The Effect of Network Hierarchy Structure on Performance of ATM PNNI Hierarchical Routing

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

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

Abstract

Networks deploying hierarchical routing are recursively partitioned into sub-networks that do not reveal the full details of their internal structure outside their domains. Instead, an aggregated view of certain parameters that are associated with traversal within such sub-networks between their border nodes is advertised. The ATM PNNI standard and the Internet Nimrod architecture both adopt this approach for routing.

This paper studies the effectiveness of ATM hierarchical routing protocols on networks with different hierarchical structures by simulation. Our study shows that, in general, the hierarchical source routing performs well compared to the global routing strategy which imposes no hierarchy, while utilizing less storage and communication overhead. For certain networks and topologies, the hierarchical routing performs better than the global routing. Different hierarchies imposed on the same topologies have significantly different performance on the throughput and routing delay. This suggests the necessity of studying the hierarchy design for communication networks using hierarchical routing.

1 Introduction

Routing algorithms can be divided into two major classes: distance vector algorithms [11, 10] and link state algorithms [12, 13]. In distance vector algorithms routers maintain tables with an entry per destination network where the route cost and the next hop to the destination are kept. These tables are exchanged between immediate neighbors, thus the propagation of information about changes in route costs might be slow. It has been shown [7] that distance vector algorithms suffer from slow recovery from link failures, and during the slow recovery time routing loops may occur. In link state algorithms, network routers flood their emanating link costs to the network. Next hop routes are calculated based on a full topology database maintained in each router. Link state algorithms do not exhibit the slow recovery time occurring during link failures.

Both algorithm classes suffer from scalability problems: in distance vector algorithms table size grow linearly with the number of nodes; in link state algorithms communication cost grows with the number of links, and so does the space requirement for topology maintenance. To cope with scalability, hierarchical routing was suggested [1, 8]. In this approach a network is grouped into hierarchy of nodes at various levels. Each node at a high level reports information for underneath topology to other interested nodes. Each node also receives reported routing information from nodes in other domains and exchange them information to other lower level nodes. As a result, each node needs only to maintain partial information about the entire network. Requirement on routing information storage at each node is thus substantially reduced. ATM PNNI [1] routing standards and Internet Nimrod [8] routing protocols both adopts such strategy.

Domains typically correspond to a management organization or a geographical region. We need to either reduce the size of the representation of the structure of these domains, or to hide the details of internal struc-

[*] 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.

[†] Corresponding author. Supported by DARPA/Airforce contract F30602961029.

ture for these domains for security reasons [1, 8]. Instead of the full information for the internal structure, a summary, or *aggregation* of routing information for the domain is maintained and reported by the domain. To achieve scalability for routing protocols, aggregation is needed. This is because the complexity of routing protocols grows at least linearly with the number of links in the network for both source routing and link-by-link routing. Many ways to aggregate routing information, called *aggregation schemes*, exist with various tradeoffs on representation size and representation accuracy for routing information.

In this paper we study the performance of hierarchical routing for various link cost metrics and aggregation methods. We are also interested in understanding the performance implication of different hierarchy embedding upon a same network graph to study the effectiveness for hierarchical routing. We choose ATM PNNI standard [1] as representative for the study, since PNNI routing and signaling protocols are mature and its performance has significant implication on the future deployment of ATM networks. We perform simulation studies to compare the performance of both the global routing and hierarchical routing strategy, in terms of their network throughput and control overhead such as signaling delay.

We show that, hierarchical routing could be as effective as global routing when exponential link cost metric is used, though it could incur slightly higher signaling delay. For some topologies, imposing hierarchies would lead to improved routing performance than global routing. We also observe that an increase on the levels of hierarchy does not necessarily lead to better performance. This implies that different hierarchies for same topology could have much different performances. Therefore it is important to study hierarchy designs for hierarchical routing. With saving in storage for routing information and communication overhead, and its capability for security and scalability enhancement, hierarchical routing provides an effective and scalable routing solution for future global teranode networks.

