On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design

Pavlík Radoslavov
Hongsuda Tangmunarunkit
Haobo Yu
Ramesh Govindan
Scott Shenker
Deborah Estrin

Abstract

Recently there have been several papers examining aspects of the Internet topology. This paper follows in that tradition and addresses two issues related to Internet topology. First, we use three properties—expansion, resilience, and distortion—to characterize real and generated networks. For these metrics, we find that existing network topology generators differ qualitatively from most real networks. Second, we ask what impact topology has on four different multicast design questions. We find that, for many of these questions, a single topology metric appears to influence the answer.

1 Introduction

In the last year or so, interest has turned towards characterizing various aspects of Internet topology. This has been sparked by attempts to infer the router-level topology of the Internet [14, 2, 4, 25] and of the multicast overlay, the Mbone. Maps of autonomous-system (AS) level connectivity have also been available for some time now [11]. The availability of these topologies has motivated recent work on understanding their structure. One such effort revealed that several graph-theoretic properties of some of these real topologies could be described using power-laws [10]. Another argued that, for some real topologies, the size of neighborhoods grows exponentially with distance. This has interesting implications for the size of multicast trees [20].

Part of this paper continues in this vein, focusing on topology characterization. We are motivated by two questions. What are other interesting topology characterizations? Where do topology generators diverge from real networks? Exploring alternative characterizations can give us some intuition about what, if any, regular topologies best model real networks along these different dimensions. Topology generators attempt to emulate, at a macroscopic level, how networks are built. Comparing the properties of generated networks with those of real networks can reveal the limitations, if any, of this emulation. By asking these questions, we hope to understand something about the phenomena that determine network structure.
Another focus of this paper is the impact of network structure on protocol design. We ask whether topology has any impact at all on protocol performance, and whether different characteristics of topologies influence designs in different ways. The practical significance of topology characterization hinges on the answers to these questions.

Our methodology for investigating alternative topology characterizations (Section 2) is simple. First, we select three topology metrics: expansion, which captures neighborhood sizes in topologies; resilience, which is analogous to cut set sizes in graphs; and distortion, which indicates average increase in path lengths as a result of link failure. This selection is not based on any fundamental understanding of network structure, but is merely initial guess as to what factors might be most relevant to network and protocol design. Then, we describe some canonical topologies (the tree, the mesh, the reduced mesh, and the random graph) and list their metric values. We choose these topologies because they represent qualitatively different combinations of topology metrics. Finally, we discuss where real and generated networks fall in this space of topology metrics. Our real topologies include the Internet, the inter-AS connectivity topology and the Mbone. We also include most of the commonly used topology generators such as Tiers, Transit-stub, and Waxman.

Results from this investigation reveal that, with one exception, real topologies exhibit consistent behavior with respect to the three metrics we study: i.e., they all display similar characteristics as measured by these metrics. However, again with one exception, topology generators generally fail to capture the qualitative behavior of real networks for these metrics.

To study the impact of network structure on protocol design, we select some multicast protocol design questions. We choose to study multicast because we suspect that it is fairly sensitive to topology (more so than, say, TCP). Our four case studies are: (1) the comparison of source-receiver distances in shared and shortest path trees (Section 4), (2) the aggregatability of multicast forwarding state (Section 5), (3) the performance of end-system multicast designs (Section 6), and (4) the feasibility of alternate path routing (Section 7). This list is not intended to be a complete enumeration of topology sensitive multicast design issues. Rather, our choice was colored both by the current research interest in these areas, and by our attempt to find some diversity in our case studies.

Our case studies indicate that, in some cases, there is one topology metric that most influences protocol performance. Furthermore, we find that this metric can be different for different case studies. In some other cases, we were unable to point to any particular topology metric as influencing certain aspects of protocol performance. This indicates that there may exist alternative topology metrics that determine performance.

Before we proceed to characterize topologies, we emphasize several limitations of our overall methodology. First, all the real topologies that we use in this paper are incomplete. They may not capture all the nodes in the network and, for the nodes that do appear in the topology, they may not include all adjacencies at each node. Furthermore, these inaccuracies are difficult to quantify. For this reason, there is a possibility that significantly more accurate maps will reveal qualitatively different results than ours. Second, in our protocol evaluations, path selection is determined by the shortest path in the topology graph. This may not reflect reality, particularly for the Internet topology, where path selection is constrained by routing policy. Finally, in our case studies, we assume uniform receiver placement because we lack
representative data and models\(^1\) for receiver clustering. Clustering may alter some of our results, and should be the subject of future research.

