Topology Aggregation for Networks with Hierarchical Structure: A Practical Approach

Yuval Shavitt
Network & Service Management Research Dept.
Bell-Labs, Lucent Technologies
101 Crawfords Corner Rd., room 4G-627, Holmdel, NJ 07733-3030
shavitt@lucent.com

Abstract

This paper addresses the problem of aggregating the topology of a sub-network in a compact way. The problem arises from networks that have a hierarchical structure, where each sub-network must advertise the cost of routing between each pair of its border nodes. The straight-forward solution of advertising the exact cost for each pair has a quadratic cost which is not practical.

We seek an aggregation that minimizes the distortion, defined as the ratio between the advertised cost of traversing the network between two border node and the actual cost. We compare the average and maximum distortion of several heuristics, based on spanning trees, with a theoretically bounded aggregation scheme. We show that in practice, a combination of a minimum spanning tree with random spanning trees achieves better aggregation than a theoretically bounded aggregation scheme.

1 Introduction

Future global networks will consist of hierarchies of subnetworks, or domains. 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 [1, 6]. 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 [7]. 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 [1], 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) the connection 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*. 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.

Methods for compact, i.e., linear in the number of border nodes, graph representation that can be used to aggregate topology information were studied in the past years [9, 8, 2, 5]. These methods have a a logarithmic bound on the worst-case distortion of costs between any pair of nodes. In other words, for any pair of nodes the ratio between the advertised and the true cost of the shortest path that connect them is bounded by a $\log b$, where b is the number of border nodes. Awerbuch and Shavitt [4] presented an aggregation scheme based on Bartal trees [5] and showed that it achieves the promised logarithmic bound. Awerbuch et al. [3] conducted a simulation study of PNNI routing where they found that using a Minimum Spanning Tree (MST) aggregation does not degrade performance. Lee [7] suggested to use a maximum delay spanning tree as an upper bound estimate for the delay, which is an additive metric. The bounds calculated from his tree are not tight when the difference between the maximum and the minimum traversing delay is high, a situation occurring, e.g., when the original topology is sparse.

In this work, we investigate the aggregation performance of some heuristics based on MST and Random Spanning Tree (RST), and compare them with the t-tree aggregation suggested in [4], and with the maximum spanning tree suggested by Lee [7]. We show that MST is a good aggregation scheme, which corresponds to the simulations conducted in [3], and show that it can be enhanced by combining the MST with RSTs. We also show that a combination of three RSTs is very bad, which again corresponds to the results obtained in [3].

The rest of the paper is organized as follows. In the next section, we describe the PNNI standard that is used as our network model. In section 3, we describe previous relevant work. The suggested aggregation heuristics and their performance are described in the next section. Finally, we conclude by suggesting how to use our results in practice.

2 The PNNI Model

The PNNI standard is part of the ATM protocol suite that defines the network-to-network interface. It defines, among all, how the network is organized in a hierarchical manner into peer groups, how aggregated topology information is exchanged among peer groups, and how hierarchical source routing is performed. We selected this standard for our network model as it is a mature standard with a growing deployment base. However, our results hold in general for any topology aggregation, as our reliance on PNNI is very limited (see below).

In the PNNI standard [1, section 3.3.8], a network is aggregated ("complex node") by a star. A virtual node ("nucleus") is the interior reference point. For each state parameter, a single default value is given to represent the value of this parameter for the connection between a border node and the nucleus ("spoke") in each direction.

To cope with asymmetry, the standard allows "exceptions" of two types: a different value for a spoke of a certain border node, or a value for a "bypass" between two border nodes. In guidelines that are supplied with the standard [1, Appendix C] it is recommended that the number of exceptions used to configure a complex node will be kept smaller than $3b$, where b is the number of border nodes. There is no suggestion on how to select exceptions, or how to compute the default spoke value.

In our heuristics, we use only the default spoke value and up to $3b$ bypass exceptions. The default spoke value is the only part of our heuristic that might be considered, as specific to PNNI. A bypass exceptions is, in our view, a general way to represent any graph and we believe will be part of any future aggregation standard.

