Routing Through Networks with Hierarchical Topology Aggregation

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

Yuval Shavitt[§]
shavitt@lucent.com
Bell Laboratories, Lucent Technologies
101 Crawfords Corner Rd., room 4G-627
Holmdel, NJ 07733-3030

Abstract

In the future, global networks will consist of a hierarchy of subnetworks called domains. For reasons of both scalability and security, domains will not reveal details of their internal structure to outside nodes. Instead, these domains will advertise only a summary, or aggregated view, of their internal structure, e.g., as proposed by the ATM PNNI standard.

This work compares, by simulation, the performance of several different aggregation schemes in terms of network throughput (the fraction of attempted connections that are realized), and network control load (the average number of crankbacks per realized connection). The simulation emulate a connection oriented network with a PNNI-like hierarchical source routing algorithm.

Our main results are: • Minimum spanning tree is a good aggregation scheme; • Exponential link cost functions perform better than min-hop routing; • Our suggested *logarithmic update* scheme that determine when re-aggregation should be computed can significantly reduce the computational overhead due to re-aggregation with a negligible decrease in performance.

[*] also with MIT Lab. for Computer Science. Supported by ARPA/Army contract DABT63-93-C-0038, DARPA/Airforce contract F30602961029, NSF Award ID CCR9700157, and ARO contract #DAAH04-95-1-0607

[†] Supported by DARPA/Airforce contract F30602961029

[‡] Partial support from Kaman Sciences Corp. at the Center for Computational Sciences, Naval Research Laboratory, Washington DC.

[§] Corresponding author. Tel: (732) 949 2303, Fax: (732) 949 0399

1 Introduction

Future global networks will consist of hierarchies of subnetworks, or domains[1]. Such domains might, for example, correspond to those portions of the network that reside in a particular geographic region, or they may reflect the network management structure within organizations. Reasons of both scalability and security will mandate that domains not reveal the details of their internal structure to outside nodes [PNN96, CCS96]. Instead, domains will disclose only a summary, or aggregated view, of the costs and availabilities associated with traversing them. Even if one considers scalability issues alone, the need for aggregation is obvious, since the computation and communication complexity of both source and hop-by-hop routing protocols grows at least linearly in the number of links in the network representation.

At one extreme, we might require that an aggregation scheme must not misrepresent *any* information about a domain's internal structure. In this case, the aggregated representations generated would have to include the costs of all possible transits through the domain. Such a representation may be viewed as a complete weighted graph whose vertices are the border nodes of the domain and whose edges represent the costs of the corresponding transits. Unfortunately, for additive parameters, such as delay, the size of such a representation grows quadratically in the number of border nodes, so this scheme, while absolutely faithful to the true structure of the domain, is not scalable enough to be useful in practice. For path parameters that are calculated by using the maximum or minimum functions on the path's link values, such as maximum available bandwidth, a linear representation by a tree is sufficient to represent the full details of the domain [Lee95a, Lee95b]. In this work, we concentrate on additive parameters.

If, however, we relax the requirement that the aggregated representation must be absolutely truthful, then many alternative schemes become feasible. The performance benefits of having a higher-fidelity aggregation scheme need to be balanced against the real-world burdens imposed by the use of larger representations. The burdens imposed by having larger, more accurate representations include greater space requirements for the topology databases within each domain, increased background traffic between domains due to topology updates, and longer computation times for determining the least-cost routes.

The benefit of larger, more accurate representations is principally better route selection. In PNNI [PNN96], for example, a connection setup slated to travel through some domain (based on a source route calculation using *advertised* information) might discover upon arriving in the domain that the intended transit is actually not feasible. When this type of a blockage occurs (due to a discrepancy between truth and advertising, or due to changes occurred since the last advertising) the connection

[1] We use the term domain here to refer to a sub-network that is aggregated, a "peer group" in the PNNI terminology.

setup packet must retrace its steps from the point of blockage back to its original entry point into the domain. The act of retracing one step is called a *crankback*. Once the packet has returned to the point where it first entered the domain, a new route through the domain is computed: If a route is found the connection attempts to establish itself along this path, otherwise the packet must crankback further still. Crankbacks continue until either the network decides that no route can be found, or certain protocol timers expire. Clearly, lowering the frequency of crankbacks is a significant benefit of having higher fidelity aggregation schemes, since it results in faster setup times for connections, and less load on the network control.

Surprisingly, the problem of quantifying the tradeoffs in topology aggregation and determining its effects on network performance has received little attention to date. Bar-Noy and Gopal [BNG90] study such trade-offs with some simple topology representations. There is a large body of theoretical research addressing the problem of compact graph representations [Bar96, PU88, PS89, ADD$^+$93], however, the emphasis of these efforts has been on minimizing the worst-pair distortion of costs[2]. It is not clear *a priori* that low worst-pair distortion is a good criterion for predicting the performance of an aggregation scheme in practice. In fact, the results presented here indicate with certainty that worst-pair distortion is neither the only factor to consider, nor is it the most important one.

There are many possible, yet untested, aggregation schemes. One possible choice has been recently adopted by the Private Network-to-Network Interface (PNNI) group of the ATM Forum. To circumvent the unscalable nature of the complete graph aggregation mentioned above, the PNNI [PNN96] standard specifies that the internal structure of each ATM domain (peer group) be represented as a "star" graph that has one virtual center node (nucleus) and weighted spokes between this nucleus and each of the domain's border nodes. Other possible aggregation schemes might involve advertising a minimum (or randomly chosen) spanning tree of the induced topology on the domain's border nodes. Star and tree representations are desirably compact, since they grow only linearly in the number of border nodes, but unfortunately, both can exhibit very high worst-pair distortion. In fact, the worst-pair distortion of tree and star-based schemes can be linear in the ratio of the maximum to minimum costs of paths in the complete graph representation, and such examples are not difficult to construct.

We seek to determine which aggregation strategies yield representations that are compact, yet still preserve the information about the domain's structure that is critical to the network's overall performance. In this paper, we simulate several aggregation schemes and compare them. Aggregation strategies based on star graphs, spanning trees, and spanners [PU88, PS89, ADD$^+$93] are compared

[2] distortion $= \max_{u,v \in V} d_G(u,v)/d_A(u,v)$, where $d_A(u,v)$ is the least-cost path from u to v in the aggregated representation and $d_G(u,v)$ is the actual least-cost path from u to v in domain.

to the case of perfect aggregation (i.e., a lossless scheme that uses the complete graph). Performance is measured in two ways: network throughput and network control load. **Throughput** is measured either as the cumulative length of connections that were successfully realized, or as the fraction of attempted connections that were realized[3]. **Control load** is represented by the number of crankbacks per realized connection. Note that this is also a measure of the average set-up delay since route recalculation when cranckbacks occur is a time consuming task.

We show that a well-chosen aggregation scheme results in significantly higher network throughput and, even more dramatically, reduces the network control load several fold. Aggregation using the minimum spanning tree (MST) is seen to yield very good overall network performance. This fact, together with the simplicity of calculating spanning trees makes them an attractive candidate for use in practice, e.g., Awerbuch and Shavitt [AS98] suggest a profitable enhancement to the current PNNI standard based on random trees.

In addition, we also investigate the impact of link cost functions, and re-aggregation policy.

A *link cost function* is a mapping from the resources available on the link (e.g. bandwidth) to a real number. Link costs are generally computed during the pre-processing phase of any min-cost route selection algorithm. We compare network performance of traditional min-hop routing (i.e., where the link cost is a constant function) against the performance of min-cost routing under an exponential link cost function (i.e., where the cost of a link grows exponentially with the requested resource consumption). Our experiments show that the latter is superior.

Re-aggregation policy is the set of criteria used by a domain to determine when it should recompute (and re-advertise) a new aggregated representation of its internal structure. We present here a new re-aggregation policy called *logarithmic update*, in which re-aggregation is triggered only when the residual bandwidth within the domain crosses certain thresholds. We set the spacing between these thresholds to grow exponentially. We show that the logarithmic update policy can save half of the computational overhead due to re-aggregation, with minor decrease in performance.