The rest of the paper is organized as follows. Background on hierarchical routing and network hierarchy structures is introduced in section 2. Section 3 discuss the simulation environment we develop for studying ATM PNNI routing. Simulation parameters under study are detailed in section 4. Section 5 presents the result for the performance study. Section 6 gives the conclusion.

2 Background: hierarchical routing

In this section, we introduce hierarchical routing with topology aggregation to reduce the topology representation expense. Also discussed is background on the hierarchical routing performance and our focus on the effect of hierarchy structures in hierarchical settings; that is, how network hierarchy structures would affect performance of the hierarchical routing.

2.1 Hierarchical routing with aggregation

In this paper we consider *source routing* in general, which is of the broad link-state routing category. To realize a connection request, source node receiving the request would use local routing information maintained by itself to make decision as to which path the connection would take. Routing information for a link is usually presented to routing protocols in the form of a cost metric value. An routing algorithm is executed to use all these link cost metric values to find a feasible path for the connection request. In fact, shortest path algorithm, such as Dijkstra's Algorithm, is widely used for this purpose.

Note that if information about all physical links in the routing domain is maintained at each node, a physical path can always be found for a connection request by the source node. This style of source routing is called *global source routing*. The found path is not necessarily feasible (in terms of resources for all path links for the request), though. A signaling process is invoked to traverse all links on the path according to their order on the path, while reserving bandwidth on the link which has enough bandwidth to support the request. When a link can not support a request, a backtrack (or *crank-back*) is needed to go back to previous node and try another link. This routing is also called *link state* routing and is the approach adopted by Internet routing protocols.

However, often it is not feasible that a source node maintain full information for all physical links. A node can maintain processed, or *aggregated*, information about the network. Using these information, source can find a sequence of partial paths at different abstraction levels, while they together consist of a path for a connection request. The successful establishment for each partial path in this sequence thus realize a route of physical links for the request. These paths are usually in different levels in a topology hierarchy, as this kind of routing is called *hierarchical source routing*. This is the approach deployed by ATM PNNI standard

[1]. After the partial paths are determined, a *signaling* process is invoked to traverse links on these path to reserve bandwidth on those links that have enough bandwidth to support the request. When a link can not support a request, a backtrack (or *crank-back*) action makes the traverse back to the previous node and try another link from there.

These two styles of source routing both require each node to maintain a view of the entire routing domain, though they differ in the degree of view details to be maintained. Besides routing information storage, there is network overhead incurred when state change for each link is transmitted to nodes that maintain relevant information. Communication overhead is thus incurred to update nodes with the up-to-date information. Source routing requires each node to store its own link state routing information, and the network must support link state update among nodes. The routing information storage and the communication cost are the major expenses for these routing protocols.

Routing information advertised by a node to others is aggregated by *topology aggregation method* to reduce routing information storage in nodes and topology information exchange, the main reason why hierarchical routing is scalable. Topology aggregation essentially derives a simpler connectivity graph out of a much more complicated real connectivity graph. Various topology aggregation methods exist, among them are tree scheme [6] and spanner scheme [2, 14].

2.2 Effect of the hierarchy structures

We study the effect of network hierarchy structures on the hierarchical routing performance where various aggregation methods are used. That is, we consider how various hierarchies imposed on the same topology would affect the performance of hierarchical routing. We use ATM PNNI routing protocol in the study.

Aggregation methods intend to reduce the topology representation expense for large networks through the process of summarizing the network topology. This process would nonetheless affect the routing performance in these networks. While aggregation reduces the representation size for a single domain, various hierarchy structures imposed on the same lowest level network connectivity would lead to difference on the topology representation expense for the network. In this paper we focus on the performance implication of hierarchy structures on the hierarchical routing without considering the topology representation expense introduced by these structures. Work in [5] studies the relationship between the topology distribution cost and the efficiency of routing in large networks.