The performance measure used in this study is the *distortion*. The distortion between two border nodes is defined as the ratio between the cost of a parameter as calculated from the advertised topology and this cost in the real topology. As opposed to the usual practice in theoretical work [9, 8], we do not solely compare the worst case distortion, but examine also the average and the variance. It has been shown in the past [3] that worst case analysis cannot predict the efficiency of aggregation algorithms for routing in hierarchical networks.

3 Previous Work

Lee [7] suggests to use a maximum or minimum spanning tree to accurately represent path metrics that are calculated by taking the maximum or the minimum of the path's link values. E.g., he suggests to use maximum spanning tree to represent the maximum available bandwidth between any two nodes. For additive parameters, such as delay, he suggests to use a maximum spanning tree. The delay between any two points, u and v, that are not directly connected by the tree is estimated by the minimum delay of the links in the cycle generated when the link $(u,\ v)$ is added to the tree. If the network is highly symmetric his methods does not improve significantly the estimate given by the default spoke value. When the network is highly asymmetric the estimates are very bad. In particular, differences among the best routes (i.e., the shortest) tend to vanish, making the estimation not very useful. Our heuristic, attempts at reliably estimating the links with the low cost, while compromising accuracy for links with higher cost.

The calculation of the estimates using Lee's tree is problematic since it requires the receiving node to use a minimum function on the path between two nodes and not the standard addition. The use of the standard addition results in extremly bad estimates.

Thus Lee's suggestion cannot be incorporated into PNNI without some provisioning in the standard.

Bartal [5] suggests a tree representation for graphs with an average distortion of $O(\log n)$, where n is the number of nodes in the graph. The tree is not built from the edges and the nodes of the original graph. Instead, as many as $O(n)$ virtual nodes are required for this construction. If a Bartal tree is to be used to advertise a "complex node" the added virtual nodes should be advertised together with the link cost, which increases the representation cost of the "complex node".

Awerbuch and Shavitt [4] present a framework for aggregating a "complex node" topology based on Bartal trees. Their aggregation has a probabilistic bound on the worst case distortion in the advertised parameters which is $\log b$. They embed a variant of a Bartal tree in the network to keep the representation size of a single tree to be not higher than $b-1$. They suggest two structures based on the above embedded trees. A *t-tree*, which is the union of t random Bartal trees, and a slightly more complex construction called quality partition forest (QPF). They tested the t-tree construction and showed that the logarithmic bound is achieved in practice [4].

4 Heuristic Aggregation Algorithms

The construction of a Bartal tree (and thus of the t-tree aggregation) is complex and hard to code. This work suggests some simpler aggregation schemes that are based on spanning trees. Theoretically, an aggregation based on a tree have a maximal distortion which is linear in the graph size. For example, if a homogeneous ring (with the cost of all the links equal to 1) of size n is to be aggregated using a single tree the cost of the link which is removed is increased by a factor of $n-1$. However, the experiments we conducted and report here, show that, in practice, tree based aggregations have good distortion, better than the bounded aggregation of [4]. The good performance together with the ease of their implementation make them an attractive candidate as a heuristic method for practitioners.

We tested the following methods:

MST — the minimum spanning tree, comprised of $b-1$ edges.

3RST — three random spanning trees. This structure can have up to $3b-3$ edges. In practice, since the random trees overlap we used about $2.5b$ edges.

MST + 2RST — a MST plus two random spanning trees. The maximal number of edges here is $3b-3$, the actual number of edges was similar to the 3RST case.

t-tree — several Bartal trees as described in [4]. In most cases, the number of edges used was just over $2b$.

Lee — a maximum spanning tree [7], where the distance evaluation is done by taking the minimum of the links connecting two nodes (see Section 3).