The remaining sections of this paper are organized as follows. First, we will describe the simulation environment. Section 3 contains the results from the simulations conducted. The final section presents our conclusions and outlines future research directions.

2 The simulation environment

We designed and implemented a simulation toolkit with which to investigate routing and performance issues in hierarchical networks[4]. This simulator is a high-level implementation of the PNNI routing

[3] we observed only minor differences between the two formulations.

[4] The simulator is available at `http://www.cnds.jhu.edu/aggregation`.

protocol for ATM networks. The C++ software has a modular architecture, allowing users to readily "plug in" new code and quantitatively assess the impact of different proposed strategies on network performance. For the sake of efficiency in large simulations, all aggregations are carried out by a separate *aggregation server* that accepts the actual topology information from each domain, and then computes the aggregate representation according to the specified scheme. This aggregation server can process multiple requests simultaneously. In addition, a graphical interface is supported to monitor the state of the simulation process over time. We have used this simulation toolkit to study the effects of aggregation schemes, link cost functions, and re-aggregation policy on the network performance, in the context of numerous different network topologies and environments. The specific setup of our experiments is outlined below.

2.1 The model

Connection requests are modeled by a Poisson process, and the connection holding time is exponentially distributed. The inter-arrival time for connection request are exponentially distributed with mean λ. For simplicity we consider normalized inter-arrival time. That is, we use mean $\lambda = 1$ when the traffic load is 100%. Most simulation points represent 10000 requests. In order to be able to use crankback frequency as a measure of network performance, we assume that connection requests are not persistent. In other words, if a connection fails to reach its destination in spite of all crankbacks, it does not re-initiate another attempt[5]. Without this minor assumption crankback frequency could be driven arbitrarily high and would no longer be a meaningful measure by which to distinguish between the effectiveness of different aggregation schemes. Finally, we wish to quantify how the results of our experiments are influenced by environmental parameters. To this end, we define the **load** being placed on the network, as the product of the average request rate and the average connection holding time, divided by the capacity of the average minimum cut (over all source-destination pairs).

There are two measures used to compare the performance of the aggregation schemes:

Throughput — which is measured by the fraction of attempted connections that are realized.

Control Load — which is measured by the average number of crankbacks per realized connection.

Many network attributes can influence the network performance observed in a simulation. First we address the concern of finding an optimal aggregation scheme (from among those described in section 2.2). Another parameter we investigate – that can enhance or suppress the differences among aggregation schemes – is the choice of a link cost function. This function is used implicitly by the

[5] Of course, an identical independent request *may* be generated by the Poisson process for some future time.

routing algorithm when computing the shortest path. We also evaluate the impact of environmental parameters on network performance, giving close attention to link transmission delay, link capacities, and the load placed on the network. Separately, we address the effects of re-aggregation policy choices on network performance.

To determine how each of these parameters affects network performance, we have sought to simulate topologies that best illustrate the tradeoffs in the possible values of the parameters. We also report our experimental results for simulations of randomly generated network topologies, using the standard random graph generation techniques [Wax88], with a modification that limits the maximum node degree. The result was that even graphs with large number of nodes and links had large diameters.

2.2 Aggregation schemes

The following aggregation schemes were simulated:

Complete No aggregation is done. The full cost matrix between the border nodes is advertised.
DIA A star with radius equal to half the cost of the network diameter.
AVE A star with radius equal to half the average cost between (all pairs of) border nodes.
MST A minimum spanning tree.
RST A random spanning tree. For every specific aggregation only a single tree was generated.
Spanner A t-spanner. (A t-spanner is a subgraph that guarantees a worst-pair distortion of at most t. We use $t = 2$ in most of our experiments.)

Table 1 summarizes communication and computational complexities associated with the aggregation schemes, as a function of the number of border nodes, b. All the aggregation schemes are computed on the latest reported topology induced by the border nodes, no averaging was used. The are several reasons not to use the full network topology in calculating the aggregation. The most obvious one is that this way it is easy to encode the aggregation in the PNNI standard using exceptions [AS98]. When security is of concern, this way no details of the internal network structure are revealed. Finally, for aggregation schemes with high calculation complexity, the use of a smaller topology reduces the calculation overhead. It is important to note that aggregation calculation does not come for free, and the calculation cost of an aggregation scheme must be weigh against the advantage of its accuracy.

The performance of the Complete scheme serves as a control variable in our experiments, and is used to assess the loss in network performance due to aggregation. In some cases, especially when the link delay is low, Complete exhibits slightly worse performance than other schemes. This fact, perhaps surprising at first, is due to the on-line nature of the problem. Even an “optimal” (alas near-sighted) decision made by the omniscient Complete algorithm may prove to be very bad in

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

terms of the success of future (yet unknown) connection requests.

2.3 Link attributes

Simulations were carried out using two different link-cost functions:

constant — The link cost remains constant regardless of the bandwidth available on it. Using this link cost function corresponds to min-hop routing when all the links have the same cost. If weights are assigned to the links the result would be weighted min-hop routing.

exponential — The link cost is an exponential function of the residual bandwidth. This cost function stems from queueing theory and opportunity cost analysis: Briefly stated, we desire the link cost to be proportional to the expected queueing delay of a packet across the link, (which is the major factor in the total delay). This 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 analysis suggests maximizing utilization by charging link usage according to a function that increases exponentially as the residual bandwidth decreases.

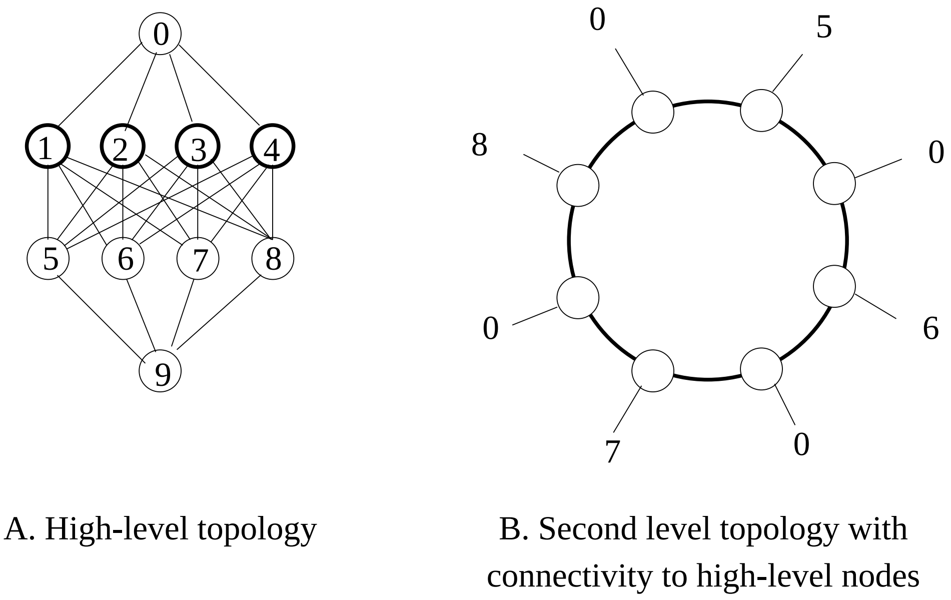
In addition, since network performance was greatly affected by the interaction between concurrent reservations, link transmission delay was a very important environmental parameter in our experiments. When the link delay was low, the network throughput was almost the same, regardless of the aggregation scheme. As link delay was increased, connection requests took longer to setup (due to crankbacks), so larger numbers of connection requests existed concurrently in the network. This induced heavier penalties for poor routing decisions, and consequently magnified the differences in the relative performance of the aggregation schemes.

3 Simulation results

We study the performance of link cost metrics and topology aggregation schemes for hierarchical routing. Using the simulator described in above we simulate the PNNI routing on various topologies and network traffic (connection requests). In the following, we first introduce the topologies we will study. Next we compare the performance of exponential metric and minimum hop metric. Subsequently, we present our results regarding the performance of various aggregation schemes. Finally we consider re-aggregation expenses and suggest an algorithm to reduce this cost.