Note that, aggregation methods introduce inaccuracy on routing information as a summarized connectivity is derived. Researches on how routing information inaccuracy would affect routing performance have been performed. In [9], routing connections with QoS requirements across one or more networks is investigated under the situations that the information available for making routing decisions is inaccurate and expressed in some probabilistic manner. The main focus of this study is to determine the impact of such inaccuracies on the path selection process to identify the path that is most likely to satisfy the QoS requirements. Our study focus on the experimental investigation of aggregation methods and hierarchy structures on such network performance measures as the network throughput and crankbacks; namely, we investigate the impact of hierarchy structures on the performance of routing algorithms when aggregation methods are deployed.

3 STARS: An ATM PNNI simulator

We have implemented **STARS** [4], a simulator for ATM routing to study how aggregation methods affect performance for hierarchical routing [3]. **STARS** is an object-oriented discrete event simulator written in C++ that emulates PNNI routing protocols over hierarchical structured networks. It has a modular architecture that enables easy integration of different modules to achieve various functionality. For example, an generic interface of routing link cost function is provided to permit insertion of various cost functions for study. Similar interface for aggregation methods for domain routing information summary and topology graphs are also provided. Below we briefly discuss the design of STARS. More details on the design and implementation of **STARS** can be found in [4].

STARS consists of the following components: Simulation Engine, Topology Aggregation Server, Network Topology Generator, Network Traffic Generator, Statistics Collector, and Graphical Display.

Simulation engine adopts the shortest path algorithm using certain link cost functions to select feasible paths. Many existing routing schemes fit well with this formulation. For example, the so-called min-hop scheme used in Internet routing can be realized by a shortest path algorithm with a constant link cost. On the other hand, link cost functions that depends on other link characteristics such as link utilization and link capacity would capture more information and promise better performance. Theory suggests that, there exists on-line routing algorithms based on exponential cost link which achieve optimal performance in terms of throughput against off-line algorithms. To facilitate such study, our simulator allows easy incorporation of different link metric functions.

Aggregation information for domains is re-computed whenever a "significant" change occurs in the topology parameters. Such changes justify an aggregation for obtaining the new topology information. *Aggregation server* accepts a topology and an aggregation method and returns the aggregation metric resulted from the aggregation calculation. Besides providing a capability for direct calls to aggregation functions (which is executed sequentially on one machine), **STARS** can run aggregation servers on several other machines to alleviate loads on one computer for improved performance.

Network topology generator produces topologies for study. Network structures are generated with certain graph properties such as number of nodes, number of edges, average degree of nodes, border nodes (for hierarchy), inter-connection pattern between domains (for hierarchy). Network traffic generator generates dynamic connection requests to **STARS** to drive the simulation. A request is represented by its staring time, its source and destination nodes, and its traffic bandwidth requirement. Connection request sequences with various arrival/departure distribution and bandwidth requirement can be specified as input to **STARS**. Statistic collector is used to collect data for further analysis. Graphical display shows the routing signaling progress for the current connection request being established.

In the next section, we discuss the simulation parameters and performance measures we use for the simulation experiment.

4 Simulation parameters

Cost link functions and aggregation methods are clearly two important parameters for routing performance. The network performance is measured by network throughput which is defined as the total flow successfully transmitted by the network, and control overhead such as signaling delay which measures the efficiency of routing establishment. These parameters are detailed in the following.

4.1 Link cost metric

As discussed in above, constant link cost metric is currently used in Internet by min-hop routing, while exponential link metric is promising with theoretical guarantee in performance for on-line routing algorithm. The performance of these two metric is studied in this paper. These two metric are detailed as follows.

- **constant**: The cost for each link is a constant regardless of the currently available bandwidth on the link. This corresponds naturally to minimum hop routing in Internet;
- **exponential**: Link cost is an exponential function of the remaining bandwidth on the link. This cost function is justified by both queueing theory suggestion and opportunity cost functions [3].

4.2 Aggregation methods

Aggregation methods play an important role in network performance. Different aggregation methods incur different combination of computation burden and representation size. This would affect the overall overhead for managing network routing information.

