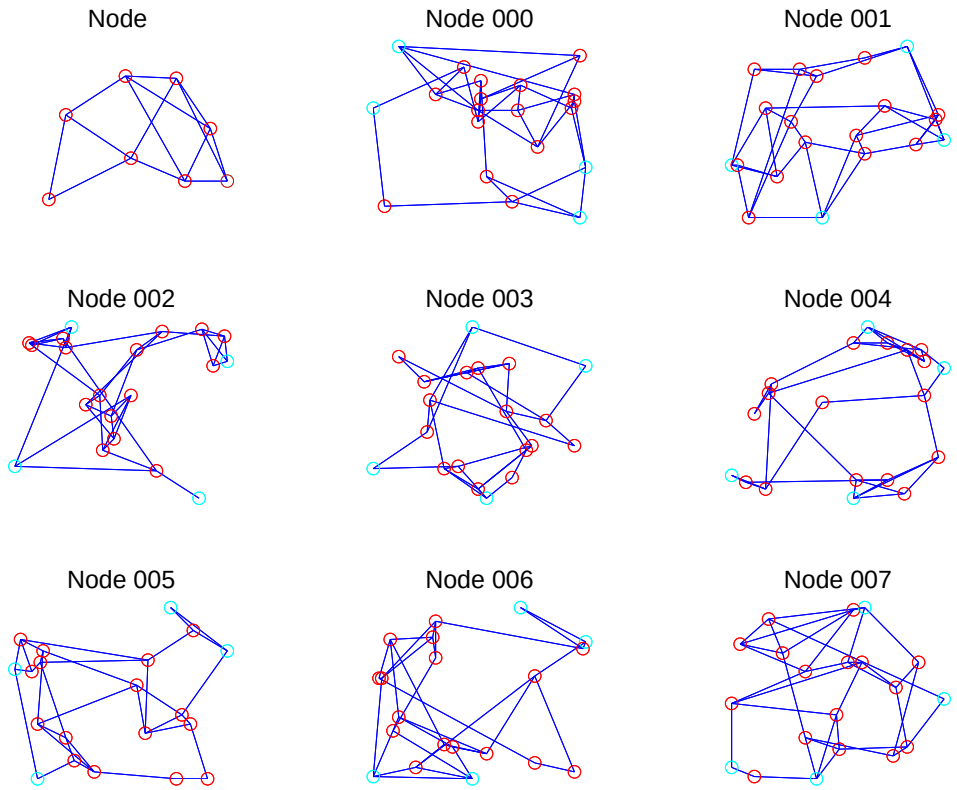


Figure 9: A random topology with eight subnetworks each is comprised of 20 nodes.

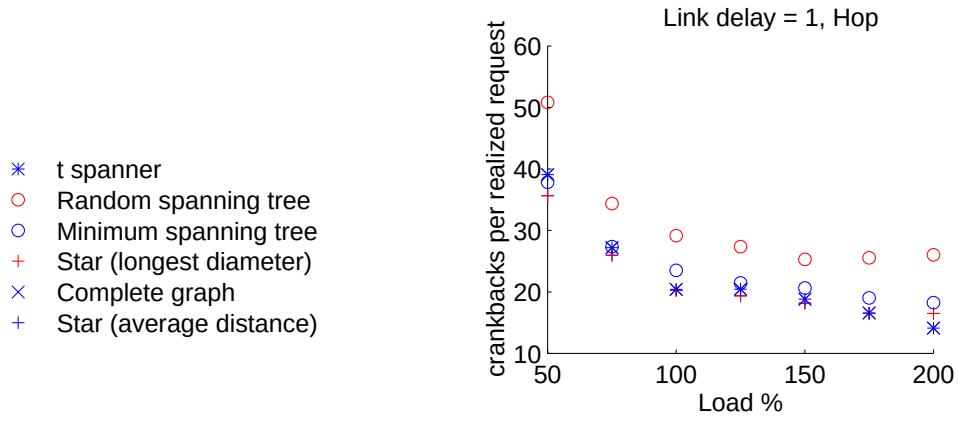


Figure 10: Backtracks in the layered ring topology of figure 1 when minimum hop is used.

The difference in realized flow for this topology is larger than for the previous one. When the link delay is 10, star aggregation scheme has up to 10% less throughput than the other aggregation schemes. RST for this topology performs well. In general, we find that the performance of RST is very sensitive to the topology.