2 Topology Characterization

One aim of this paper is to attempt to understand the characteristics of real network topologies beyond that described in recent work [10, 20]. To this end, we first describe the various topologies we use in the paper, and the metrics that we choose to compare these topologies.

2.1 Networks

A sub-goal of the paper is to understand how well topology generators model real networks. This section describes the specific generated and real topologies we use in this paper. Figure 1 lists the size and average degree of these topologies.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Number of Nodes</th>
<th>Average Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transit-stub</td>
<td>1008</td>
<td>2.77</td>
</tr>
<tr>
<td>Tiers</td>
<td>5000</td>
<td>2.83</td>
</tr>
<tr>
<td>Waxman</td>
<td>5000</td>
<td>7.22</td>
</tr>
<tr>
<td>Mbone</td>
<td>4179</td>
<td>2.05</td>
</tr>
<tr>
<td>AS</td>
<td>4830</td>
<td>3.76</td>
</tr>
<tr>
<td>Internet core</td>
<td>54533</td>
<td>5.37</td>
</tr>
</tbody>
</table>

Table 1: Topology sizes and average degrees.

Our work uses the following generated topologies (the label preceding each item in the list is also the label under which the topology appears in our graphs):

**Transit-stub** This topology was generated using the Georgia Tech Internet Topology Modeler (GT-ITM) [3]. This topology generator has many parameters. Its basic generation procedure creates a number of top-level transit domains within which edges are connected randomly. Attached to each transit domain are several stub domains, generated analogously. Additional stub-to-transit links are added randomly based upon a specified parameter.

**Tiers** This topology is obtained from a different implementation of the transit-stub generation methodology described above [8]. First, it creates a number of top-level networks to each which are attached several intermediate tier networks. In the same way, several LANs are randomly attached to each intermediate level network. Within each tier, (except LAN), Tiers uses a minimum spanning tree to connect all the nodes, then adds additional links in order of increasing inter-node Euclidean distance. LAN nodes are

\(^1\)The methodology proposed in [20] for studying clustered receiver placements doesn't scale well to large numbers of receivers.
connected using a star topology. Additional inter-tier links are added randomly based upon a specified parameter.

**Waxman** This topology, also obtained from GT-ITM, is generated by placing nodes on a plane, and randomly creating an edge between every pair of nodes with a probability that depends on the Euclidean distance between the two nodes [27].

The generated topologies for which we show results in this paper were generated with a specific set of parameters\(^2\). We have verified that using a different set of parameters does not qualitatively change the characteristics of these generated topologies. As such, the topology characteristics we present in Section 2.4 reflect, we believe, the generation techniques rather than the specific parameters we used to generate these topologies.

This paper uses the following real topologies:

**Mbone** This router-level topology of the multicast overlay network is obtained by recursively querying each multicast router for its neighboring routers [21].

**AS** This topology, representing inter-autonomous system (AS) connectivity, is obtained from the AS path information in routes carried in backbone routing tables [11]. We emphasize here that the AS map is not a router-level map. It is therefore not entirely meaningful to include the AS map in some of our protocol analyses, such as the analysis of multicast forwarding state aggregation (Section 5). Regardless, we do so if only because it is one of the few realistic topologies available to us.

**Internet core** This topology is derived by inferring router adjacencies [14] using traceroutes to carefully chosen segments of the IP address space. However, this technique does not give much detail outside the transit portion of the Internet. These non-transit parts of the Internet appear as chains in the resulting topology. In this paper, we eliminated these chains, extracting a possibly high fidelity map of the Internet core.

### 2.2 Metrics

Having described our real and generated topologies, we now explain what topology properties we intend to study. Before doing so, we introduce some notation. We use \( h \) to denote the number of hops between nodes in a network topology. \( n \) signifies the number of nodes in a subgraph of a topology—typically, this subgraph is obtained by taking a ball of radius \( h \) around some network node. Finally, \( N \) denotes the number of nodes in the complete topology.

Recent research has studied some simple, graph-theoretic properties of real networks. Such properties have included the degree distribution [10] and the neighborhood size [10, 20]. Continuing in the same vein, we attempt to characterize network topologies in terms of three metrics: expansion, resilience, and distortion.