Aggregation methods being studied include Complete (full information is provided), Stars (Average and Diameter), Spanning Tree (Minimum spanning tree or MST, and Random spanning trees or RST), and graph spanner. Method Complete is used as a basis for comparing performance loss due to aggregation. Due to the on-line nature of the problem, in some cases, especially those with low link delay, Complete exhibits slightly worse performance than other methods. This is because previously made "optimal" decision could turn out to be bad for future connection requests.

A separate study is conducted on the performance of different aggregation methods [3]. It is shown that Complete and Minimum Spanning Tree provide good performance almost always among these methods, while spanner can achieve good performance though with high aggregation computation expenses.

4.3 Network dynamics

Of the many other factors affecting performances besides the link cost functions and aggregation methods mentioned in the above sections, we are most interested in Network hierarchy structure, link delay, network load, and link capacity.

- Network hierarchy structure: various hierarchies of network structure, even for the same connectivity graph of lowest-level nodes, could lead to much different network performance. Network hierarchical structure is the main focus in this paper;

- Link delay: link delay is normalized to the connection average holding time. Link reservation process becomes asynchronous due to link delay. Some link bandwidth could be held by some connection request which would eventually fail to use it, yet other requests are prevented from using the bandwidth as it is held by this connection request;

4.4 Performance measures

Two 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 per connection.

Many parameters and environment attributes can affect the performance. Aggregation methods as described in 4.2 could affect performance since the aggregated information used by routing protocols could differ much for various aggregation methods. Other parameters might enhance or suppress the differences among the aggregation methods. The link cost metric used by the routing algorithm in calculating shortest path is studied, as the minimum hop metric and the exponential cost function (described in section 4.1) is compared. Other parameters under which the hierarchical routing and global routing were examined are link delay, network load, and link capacities.

An observation is that, when backtracks are used in both link cost metrics, the entire network bandwidth is up to consumption regardless of the aggregation method in use. The result is in many cases, throughput is expected to be very similar for various aggregation methods with differences only due to interaction between concurrent reservations.

To gain insight on how different parameters (especially aggregation methods) affect network performance, we examine topologies which emphasize the differences between parameters under examination. Usually this is carried out on graphs of regular pattern, though the "degree" of regularity differs much among these graphs. The study on these graphs would provide a guide for further investigation on other graphs. In this study, graphs under investigation have the same lowest level connectivity. Different hierarchical structures are imposed to induce different hierarchical levels.

5 Simulation results

We intend to compare the performance of the global routing and hierarchical routing strategy. Networks on the same physical topology but with different imposed hierarchy structures are used for the performance study of the hierarchical routing, where a performance comparison among these hierarchy structures is achieved.

Theory suggests that the exponential cost metric is an optimal policy to ensure that an on-line algorithm is competitive with its optimal off-line counterpart. In section 5.3 we report simulation results that confirm this theoretical result. Our study on other topologies also verifies the superiority of the exponential cost metric for hierarchical routing [3]. As a result we select this metric for our simulations.

We first introduce the network hierarchy structures for our performance studies.

5.1 Topologies under study

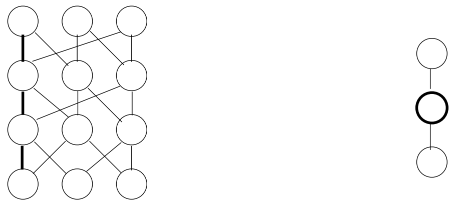
We study topologies with certain patterns. That is, we concern about both connection pattern at the lowest level, and hierarchical structures imposed on such lowest level connectivity. These rather "regular" topologies allow us to study on the effect of some individual network parameters.

In the following sections, a particular topology called *self-similar hierarchical* (SSH) is studied. This topology contains several levels, with each level as a switching network (shuffle-exchange). Figure 1 illustrates this topology in more details.

5.2 Simulation strategies

To compare the performance of network hierarchy structures with the same physical connectivity, simulations are performed on each of them with the same sequence of connection requests. That is, the same sequence of requests is used to drive the simulation on these hierarchy structures. Connection requests are modeled by a Poisson process with exponentially distributed connection holding time, while inter-arrival time between requests are also exponentially distributed.