3.1 Topologies used

We studied both regular topologies and randomly generated networks. The two regular topologies we simulated are depicted in figures 1 and 2. The former figure shows the topology of two level hierarchical network, where the internal structure of some of the nodes is a *staged ring (SR) topology.* The latter, figure 2, is a three level hierarchical topology, where each level is a (similar) multi-stage graph, or *self-similar hierarchy (SSH)* topology. These topologies were designed to emphasize the performance difference between aggregation schemes by inflicting high penalty on routing errors.

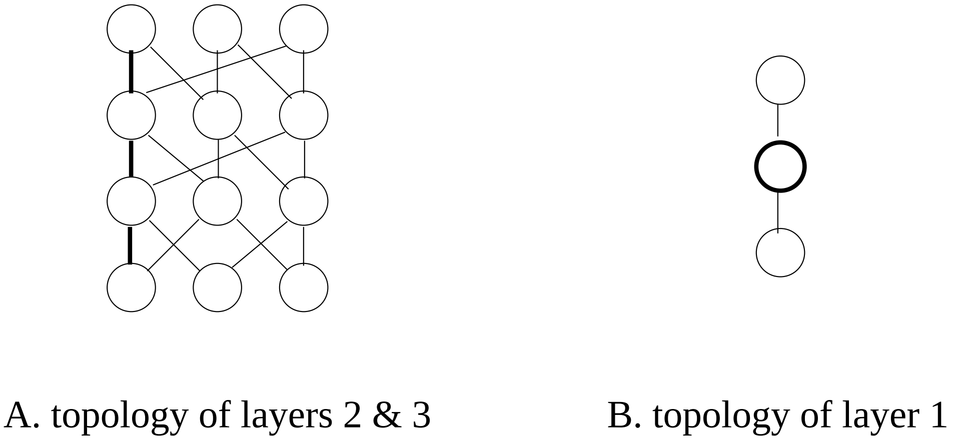


A. High-level topology

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

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

The random topologies we used is a two level hierarchic network (depicted in figures 3 and 4)



A. topology of layers 2 & 3

B. topology of layer 1

Figure 2: A SSH topology. 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 in the same way. Link capacities are 5 for internal links in layers 2 and 3, 10 for **bold** internal links in layers 2 and 3, 10 for links between border nodes, 15 for the **bold** links between border nodes.

generated as follows. At each level, within each domain the nodes are assigned random locations on a grid. There are 7 (or 8) nodes at the higher level domain, and 20 nodes in each lower level domain. Within each lower level domain, four of the 20 nodes are randomly selected, and forced to move to random locations at the periphery of the domain's grid. These four nodes are marked as potential border nodes for that domain (and are indicated using lighter circles in figure 3 and 4). Links are added via a random process that repeatedly generates a random node-pair within each domain, and adds a link between them with probability that decays exponentially with the Euclidean distance between the two nodes [Wax88]. Links that would cause the degree of a node (external links not counted) to exceed four are rejected by the random process to keep the graphs reasonably sparse. The process of adding links terminates when all nodes have a degree of at least two. The higher level links are then assigned arbitrary border nodes in the corresponding lower level domains. Note that at least in one case (subnetwork 002 in figure 4), the algorithm failed to increase the degree of one of the border nodes to two (and reported this problem). Connection request sequences are then randomly generated between pairs of nodes in different lower-level domains.

3.2 Performance of Link Cost Function

Theoretical results [AAP93] suggest that an optimal competitive on-line routing algorithm can be obtained by using an exponential link cost function. We check these results empirically by comparing exponential link cost metric to constant link metric that correspond to min-hop routing.

Figure 6 shows the relative degradation in the throughput (for the staged ring topology), when

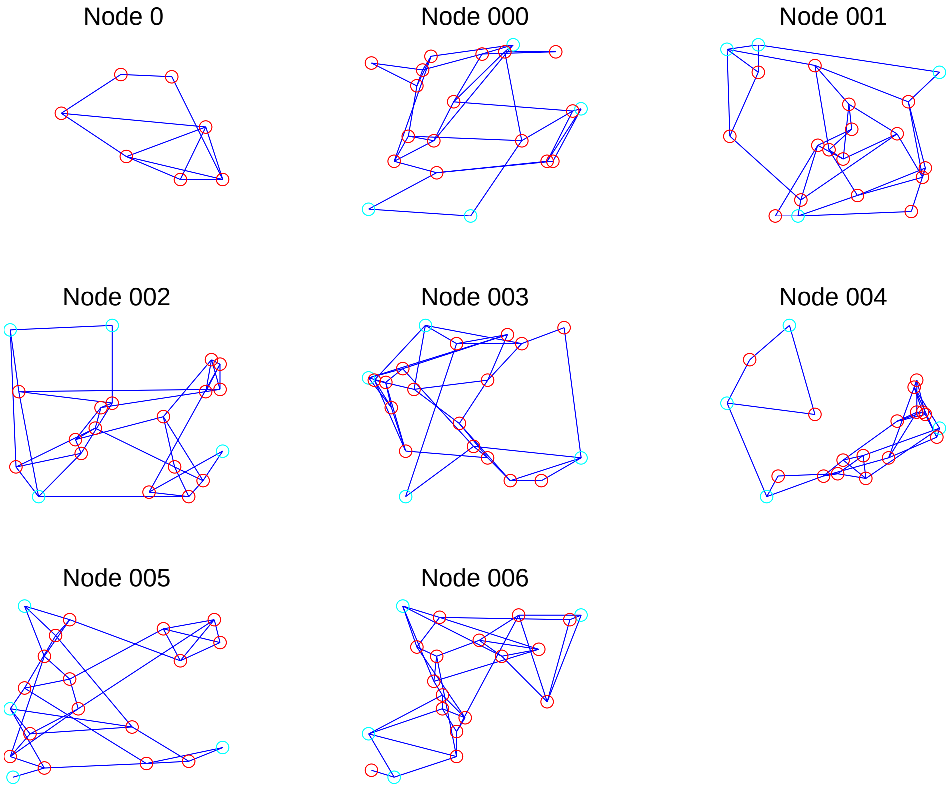


Figure 3: A random topology with seven subnetworks each is comprised of 20 nodes.

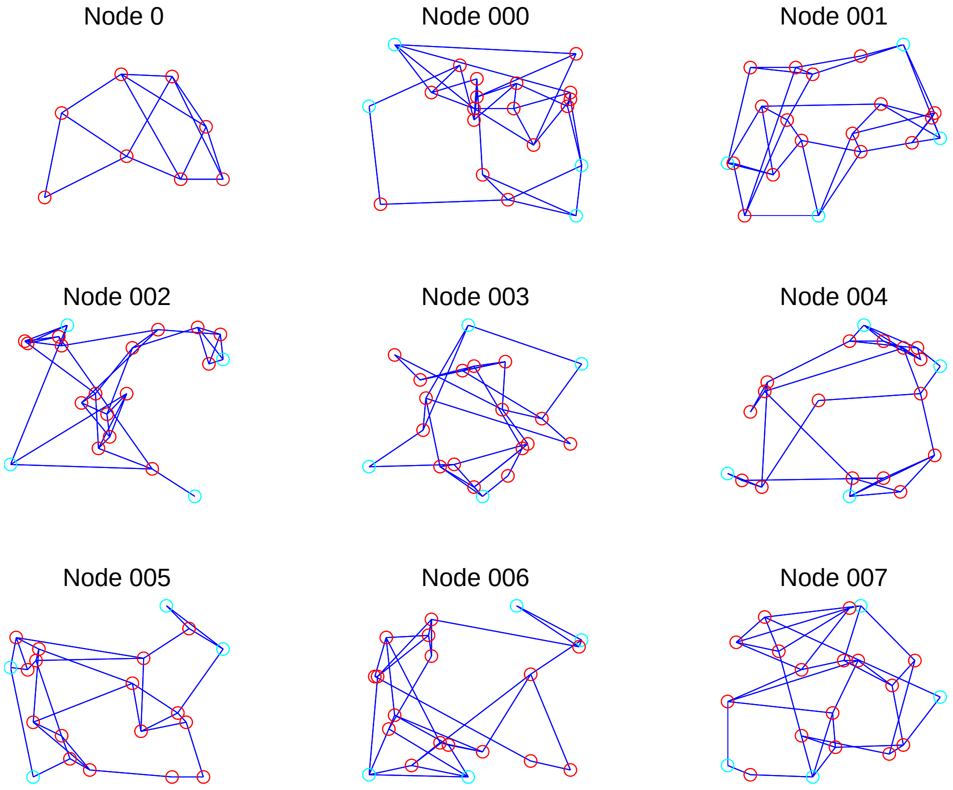


Figure 4: A random topology with eight subnetworks each is comprised of 20 nodes.