3.1 Comparison to min-hop routing

Figure 11 show a degradation of over 10% in low loads and over 100% in high loads in the throughput when min-hop is used instead of exponential cost metric (on the topology in figure 1). Figure 10 shows an increase of hundreds of percents in the number of backtracks when min-hop is used for routing. It is interesting to see that when min-hop is used, both tree aggregation schemes exhibit a throughput that is up to 15% less than Complete. Star aggregation is only a few percent less than Complete and always above both MST and RST. RST exhibit 10-30% more backtracks that Complete, while MST is only slightly worse that Complete. The rest of the aggregation schemes, Spanner, DIA, and AVE, have similar performance as Complete. However, both for throughput and for crank-backs, the worst aggregation scheme when exponential metric is used performs better than the best aggregation scheme when minimum hop is used (compare to figures 3 and 4).

Figure 13 compares the performance of the two metrics for the simple topology of figure 12. The aggregation schemes for this topology play no role as the only routing decision that has to be made is inside the ring, and at this stage the full topology is known. The Exponential metric has a slight advantage (up to 5%) over the min-hop in throughput. The difference in the number of backtracks per successful connection is 100%-400%.

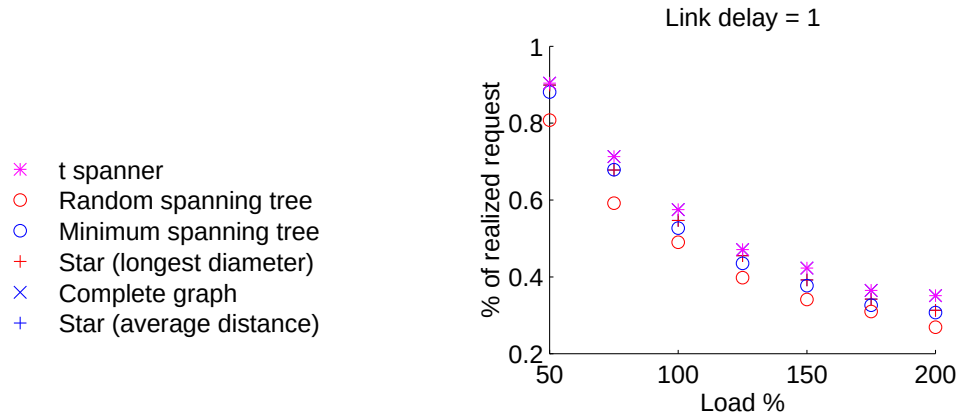


Figure 11: Throughput in the layered ring topology of figure 1 when minimum hop is used.

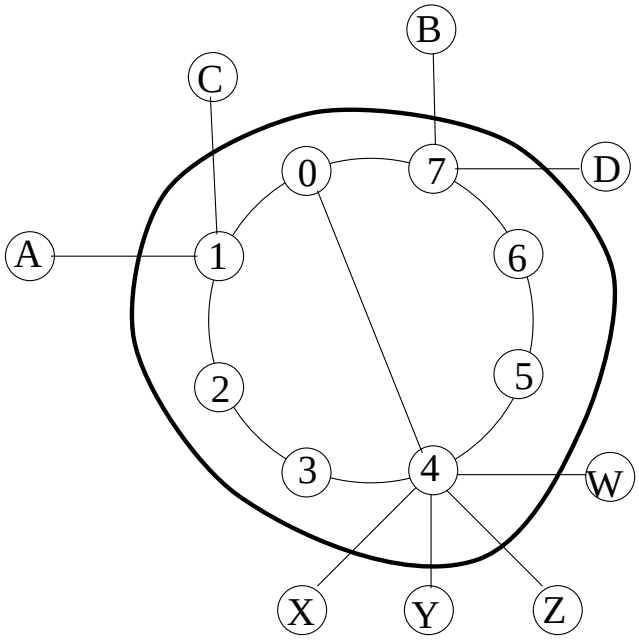


Figure 12: A topology where a ring with a single chord is the only complex node.

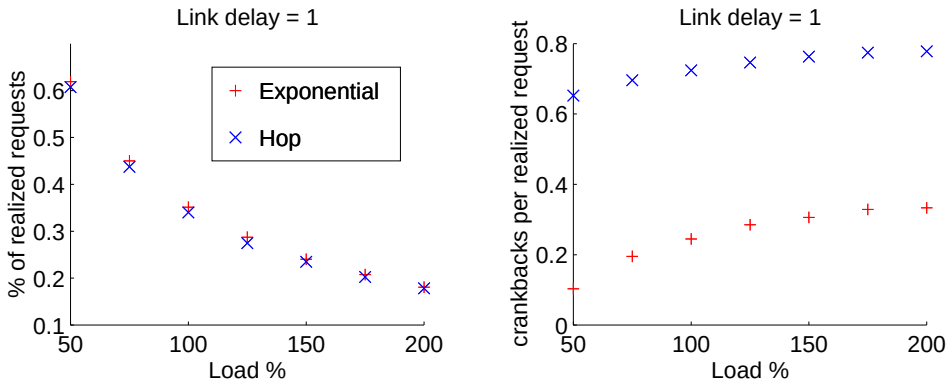


Figure 13: Comparison between minimum hop and exponential metric for the topology of figure 12.