Parameters of the sequence of requests of our experiments are used to quantify the network dynamics. 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



A. topology of layers 2 & 3 B. topology of layer 1

Figure 1. Self-similar Hierarchical 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 as a shuffle-exchange. Link capacities: 5 for internal links in layers 2 and 3, 10 for the bold internal links in layers 2 and 3, 10 for links between border nodes, 15 for links between border nodes that both have bold links.

the average minimum cut (over all source-destination pairs). The load is a relative measure.

5.3 Effect of the link cost metrics

Consider the performance of the exponential cost metric against that of constant metric.

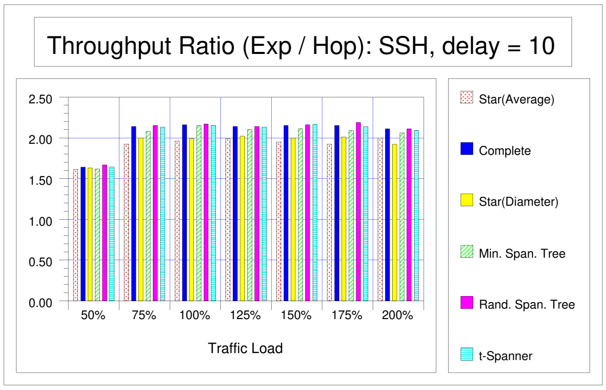


Figure 2. Ratio of throughput between exponential and constant cost metric for SSH topology in figure 1

Figure 2 shows the ratio of the throughput between the exponential and constant link metric, and Figure 3 shows the ratio of the crankbacks between these two metrics. We observe that the exponential metric achieves more than 50% (sometimes more than 100%) gain in network throughput against the constant metric, while suffers less than 20% of crankbacks.

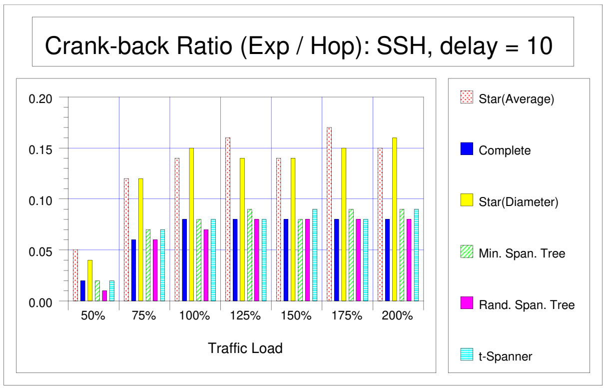


Figure 3. Ratio of crankbacks between exponential and constant cost metric for SSH topology in figure 1

This shows that the exponential cost metric performs much better than the constant cost metric. As a result, we use the exponential link cost metric for the rest of the studies. In the following, we study the effect of imposing hierarchies on the routing performance.

5.4 Effect of the hierarchies

We focus on the hierarchical routing performance for different hierarchies that are imposed on the same topologies. That is, we study how different hierarchies would affect the routing performance for the same physical network.

The SSH topology (in Figure 1) contains 2 levels of imposed nodes. Each level has the same interconnection among nodes. It is natural to study the effect of hierarchy structures using the SSH topology: besides the 2-level hierarchy structure and the 0-level structure (no hierarchy, i.e., the physical network), we induce a 1-level hierarchy structure from the 2-level structure by removing the middle node in the highest level to remove one level. These three structures, namely, 2-level SSH, 1-level SSH and SSH with no hierarchy, are used to study the effect of hierarchy structures on the SSH topology in the following sections. Note that, routing on the SSH topology with no imposed hierarchy is the global routing.

5.4.1 Hierarchical routing vs. global routing

Consider the SSH hierarchy with 2 levels and the SSH hierarchy with no levels (physical connectivity, the routing upon which is the global routing). The performance of these two structures would compare the performance of the hierarchical and the global routing.