**Expansion**, denoted by \( E(h) \), attempts to capture the growth of neighborhood sizes in a network as a function of distance. \( E(h) \) is the average of the number of nodes within

\(^2\)The sizes of our generated topologies are chosen so that our results are comparable to those in [20].
distance \( h \) from a node in the topology. This definition is identical to the reachability function described in [20] and is similar to the hop-pair distribution defined in [10]. In fact, [20] has analyzed the expansion of many, but not all, of the topologies described in Section 2.1. We repeat those analyses here for completeness.

The resilience metric attempts to capture the robustness of communication between any two nodes in the network. For resilience, we use a classical graph-theoretic definition: the resilience of a topology is the minimum cut-set size for a balanced bi-partition of the topology. This is the smallest number of links that need to be cut in a network of size \( N \) so that the two resulting components each have approximately \( N/2 \) nodes. This definition gives us, for each topology, a single number. Such a number is uninteresting because it defines a macroscopic property of the entire topology. Missing are details about the resilience at different scales within the topology. This behavior is essential for some of our protocol analyses; if a topology exhibits radically different behavior at different scales, knowing this is essential for deciding which aspect of the topology’s behavior impacts the protocol. For this reason, our resilience metric \( R(n) \) is defined as the average of the minimum cut-set size for balanced bi-partition of subgraphs of \( n \) nodes within a “ball” around different nodes in the topology.\(^3\) This definition also allows us to compare the qualitative behavior of the resilience function across topologies of different sizes.

Computing the minimal cut-set size for a balanced bi-partition of a graph is NP-hard [18]. We use the well-tested heuristics described in [18] to compute \( R(n) \).

Our final metric, distortion, requires a more careful explanation. Consider any spanning tree \( T \) on a graph \( G \), and compute the average distance on \( T \) between any two vertices that share an edge in \( G \). Intuitively, this number measures how \( T \) distorts edges in \( G \), i.e., it measures how many extra hops are required to go from one side of an edge in \( G \) to the other, if we are restricted to using \( T \). We define the distortion\(^4\) of \( G \) to be the smallest such average over all possible \( T \)s. Intuitively, the distortion is a structural property of a network that captures how lengths of paths are increased with increasing link failures. In this respect, it appears to be related to the resilience metric. We show in Section 2.4 that there exist topologies which can distinguish between the two metrics.

For a given graph, distortion is a single number. As for resilience, we define the distortion metric for a topology \( D(n) \) to be the average distortion of a subgraph of \( n \) nodes within a “ball” around a node in the topology.\(^5\) Computing the distortion can be NP-hard [24]. For the results described in this paper, we use the smallest distortion obtained by applying several of our own heuristics to sample spanning trees. All of these heuristics use the total number of all-pairs shortest paths traversing a link to inform the selection of links forming the spanning tree.

Unlike expansion, resilience and distortion are defined as functions of \( n \), not \( h \). As we show below, different networks have different expansion characteristics. That is, within a

\(^3\)For each node in the network, we grow balls with increasing radius. For the subgraph formed by nodes within a ball, we compute the number of nodes \( n \) as well as the resilience of the subgraph. We repeat this computation for all other nodes, then average the sizes and resilience values of all subgraphs of the same radius.

\(^4\)This definition is a special case of minimum communication cost spanning trees defined in [15].

\(^5\)The steps in this computation are similar to that of resilience.
given ball of radius $h$, different networks may have vastly different $n$. To avoid the resilience and distortion metrics from being skewed by expansion, they are defined as functions of ball content ($n$), not ball radius ($h$).

### 2.3 Canonical Networks

<table>
<thead>
<tr>
<th>Canonical Topology</th>
<th>Expansion</th>
<th>Resilience</th>
<th>Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>random graph with average degree $k$</td>
<td>$O(k^h)$</td>
<td>$O(kn)$</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>$k$-ary tree</td>
<td>$O(k^n)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>mesh</td>
<td>$O(h^2)$</td>
<td>$O(n^{0.5})$</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>reduced mesh</td>
<td>$O(h)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>

Table 2: Topology parameter space.

In our analyses of various topologies and their impact on protocol design, we also include a set of canonical topologies.