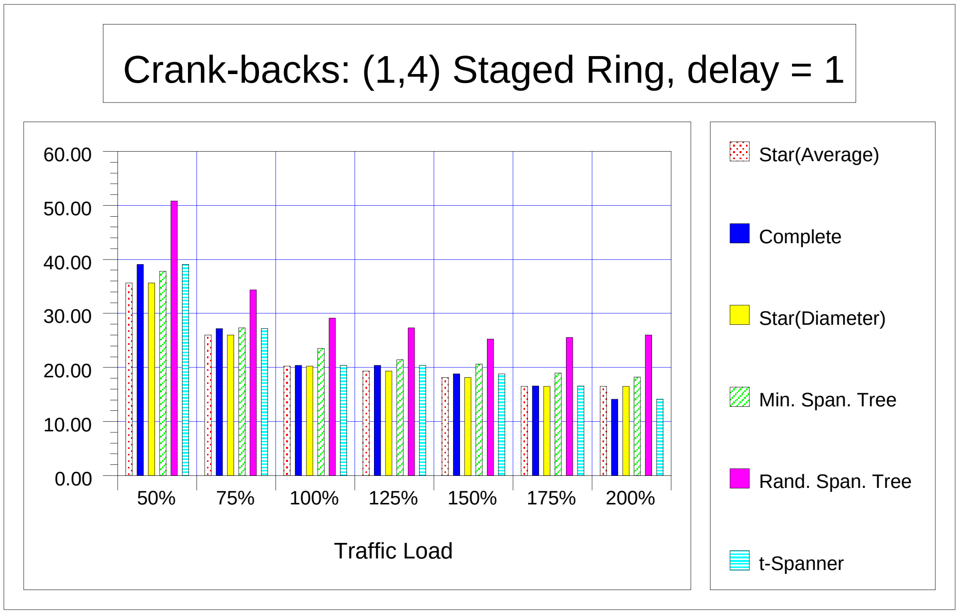


Figure 5: Crankbacks in the staged ring topology of figure 1 when minimum hop is used.

a constant link cost function (i.e., min-hop routing) is used instead of the exponential link cost function. The throughput loss observed is over 10% in low load settings and over 60% in high load settings. Figure 5 quantifies the large increase (several hundred percent) observed in the number of crankbacks when min-hop routing is used. It is interesting to note that with min-hop routing, both tree-based aggregation schemes exhibit a throughput that is up to 15% lower than Complete. The star-based aggregation schemes, however, exhibit throughput that is only a few percent worse than Complete, and the star schemes consistently perform better than both MST and RST. RST exhibited 10-30% more crankbacks than Complete, while MST was only slightly worse that Complete in terms of this measure. The rest of the aggregation schemes, Spanner, DIA, and AVE, perform comparably to Complete in the min-hop setting. We note however that *both in terms of throughput and crankbacks, even the worst aggregation scheme under exponential link cost functions outperforms the best aggregation scheme under minimum hop routing* (comparison against figures 9 and 10).

Figure 8 compares the performance of min-hop and exponential link cost functions for the single level topology of figure 7. Aggregation schemes play no role in this topology, since the routing decisions have to be made inside a single domain (hence the true topology is known). This topology is the simplest one we could design with a penalty against link cost function that are not load sensitive. The exponential link cost function is observed to have a slight performance advantage (up to 5% in throughput) over the min-hop strategy. In terms of crankbacks, the superiority the exponential link cost function is dramatic; the traditional min-hop strategy experiences between

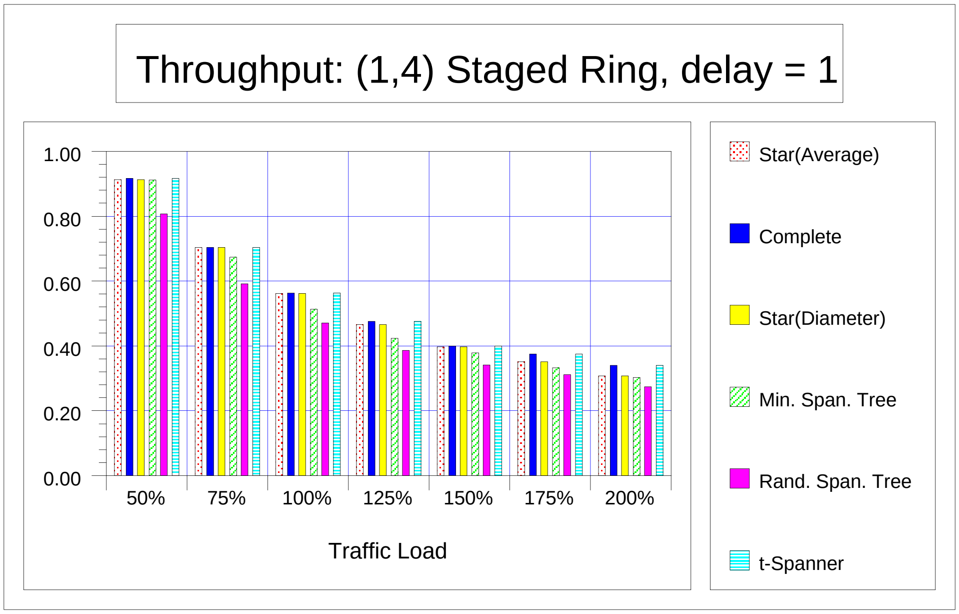


Figure 6: Throughput in the staged ring topology of figure 1 when minimum hop is used.

100%-400% more crankbacks per successful connection.

Based on the above results, we selected the exponential link cost function for the rest of our simulations in order to optimize the performance of the simulated routing.

3.3 Performance of aggregation schemes

In this section, we compare the performance of the different aggregation schemes. We start by reporting the results for the regular topologies, and then report results for randomly generated topologies.

Regular topologies

Staged ring topology. Figures 9 and 10 compare the performance of the aggregation schemes on the topology where rings are connected in stages, as depicted in figure 1. We simulated call requests from a single entry point (node 0 in figure 1) to a single exit point (node 9 in figure 1). Our results indicate that when the link delay is low, there is negligible difference between the aggregation schemes. The only exception to this was RST, for which the throughput was lower by 10% in medium load environments, and lower by 50% in high load environments. As the link transmission delay is made larger, we find that the throughput of star-based aggregation schemes (DIA and AVE) drops to a level that is 10% lower than that of Complete. RST continues to be the worst aggregation

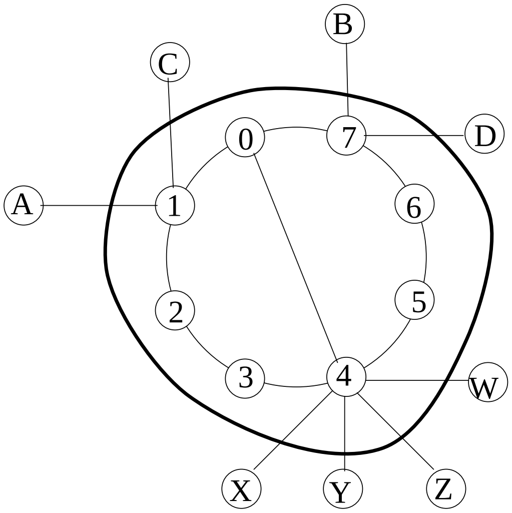


Figure 7: A topology consisting of a ring with a single chord.