Figure 4 shows the ratio of the throughput between the 2-level hierarchy and the 0-level hierarchy, and Figure 5 shows the ratio of the crankbacks between the two hierarchies.

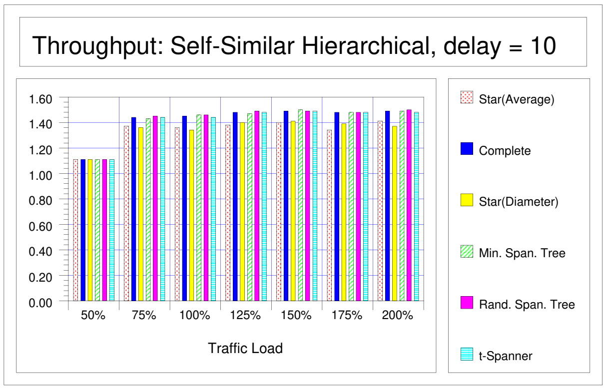


Figure 4. Ratio of throughput between SSH with 2-level hierarchy and SSH with no hierarchy (global routing). SSH topology is in Figure 1

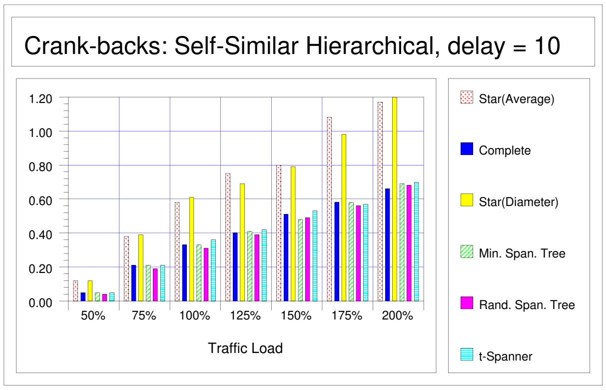


Figure 5. Ratio of crankbacks between SSH with 2-level hierarchy and SSH with no hierarchy (global routing). SSH topology is in figure 1

We can see from the figures that the 2-level hierarchy structure performs much better than the 0-level hierarchy structure in terms of the network throughput. For most of the situations studied, the 2-level hierarchy would produce more than 40% throughput. For the remaining network situations, the 2-level hierarchy still outperforms the 0-level hierarchy by 10% in terms of the network throughput. Overall, the 2-level hierarchy consistently outperforms the 0-level hierarchy on the network throughput.

For the crankbacks, except for very few situations, the 2-level hierarchy performs even better than the 0-level hierarchy. For most of the situations, the 2-level hierarchy incurs less than 60% of the crankbacks to that of the 0-level hierarchy. For the very few situations, the 2-level hierarchy incurs at most 120% of crankbacks to that of the 0-level hierarchy.

Therefore, for certain networks such as the SSH topology, imposing hierarchies on the physical networks would lead to better performance in both the throughput and the signaling delay than deploying routing on the physical networks with no hierarchies. Good hierarchy designs would guide the hierarchical routing mechanism with useful information, and thus could lead to improved performances.

5.4.2 2-level SSH hierarchy vs. 1-level

We now consider imposing different hierarchies on the same SSH topology. We investigate the hierarchies with different levels. Two hierarchies are studied, namely the 2-level SSH hierarchy and the 1-level SSH hierarchy.

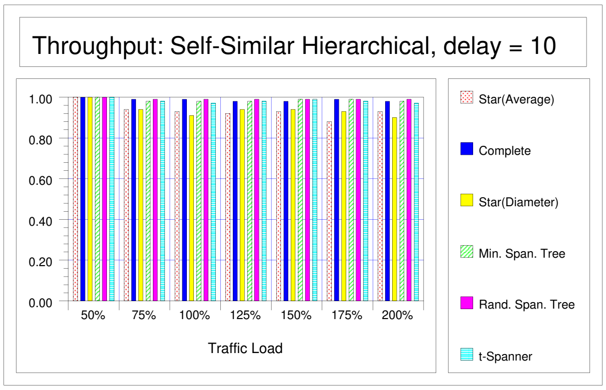


Figure 6. Ratio of throughput between SSH with 2-level hierarchy and SSH with 1-level hierarchy. SSH topology is in figure 1

Figure 6 shows the ratio of the throughput between the 2-level SSH hierarchy and the 1-level SSH hierarchy, and Figure 7 shows the ratio of the crankbacks between these two hierarchies.