To compare between the aggregation schemes, we produce random graphs according to Waxman's method [10]. In the graph creation process, links are added until all nodes reach a minimum node degree of two. Links that increase node degree over 6 are rejected. This way the resulted graphs have both the clustering effect of close nodes and a relative high diameter. We used the graphs that contain 100 nodes. The number of border nodes

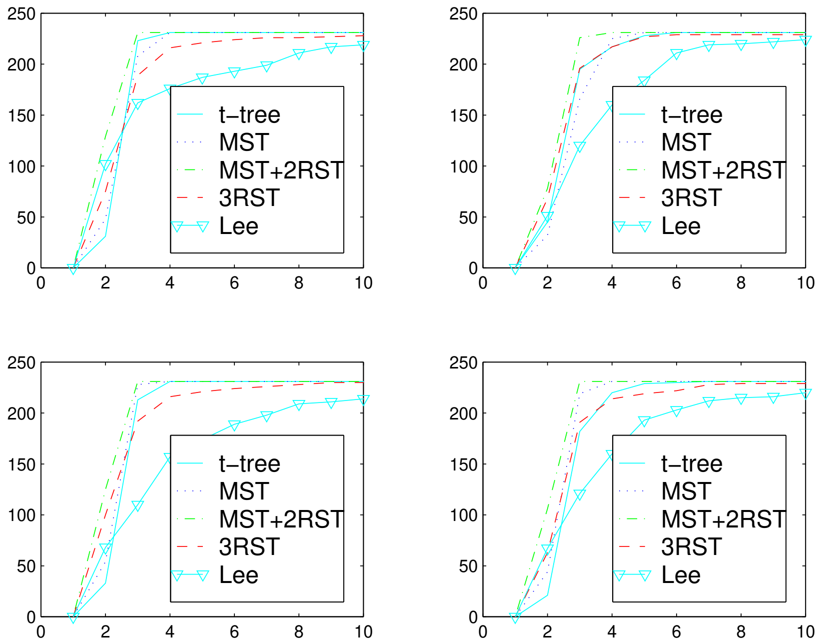


Figure 1: A comparison between the histogram of the aggregation schemes for the 22 border node graph.

was set to 22 and 33. For each graph we conducted several experiments, in which links were assigned different random values.

Figures 1 and 2 depict the cumulative histograms for the different aggregation schemes for four sets of experiments for each graph. The graphs show that MST has similar quality to the t-tree while using less than half the number of exceptions. MST+2RST has much tighter aggregation representation (with slightly more exceptions). In fact, in all but one experiment, this aggregation scheme had only a negligible number of node-pairs that were distorted by more than a factor of 2. 3RST performs similar to the t-tree, in most cases, but since its aggregation distortion is higher (and sometime by a large margin) than MST+2RST and since the calculation of a MST is not harder than the calculation of a RST, there is no reason to prefer this scheme. The histogram for the aggregation suggested by Lee [7] is, always, inferior to MST+2RST, and, in most cases, even inferior to MST.

Figures 3, 4, and 5 compare the maximum distortion, average distortion, and the variance of the distortion among the aggregation schemes. The average distortion of all the aggregation schemes is in the same range, but MST+2RST has the lowest average. The differences between MST+2RST and the rest of the aggregation schemes are between 20% and 100%, and up to 400% difference to Lee's method. However the maximum distortion graphs show that 3RST has a 1–2 orders of magnitude higher maximum distortion than the rest of the schemes. Lee's method produced even higher maximum distortion and thus was not plotted. In some of the cases, t-tree has a maximum distortion that is several times higher than that of MST. This result in a higher variance in the distortion for 3RST and t-tree. The addition of two RSTs to an MST practically reduces the variance to negligible values.

It is interesting to compare our results with the ones obtained by Awerbuch et al. [3]. In their simulation study, they compared the overall performance of the PNNI routing algorithm when several aggregation schemes are employed. The aggregation schemes studied in [3] include MST and RST, which are compared to the case where no