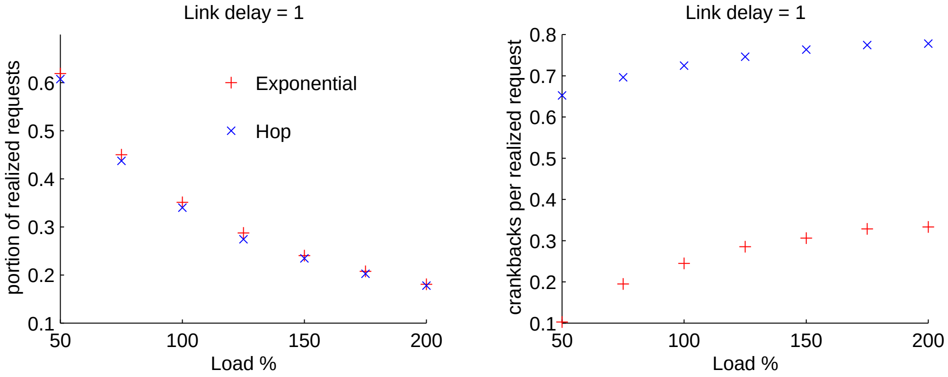


Figure 8: Comparison between minimum hop and exponential metric for the topology of figure 7.

scheme even in this longer link transmission delay setting. The number of crankbacks observed in each of these simulations is illustrated in figure 10. While the relative ranking of the schemes when considering crankbacks is the same as that obtained in our consideration of throughput, the differences between schemes is more pronounced. For RST the number of crankbacks reaches 400% of what is experienced by the Complete scheme, while for DIA and AVE the number is almost 100% higher. We observed no significant differences in either the number of crankbacks or the throughput between the Spanner, MST, and Complete aggregation schemes for this topology.

Self-Similar Hierarchical (SSH) topology. Figures 11 and 12 compare the performance of the aggregation schemes for the SSH topology (whose structure is shown in figure 2). The highest layer of this topology contains three nodes, of which the two extremal ones are physical nodes that are used as source and destination for the flows. The node in the center is a "complex node[6]" consisting of two (recursive) levels of the multi-stage graph shown on the left in figure 2. Under both star-based aggregation schemes, we observed throughput to be almost 10% lower when compared to the other aggregation schemes. More significantly, compared to other schemes, star-based aggregation witnessed over twice the number of crankbacks. There was no significant difference in the performance of the non-star-based aggregation schemes.

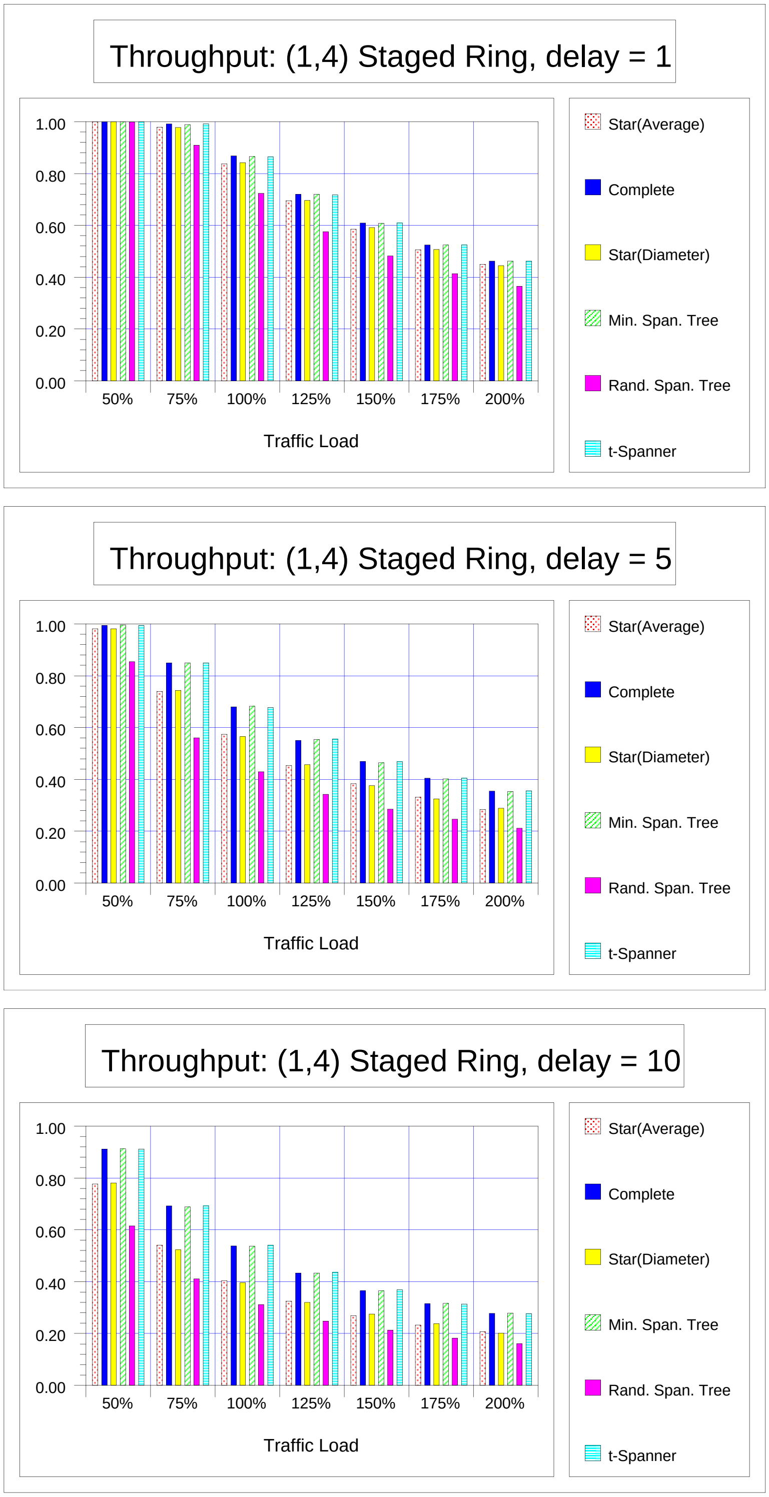
The effect of network diameter

To study the effect of the network diameter on the performance, we generalized the staged ring topology of figure 1, by creating networks that are built using S stages of W rings (where each ring is comprised of $2W$ nodes). We call these networks (S, W) Stage-Rings, or simply (S, W)-SR. The network in figure 1 is an (1,4)-SR network, whereas figure 13 depicts an (2,4)-SR network.

We ran several simulations to assess how network diameter effects the performance of aggregation schemes. In all the SR networks the behavior of the aggregation schemes is similar, in the sense that our performance plots look alike except in the scales of their Y-axes. As the network diameter is made larger (by increasing the value of S), the number of crankbacks is seen to increase. Interestingly, the rate at which the number of crankbacks increases is super-linear in the number of stages, S. Simultaneously, we observe that the total throughput of the network declines with larger values of S, because the mutual disturbances between pending connection requests increases.

If on the other hand, we increase W (it is not hard to see that the diameter of the network does not change as a result of this) we observe a sharp increase in the number of crankbacks. This sudden deterioration can be explained by the heavier cost penalties of selecting a bad entry point into a ring,

[6] In the PNNI specification a higher level node that represents an entire lower-level domain and maintains its aggregate representation is referred to as a *complex node.*



Throughput: (1,4) Staged Ring, delay = 1

Throughput: (1,4) Staged Ring, delay = 5

Throughput: (1,4) Staged Ring, delay = 10

Figure 9: Throughput in the staged ring topology of figure 1.