3.2 Re-aggregation policy

In this section we examine the re-aggregation policy, i.e., when one needs to re-aggregate. In all the previous simulations, re-aggregation was performed and the result was broadcast whenever a change in the cost of one of the links occur. This full update approach is clearly impractical, and is used to isolate the effect of re-aggregation in the comparison between the aggregation schemes.

We suggest and test here a more practical approach, called *logarithmic update*. In this approach, the link bandwidth, B, is divided to $\lceil \log B \rceil + 1$ divisions, each doubles the size of the previous. Re-aggregation is performed only when the residual bandwidth in a links crosses the division boundaries. For example, for a link with $B = 16$, the divisions are set to: $[0, 1)$, $[1, 2)$, $[2, 4)$, $[4, 8)$, $[8, 16]$.

We repeated the simulations on the (1, 4)-SR network with logarithmic update. The difference in the throughput for all aggregation schemes between logarithmic update and full update were up to 1.4% but in most cases less than 0.5% (see figure 14 lower right). These differences are within the simulation error. The difference in the number of crank-backs is usually +/-10% with a slight tendency for increase in the number of crack-back for logarithmic updates (see figure 14 lower left). An exception is MST that uses 5-20% more crank-backs when logarithmic updates are used.

The saving in the number of aggregations calculated ranges from less than 40% of the aggregations in light loads to less than 60% in high loads, and about 70% at very high loads (see figure 15 upper-right graph). For both update policies the differences in the number of calculated aggregation between the aggregation schemes is high, too. For logarithmic update, star aggregation require 25-75% more aggregations than Complete, MST, and Spanner; RST requires about 100% more. For full update, the difference in 50% load is negligible among all aggregation schemes with the exception of RST that requires twice as many aggregation calculations. DIA and AVE require about 30% more aggregations than Complete, MST, and Spanner when the load increases.

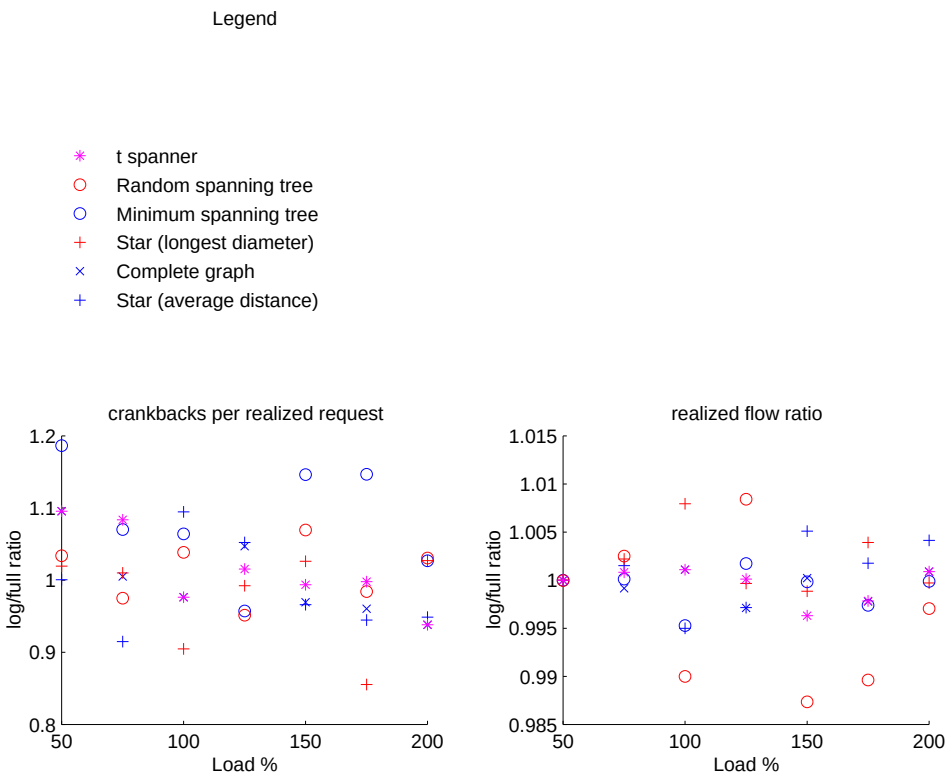


Figure 14: Comparison between the number of crank-backs and the realized flow for logarithmic update and full update re-aggregation policies for the topology of figure 1.