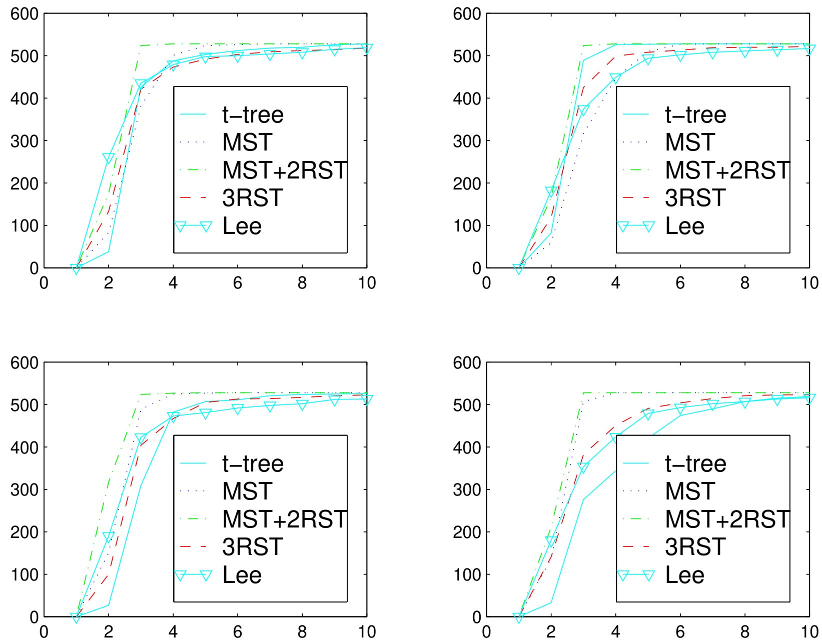


Figure 2: A comparison between the histogram of the aggregation schemes for the 33 border node graph.

aggregation is used. MST was found to perform very well with little to no difference from the full knowledge case. This corresponds to our results that show that MST has low distortion.

Awerbuch et al. [3] found that the RST aggregation performance is sensitive to the topology, sometimes it is much worse than MST and sometime almost identical. This might be explained by the large variance in the distortion found in this study.

MST+2RST is shown here to be superior to MST. However, in [3], MST is shown to perform very well. It might be due to the simulation of topologies with a small number of border nodes.