The canonical networks we choose are the mesh (or grid), the reduced mesh (Figure 1(a)), the $k$-ary tree and the random graph\(^6\). As Table 2 shows, these topologies represent qualitatively different points in our metric space. A tree exhibits exponential expansion, but low distortion and resilience. A random graph has tree-like expansion, but high distortion and resilience. The mesh differs from the random graph mainly in its expansion. The reduced mesh has mesh-like expansion but tree-like distortion and resilience. An interesting property of these canonical topologies is that they are “self-similar”: subgraphs of these topologies have the same qualitative characteristics as the original.

Our list of canonical topologies has an omission. It does not contain a topology that usefully distinguishes between resilience and distortion. These two metrics are different. Section 2.4 shows, for example, that some generated topologies reveal this difference. Furthermore, we can concoct topologies whose macroscopic resilience and distortion are quite different. One example is the dumbbell-mesh shown in Figure 1(b), which has tree-like resilience, but mesh-like distortion. However, we have not been able to find a “self-similar” topology which can distinguish between distortion and resilience.

### 2.4 Topology Characterization Results

We computed the three functions $E(h)$, $R(n)$ and $D(n)$ for our real, generated, and canonical topologies. Figures 2, 3 and 4 plot these functions for various topologies. We now discuss the implications of these graphs.

\(^6\)The specific topologies we use in these paper include a 100x100 mesh, a 100x100 reduced mesh, a complete tertiary tree of depth 8 and a 5000 node random graph with an average degree of 9.97.
Expansion

Canonical topologies fall into two categories with respect to expansion (Figure 2(b)): the tree and the random graph have high (exponential) expansion, the mesh and the reduced mesh have low (algebraic) expansion. This is simply pictorial confirmation of Table 2.

Earlier work [10, 20] has studied the expansion of some of the real networks we discuss in this paper. Our finding, that the Mbone exhibits non-exponential expansion (Figure 2(a)), confirms that of [20]. However, these earlier studies disagree about whether the AS and the Internet show algebraic or exponential expansion. We find that the AS and Internet exhibit exponential expansion. However, as we discuss in Appendix A, the expansion of the Internet falls off rapidly (i.e., for relatively small $h$) from the exponential because of edge effects.

The generated networks straddle the two classes of expansion (Figure 2(c)). Waxman shows a random-graph like exponential expansion. The Waxman generator biases the random graph edge generation probability by inter-node Euclidean distance. This does not appear to alter the expansion characteristics of this generator vis-a-vis random graphs. As [20] showed, Tiers has non-exponential expansion and Transit-stub has exponential. The former constructs a Euclidean minimum spanning tree, which is conjectured to have expansion proportional to $h^{1.5}$. 
Resilience

Canonical topologies fall into three classes with respect to resilience (Figure 3(b)). Random graphs have high (linear) resilience. The tree and the reduced mesh have low (constant) resilience. In between these two extremes is the mesh whose resilience varies as the square root of topology size.

Real networks all have mesh-like resilience (Figure 3(a)). For large $n$, the Mbone's resilience has a slope comparable to that of the mesh. The resilience of the Internet and AS maps also have a mesh-like slope, but, for a given value of $n$, have higher absolute resilience. It is striking that none of the real topologies have resiliences approaching those of random graphs.

Finally, topology generators range from high to low resilience (Figure 3(c)). Only the Tiers network models the mesh-like resilience of real networks. It adds redundant links in increasing order of Euclidean distance. This is, however, precisely the reason why it fails to capture the exponential expansion of the Internet and the AS topologies. The resilience of the Waxman topology is similar to that of the random graph. Surprisingly, the Transit-stub topology shows very low resilience. Concerned that this might be due to our particular choice of parameters, we generated differently parameterized Transit-stub topologies using GT-ITM. The results were the same, leading us to believe that low resilience is a characteristic of the topology generator.

Distortion

The distortion of canonical networks falls into two classes. The tree and the reduced mesh have a distortion of 1. The mesh and the random graph appear to have a power-law distortion as a function of ball size. We believe these topologies have logarithmic distortion (Figure 2). We explain this discrepancy as a shortcoming of our heuristic: we have not yet found a satisfactory heuristic for distortion that approximates the theoretical bounds.

Real networks do not fall into either of these categories. Rather, their distortion is
intermediate between that of the random graph and that of the tree. All of them show a
distinct dependence on \( n \), but the growth in distortion does not match that of any canonical
topologies.