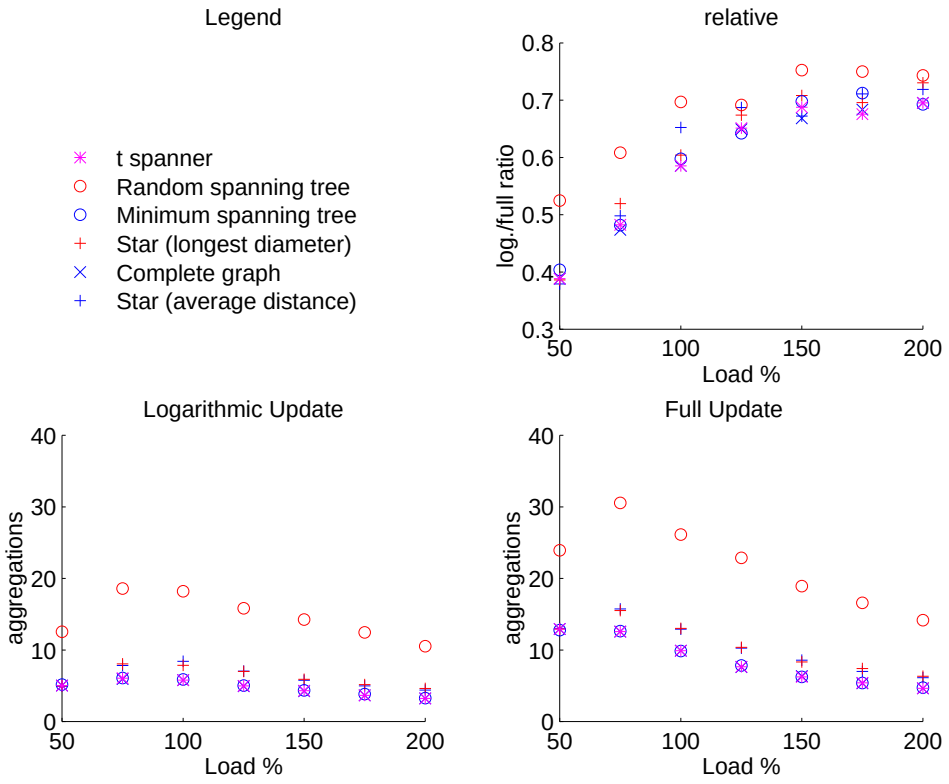


Figure 15: Comparison between the number of aggregations performed when logarithmic updates are used and when full update is used for the topology of figure 1.

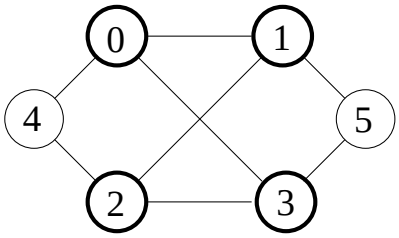


Figure 16: The high level topology. The low level topology of nodes 0–3 is random. Nodes 4 and 5 are simple nodes.

Next we report preliminary results from a two-level topology graph. The higher level, depicted at figure 16 is comprised of four complex nodes and two physical nodes that serve as a source and a destination for the generated flows. The low level structure of the complex nodes were randomly generated as before, only here the minimum and maximum node degrees are 5 and 10.

Figure 17 shows the absolute and relative number of aggregations performed per connection for full and logarithmic update policies. The savings in the number of aggregations from using logarithmic update policy is more than 60%. There is no meaningful difference in the realized flow for all the schemes at all the tested loads. The number of crank-backs for logarithmic update policy increases. However, for Complete, MST, and Spanner this increase is less than 10%.

4 Summary and Concluding Remarks

We study by simulation the performance of various aggregation schemes over a variety of topologies. The study shows that the PNNI approach of using stars to aggregate a network topology may result in a degradation in the network throughput. Star aggregation schemes also increase the number of backtracks and the number of computed aggregations, which in turn results in increased control overhead.

Minimum spanning tree and 2-spanner perform closely to the optimum, i.e., when full information is available. It is interesting to examine the performance implication on t-spanners when the allowed stretch factor is increased. This is also important since the bound [ADD$^+$93] on the size of a 2-spanner is $n\sqrt{n}$, where n is the number of border nodes, which is too costly.