As Figure 6 shows, the 2-level SSH hierarchy has almost the same throughput as the 1-level SSH hierarchy. In fact, the throughput for the 2-level SSH hierarchy is slightly less than that of the 1-level hierarchy.

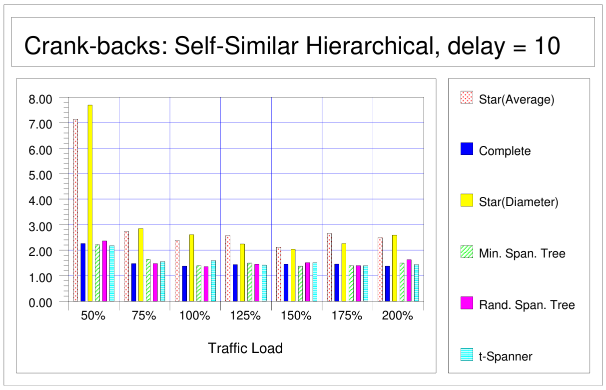


Figure 7. Ratio of crankbacks between SSH with 2-level hierarchy and SSH with 1-level hierarchy. SSH topology is in figure 1

Figure 7 shows that, the 2-level SSH hierarchy performs worse than the 1-level hierarchy in terms of crankbacks. In most of the situations, the 2-level hierarchy would incur 50% more crankbacks than the 1-level hierarchy, and there exist many situations where the 2-level hierarchy would incur 100% more crankbacks.

This shows that, different hierarchies imposed on the same physical network topologies could lead to drastically different routing performances. This justifies the necessity to study hierarchy designs for networks in order to achieve good performances. More understanding is needed on how the hierarchy structures would affect the performances, and how to predict the hierarchical routing performance for certain hierarchy structures.

6 Conclusion

In this paper, we consider the effectiveness of ATM PNNI hierarchical routing. We study the hierarchical routing performance for networks with different imposed hierarchies. In particular, we have compared the performance of the global routing and the hierarchical routing strategy, in terms of the network throughput and signaling delay for realized connections.

Our study shows that in general, the hierarchical routing performs well when compared to the global routing strategy while utilizing less storage and communication overhead with more computation overhead. For certain networks and topologies, the hierarchical routing even performs better than global routing. We also observe that different hierarchies imposed on the same topologies could have drastically different performance implications. This suggests the necessity to study the hierarchy design for networks in order to achieve good routing performances. Link cost metrics have big impact on the routing performances. Our study shows that the exponential link cost metric used in hierarchical routing provides an effective routing solution for large networks.

References

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

[2] 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.

[3] B. Awerbuch, Y. Du, B. Khan, and Y. Shavitt. Routing through networks with hierarchical topology aggregation. *Journal of High-Speed Networks*, 7(1), 1998.

[4] B. Awerbuch, Y. Du, and Y. Shavitt. Stars: A simulator for performance study of aggregation based hierarchical routing. In *SCS/IEEE SPECTS'98*, July 1998.

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

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

[7] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

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

[9] R. Guérin and A. Orda. QoS-based routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350 – 364, 1999.

[10] C. Hedrick. Routing Information Protocol, June 1988. Internet RFC 1058.

[11] G. S. Malkin and M. E. Steenstrup. Distance-vector routing. In M. E. Steenstrup, editor, *Routing in Communications Networks*, pages 83 – 98. Prentice Hall, 1995.

[12] J. Moy. OSPF Version 2, Mar. 1994. Internet RFC 1584.

[13] J. Moy. Link-state routing. In M. E. Steenstrup, editor, *Routing in Communications Networks*, pages 135 – 157. Prentice Hall, 1995.

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