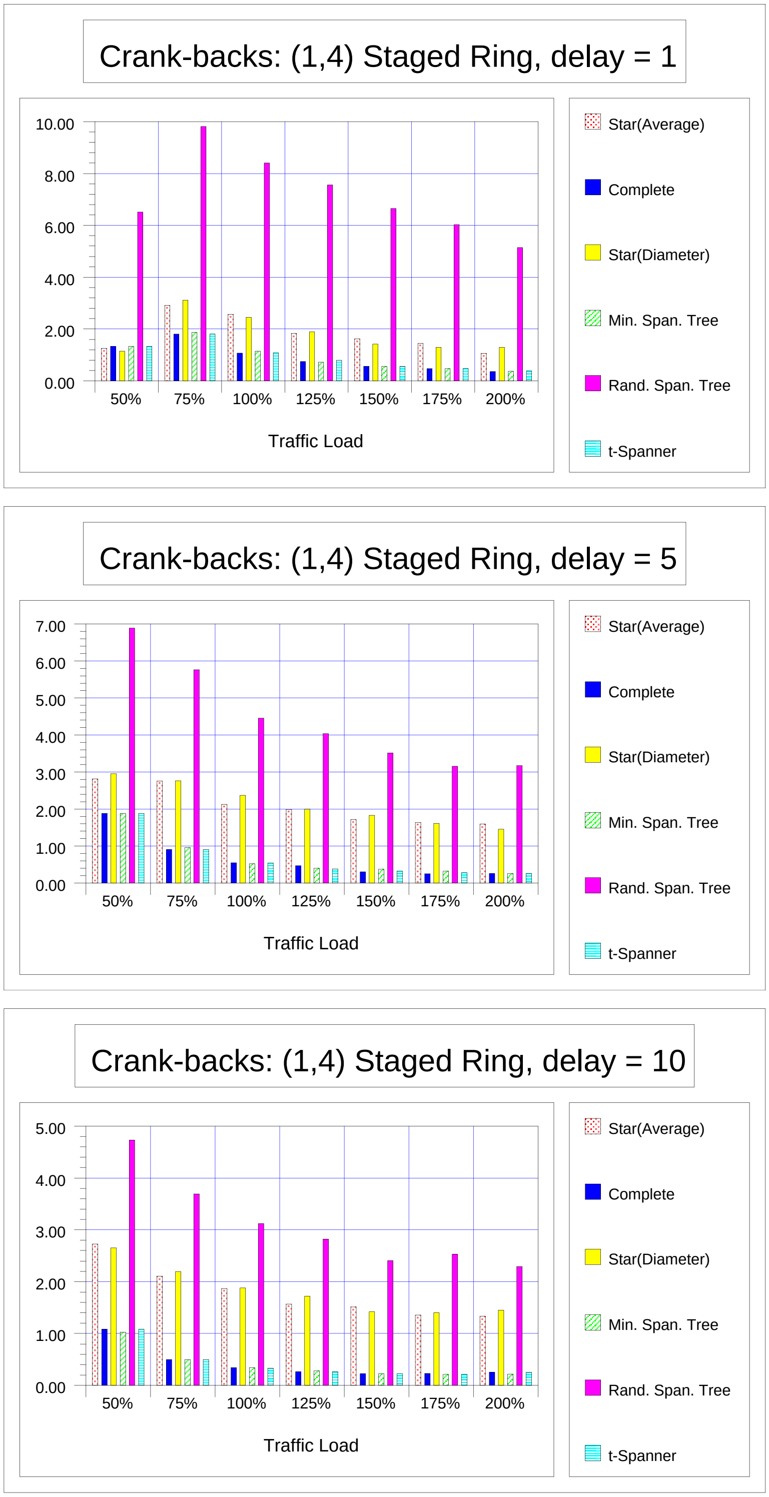


Figure 10: Crankbacks in the staged ring topology of figure 1.

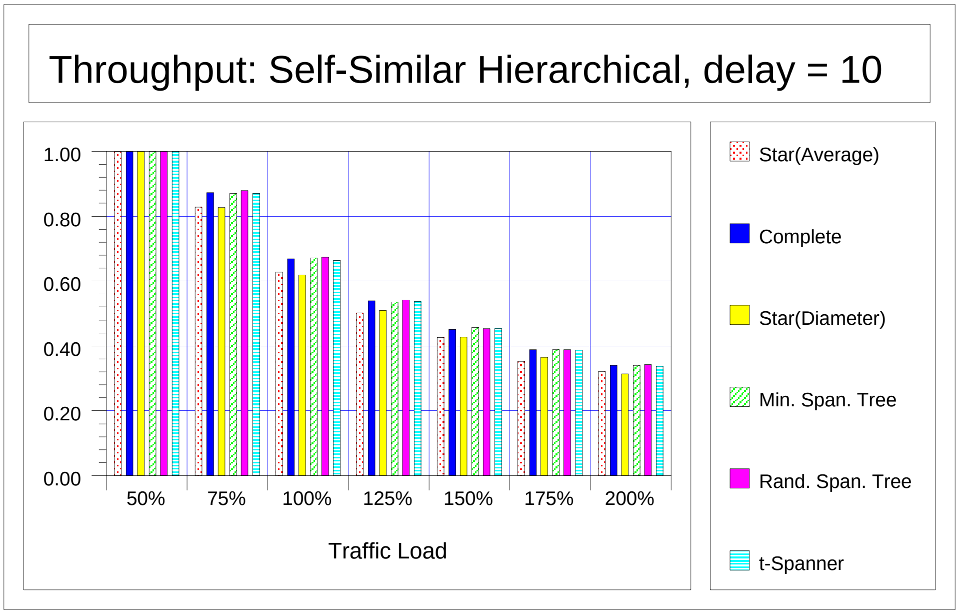


Figure 11: Throughput in the SSH topology of figure 2.

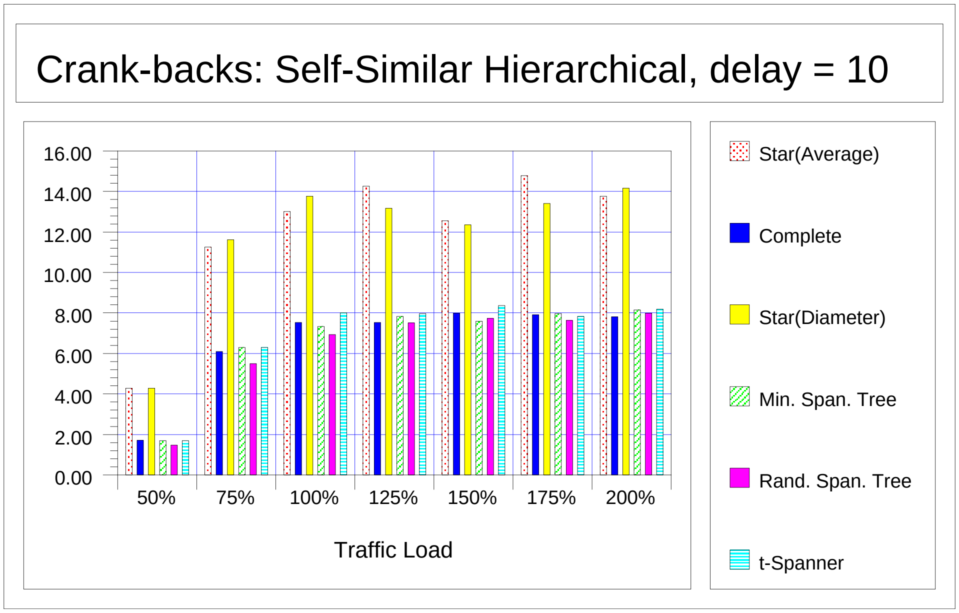
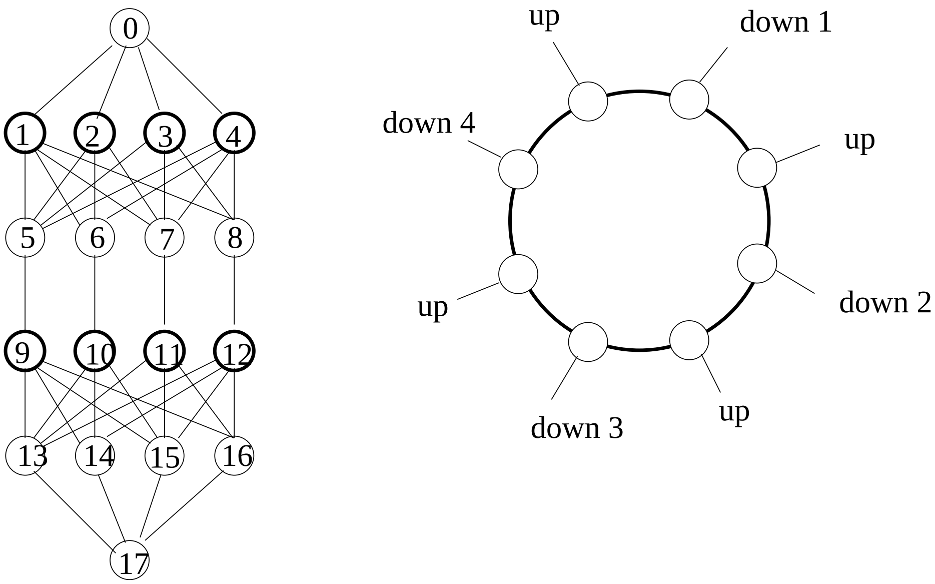


Figure 12: Crankbacks in the SSH topology of figure 2.