The performance of random spanning tree aggregation is found to be very sensitive to network topology. For some topologies it performs very well, while for others it exhibits very bad performance. Thus, it is not suitable for use.

The good performance of MST aggregation makes a strong case for deploying a tree-based enhancements for the PNNI star aggregation. Such enhancements can be facilitated by using the

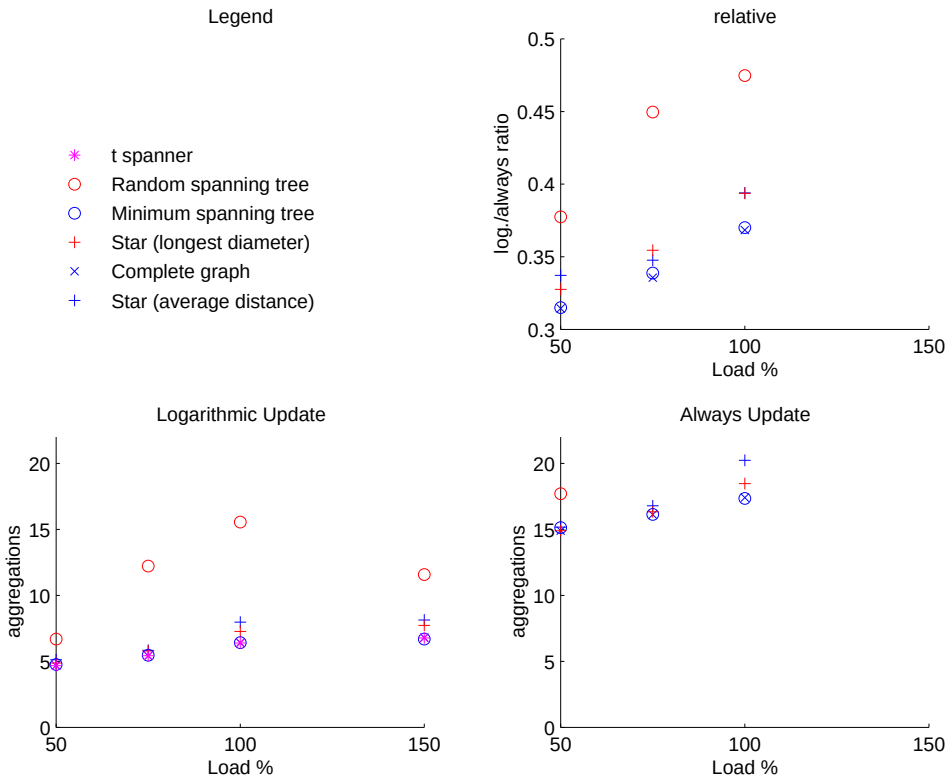


Figure 17: Comparison between the number of aggregations performed when logarithmic updates are used and when full update is used for the topology of figure 16.

exceptions allowed by PNNI [tc96, section 3.3.8]. An algorithm for a tree-based construction within the PNNI aggregation framework (using bypass exception) is described and analyzed by Awerbuch and Shavitt [AS97]. We intend to simulate it in our future work.

This paper shows that performing re-aggregation using our logarithmic update policy reduces the number of aggregations drastically, yet the performance is not compromised. We also show that exponential cost metric results in better performance than min-hop as theory suggests.

References

[ADD+93] I. Althofer, G. Das, D. Dopkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete and Computational Geometry*, 9:81 – 100, 1993.

[AS97] Baruch Awerbuch and Yuval Shavitt. Topology aggregation for directed graphs. Submitted for publication, 1997.

[Bar96] Yair Bartal. Probabilistic approximation of metric space and its algorithmic applications. In *37th Annual IEEE Symposium on Foundations of Computer Science*, October 1996.

[CCS96] I. Castineyra, J. N. Chiappa, and M. Steenstrup. The nimrod routing architecture, February 1996. Internet Draft, Nimrod Working Group.

[CRS96] Israel Cidon, Raphael Rom, and Yuval Shavitt. Analysis of one-way reservation algorithms. *Journal of High-Speed Networks*, 5(4):347 – 363, 1996.

[PS89] David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99 – 116, 1989.

[PU88] David Peleg and Eli Upfal. A tradeoff between space and efficiency for routing tables. In *20th ACM Symposium on the Theory of Computing*, pages 43 – 52, May 1988.

[tc96] The ATM Forum technical committee. Private network-network interface specification version 1.0 (PNNI), March 1996. af-pnni-0055.000.

[Wax88] Bernard M. Waxman. Routing of multipoint connections. *Journal on Selected Areas in Communications*, 6:1617 – 1622, 1988.