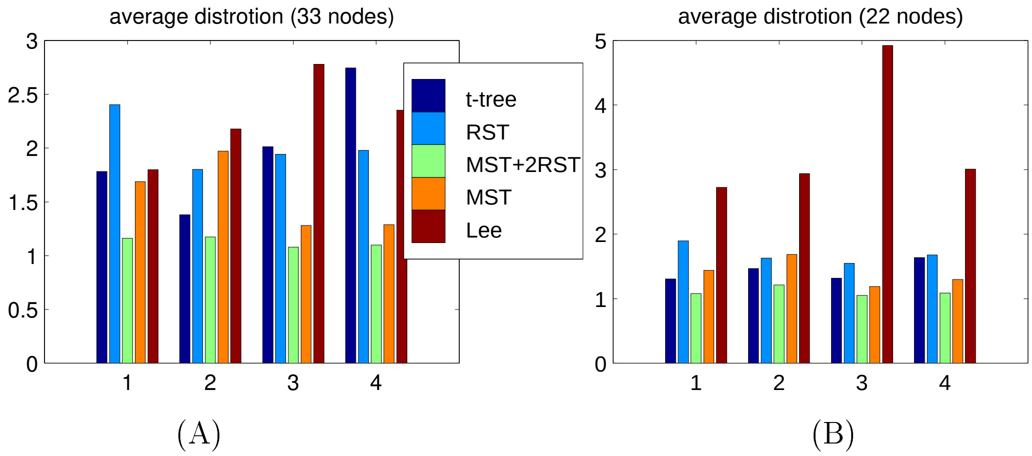
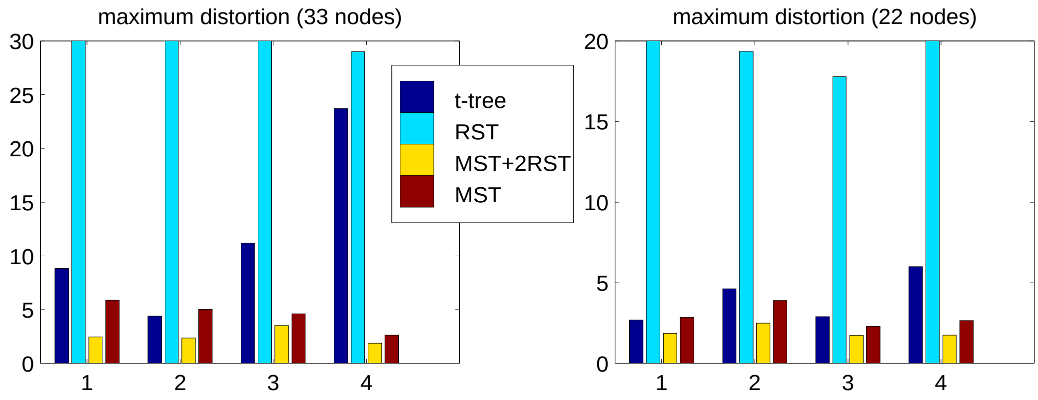
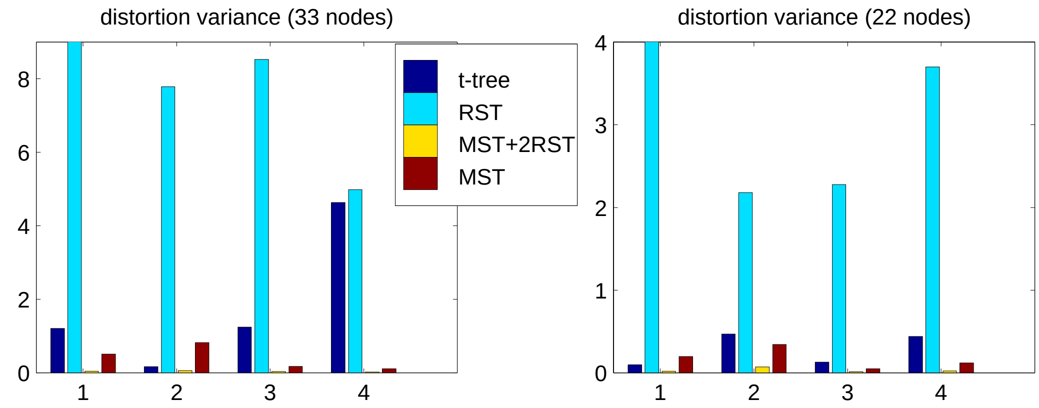


Figure 3: A comparison between the average distortion of the aggregation schemes. (A) For the 33 border node graph. (B) For the 22 border node graph.



(A) (B)

Figure 4: A comparison between the maximum distortion of the aggregation schemes. (A) For the 33 border node graph. For data sets 1, 2, and 3 the maximum distortion for 3RST is 249, 54 and 50, correspondingly. (B) For the 22 border node graph. For data sets 1 and 4 the maximum distortion for 3RST is 58.5 and 24.2, correspondingly.



(A) (B)

Figure 5: A comparison between the variance of the distortion of the aggregation schemes. (A) For the 33 border node graph. For data set 1 the variance of the distortion for 3RST is 122. (B) For the 22 border node graph. For data set 1 the variance of the distortion for 3RST is 16.2.

5 Concluding Remarks

In the past a theoretically bounded aggregation scheme that is suited for ATM PNNI standard was suggested. We presented here a heuristic, based on a combination of minimum and random spanning trees that is shown to perform much better in practice, and is much simpler to implement.

Since once an aggregation is calculated, it is easy to check its quality, one can always use the low cost MST aggregation, augment it with RSTs if needed, and revert to t-tree in those cases where the aggregation distortion is not acceptable.

References

[1] Private network-network interface specification version 1.0 (PNNI). Technical report, The ATM Forum technical committee, March 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] Baruch Awerbuch, Yi Du, Bilal Khan, and Yuval Shavitt. Routing through networks with hierarchical topology aggregation. *Journal of High-Speed Networks*, 7(1), 1998.

[4] Baruch Awerbuch and Yuval Shavitt. Topology aggregation for directed graphs. In *Third IEEE Symposium on Computers and Communications*, pages 47 – 52, June 1998.

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

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

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

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

[9] 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.

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