A. High-level topology

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

Figure 13: An (2,4)-SR network.

and also by the increased probability for any connection setup to experience an en-route rendezvous with other concurrent (competing) reservation requests.

While the number of crankbacks increases with the diameter (i.e., as we increase S), the relative differences between the performance of the aggregation schemes decreases. For example, when $S = 1$ the ratio between the star-based aggregations and the MST, Spanner, and Complete schemes is 2 at low load and over 5 in high loads. When $S = 2$ this ratio is becomes about 1.5, and for $S = 3$, it reduces further to about 1.4. We note that for $S = 1$ the difference between aggregation schemes increases as the load is increased, while when $S > 1$ the difference between schemes' performance remains constant, regardless of the load. This can be explained by the fact that the majority of requests begin to fail when the first stage of rings saturates, and once this occurs the flow cannot increase, even if the rate of connection request arrivals is increased. This phenomenon of bottleneck behavior was also observed by Cidon et al. [CRS96] for one-way reservation schemes.

Randomly generated topologies

Randomized I. Tables 2 and 3 summarize the simulation results for the network depicted in figure 3. The throughput observed for the different aggregation schemes generally varied by less than 1.5%. The only exception was RST, which under-performed the Complete scheme by almost 9% when the

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: Crankback at load 100% for the random topology of figure 3

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 3

link delay was 10. Interestingly, when the delay was 10, DIA was better than Complete by 1.4%. This small difference could be attributed to the on-line nature of the problem; even a decision that is optimized relative to complete knowledge of the past and present state of the network, is not guaranteed to be favorable in terms of the demands of future (yet unknown) requests.

A more pronounced difference between the aggregation schemes was observed in terms of crankbacks. Star-based aggregations required 8.5-16.5% more crankbacks than Complete to achieve similar throughput. On the other hand, MST performed slightly better than Complete, whereas RST required almost 25% more crankbacks and still achieved very low throughput.

Randomized II. Tables 4 and 5 summarize the simulation results for another similarly generated two level network, which is depicted in figure 4. The results for crankbacks are similar to those obtained for the previously described topology. In examining the number of crankbacks for AVE, DIA, and RST versus the number experienced by Complete, MST, and Spanner, we note that the difference is usually in the double digit percent region (although AVE is a little better than DIA, and RST).

The difference in throughput of the schemes was more pronounced for this random topology

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: Crankback at load 100% for the random topology of figure 4

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 4

than for the previous one. When the link delay was 10, star-based aggregation experienced almost throughput levels almost 10% lower than the other schemes. RST performed well on this topology– but in general, we found the performance of RST to be very sensitive to the topology.

3.4 Re-aggregation policy

In this section we examine the impact of *re-aggregation policy*, i.e., the criteria that trigger re-aggregation and re-advertisement of domain topology. In all the previous simulations, each domain performed re-aggregation of its topology whenever there was any change in the bandwidth availability of its constituent links. This *full update* approach is clearly impractical, but was employed deliberately so as to be able to distinguish the impact of the aggregation schemes, from the effects of re-aggregation policy. We present our findings concerning the latter important issue in this section.

We propose (and present our experimental assessment) of a more practical alternative to full update, which we call the *logarithmic update* re-aggregation policy. In our approach, the total bandwidth B on each link, is divided to $\lceil \log B \rceil + 1$ blocks (each double the size of the previous). Re-aggregation is performed only when the residual bandwidth in a link crosses these division boundaries. For example, for a link with total capacity $B = 16$, the divisions are set to: [0, 1), [1, 2), [2, 4), [4, 8), [8, 16]. For such a link, re-aggregation would be performed when the utilization of the link crosses the values 8, 12, 14, 15, and 16.

We repeated our simulations of the (1, 4)-SR network using logarithmic update. Regardless of aggregation scheme, the differences in throughput achieved by using logarithmic update versus full update were never more 1.4%, and in most cases the difference was less than 0.5% (see lower graph of figure 14). These differences are within the simulation error. The difference in the number of crankbacks is usually ±10% with a slight tendency for an increase in the number of crankbacks when using logarithmic updates (see upper graph of figure 14). An exception is MST that experiences 5-20% more crankbacks when logarithmic updates are used.

The advantage offered by the logarithmic update re-aggregation policy is the significant reduction of computational overhead due to aggregation calculations, which translates to a reduction of the topology updates required to be disseminated. Specifically, in light load settings, logarithmic update required only 40% of the aggregations required by full update. Even under heavy load conditions this figure stayed below 60%, and never became higher than 70% even in very high load environments (see figure 15). In our comparison of re-aggregation policies, we note that the precise difference in the number of aggregations triggered does depend on the aggregation scheme under consideration. For logarithmic update, star-based aggregation required 25-75% more aggregations than Complete, MST, and Spanner; RST requires about 100% more. In a full update policy and light load levels, the difference among the aggregation schemes is negligible, with the exception of RST that requires twice as many aggregation calculations. As the load is increased DIA and AVE require about 30% more aggregations under the full update policy, when compared to Complete, MST, and Spanner.

4 Summary and Concluding Remarks

Using simulation, we study the performance of several aggregation schemes on a variety of topologies. Our research shows that the vanilla PNNI approach of using star representations for network aggregation may result in unacceptable degradation in the network throughput on certain topologies. We also find that star-based aggregation schemes result in a larger number of crankbacks and require more frequent re-aggregation, which results in higher control overhead.

The minimum spanning tree and 2-spanner aggregation schemes performed close to the optimal, i.e., comparably to what could be done if full information was available. It would be interesting to investigate the performance of t-spanner aggregation when permitted stretch factor t is increased. This would also be important since the best bound [ADD$^+$93] on the size of a 2-spanner is $O(n\sqrt{n})$, where n is the number of border nodes. This is too costly to be scalable, but perhaps larger values of t would result in smaller representations without an unacceptable loss in performance.

The random spanning tree was found to be very sensitive to network topology. For some topologies it performed very well, while for others it underperformed significantly. We conclude, because of its inconsistency, that it is not suitable for use in practice.

The impressive overall performance of the MST aggregation scheme presents a strong case for proposing an extension to the PNNI standard. MST can already be implemented by using the exception (border-to-border) link mechanism in the complex node representation of PNNI [PNN96, section 3.3.8]. Lee [Lee95a, Lee95b] showed that MST is optimal for path cost functions that are based on the maximum or minimum values of the path's link cost values, such as, maximum bandwidth. An algorithm for another tree-based construction within the guidelines of the PNNI aggregation specification is described and analyzed in [AS98]. We intend to simulate the performance of this construction as part of our future work.

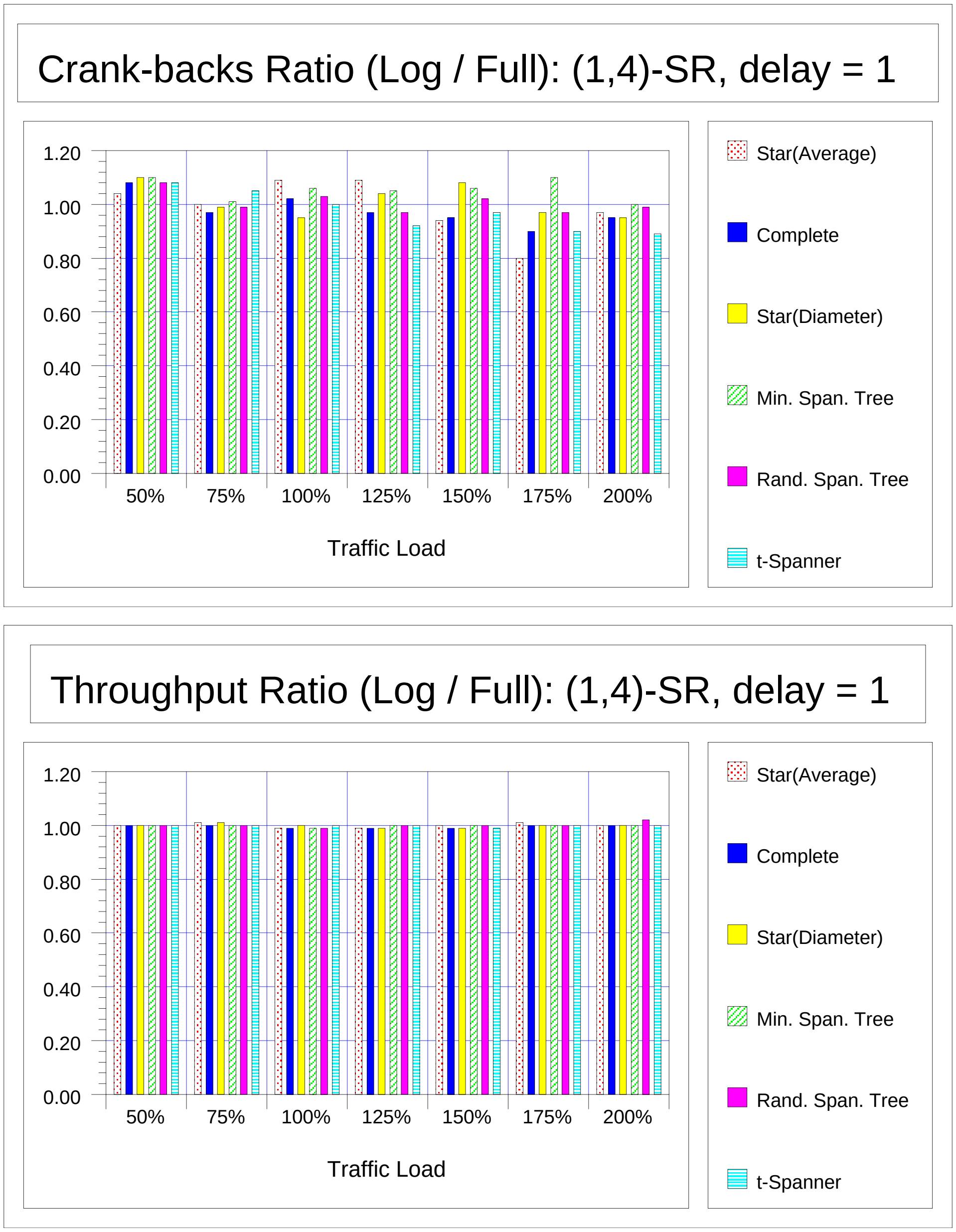


Figure 14: Comparison between the number of crankbacks and the realized flow for logarithmic update and full update re-aggregation policies for the topology of figure 1.

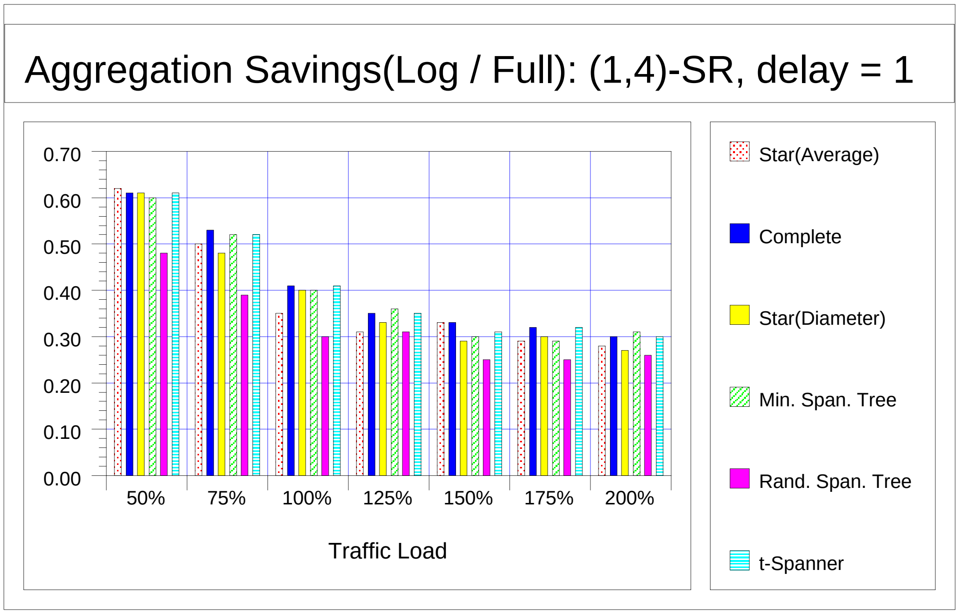


Figure 15: The reduction in the number of aggregations performed when the logarithmic update policy is used instead of the full update policy for the topology of figure 1.

We demonstrate that using our proposed logarithmic update policy reduces the number of re-aggregation computations drastically, while not compromising network performance in any significant way. Finally, we provide compelling empirical evidence that the modern breakthroughs in the theory of exponential link cost functions can be harnessed and yield network performance levels that are superior in practice to what is presently afforded by traditional min-hop routing protocols.

It is important to stress that our simulation model assumes connection-oriented networks, and thus the presented results cannot be all directly applied to datagram networks such as the Internet. In particular, adaptive shortest path routing of the sort we simulated may result in oscillation and instability [BG92, section 5.2.5] in datagram networks with destination based routing.

Clearly, there are still many open questions related to hierarchical routing and topology aggregation. Some future research directions are summarized here. It is interesting to investigate whether aggregation is better if performed on the topology induced by the border nodes, or can better aggregation result by using even the same algorithm on the original topology (e.g., spanning tree). An important open question is how can multiple cost functions can be aggregated together [Lee95b], e.g., to achieve both minimum delay and minimum jitter. This will facilitate efficient QoS routing algorithms. The simulations conducted here all assume symmetric reservation along both directions of a link. However, many applications require and the PNNI standard supports asymmetric reservation. Simulation of asymmetric aggregation schemes like the one proposed by Awerbuch and Shavitt

[AS98] is of great value [Lee95b].

Acknowledgment

The authors are thankful to the anonymous reviewers for their constructive comments and suggestions that helped to improve the presentation of the paper.

References

[AAP93] Baruch Awerbuch, Yossi Azar, and Serge Plotkin. Throughput-competitive on-line routing. In *34th Annual IEEE Symposium on Foundations of Computer Science*, pages 32 – 40, October 1993.

[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.

[AS98] Baruch Awerbuch and Yuval Shavitt. Topology aggregation for directed graphs. In *Third IEEE Symposium on Computers and Communications*, June 1998.

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

[BG92] Dimitri Bertsekas and Robert Gallager. *Data Networks.* Prentice Hall, second edition, 1992.

[BNG90] Amotz Bar-Noy and Madan Gopal. Topology distribution cost vs. efficient routing in large networks. *Computer Communications Review*, 20(4):242 – 252, 1990.

[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.

[Lee95a] Whay Chiou Lee. Spanning tree method for link state aggregation in large communication networks. In *IEEE INFOCOM'95*, pages 297 – 302, April 1995.

[Lee95b] Whay Chiou Lee. Topology aggregation for hierarchical routing in ATM networks. *Computer Communication Review*, 25(2):82 – 92, April 1995.

[PNN96] Private network-network interface specification version 1.0 (PNNI). Technical report, The ATM Forum technical committee, March 1996. af-pnni-0055.000.

[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.

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