Two generated topologies, Tiers and Transit-stub, match the distortion of real networks.
However, the Waxman topology’s distortion matches that of the random graph. Thus,
biasing the edge-selection by Euclidean distance does not appear to qualitatively alter the
random graph-like nature of the Waxman topology, at least with respect to our three metrics.

**Summary of Results**

<table>
<thead>
<tr>
<th>Real or Generated Topology</th>
<th>Expansion</th>
<th>Resilience</th>
<th>Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>AS</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Mbone</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Tiers</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Transit-stub</td>
<td>high</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Waxman</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>

Table 3: Classification of real and generated topologies by metric.

Table 3 summarizes our qualitative categorization of real and generated topologies. We
make four observations. First, the transit-stub topology has tree-like resilience, but a dis-
tortion that is different from a tree. This is another indication, beyond the existence of
the concocted topologies described in Section 2.3, that the two metrics capture different
topological properties.

Second, with the exception of the Mbone topology, real topologies seem to consistently fall
into the same categories for all metrics. The AS and the Internet are obtained using widely
different techniques, and represent different connectivity at different levels of structural abstraction. This suggests that there may be no methodological bias in our results. Instead, there may lurk a deeper, structural explanation for why these topologies are qualitatively consistent.

Third, no generated network matches the characteristics of real networks—the one exception is Tiers, which models the Mbone well. This we find intriguing, because at least the Tiers and Transit-stub plausibly try to emulate, at a macroscopic level, how networks are built. However, we do not understand what aspect of the emulation fails to capture real networks. As an aside, no canonical topology consistently matches the properties of real topologies.

Finally, it is interesting that the Internet has tree or random graph like expansion, mesh-like resilience, and a distortion that is intermediate between the tree and the random graph.

3 Impact of Topology on Protocols

Our topology characterization, and the results from the previous section, are irrelevant if topology has no impact on protocol design or performance. In the rest of the paper, we discuss four case studies in the design and evaluation of multicast protocols. We select multicast protocols because we suspect that they, more than unicast, are likely to be impacted by topology.

Our high-level motivation, then, for these case studies is to test for the existence of performance questions which are influenced by topology. In addition, our case studies have several other objectives.

- We select case studies that explore diverse aspects of multicast protocols: path length, state, traffic concentration, and path availability. Our aim is to understand how to study the impact of topology for very different kinds of protocol performance issues.

- In each case study, we try to check if any of the topology metrics discussed in Section 2.2 affect protocol performance\(^7\). This will help us understand if our initial attempts at topology characterization are headed in the right direction.

- Finally, we include our canonical topologies in order to test the impact of extreme topologies on performance and gain some intuition.

4 Case Study: Shared Tree versus Shortest Path Tree

Our first case study turns towards the topology sensitivity of multicast distribution tree characteristics. IP multicast routing protocols use one of two classes of distribution trees: a source specific shortest path tree (SPT) \([6]\) and a group shared tree \([7, 1]\). A shortest path tree minimizes each receiver’s delay (or distance) to the source. A shared tree, on the other

\(^7\)We say a topology metric affects or impacts a protocol, if the performance of the protocol across topologies is correlated with the relative values of the metric across topologies.
hand, trades off increased delivery latency for a reduction in network-wide routing state (one entry per group, instead of one for every source-group pair). A given source-receiver distance on a shared tree is equal to or larger than the corresponding distance on the shortest-path tree.

There are many types of shared trees that have been proposed in the literature including core-based trees [1] and greedy trees [29]. Increasingly, many recent multicast routing protocol designs [1, 9, 19] have chosen bidirectional shared trees as the basis for their design. In these protocols, packets from a source are directed towards the root of the tree until they reach the nearest on-tree router. That router then distributes copies of the packet towards downstream receivers. It also directs traffic from the source upstream, towards the root of the shared tree.

Simple intuition would suggest that the extent to which paths on a bidirectional shared tree are longer than on a shortest path tree might be topology sensitive. This forms the subject of our investigation.

Our work in this section is inspired by the work of Wei et al. [28]. Their work compares the quality and efficiency of different types of multicast distribution trees. They evaluate the performance of these trees in terms of path length, link cost and traffic concentration on the early ARPAnet network and different classes of random networks based on the Waxman model [27]. Our work differs from theirs in two respects. First, for reasons outlined above, we evaluate bidirectional shared trees whereas [28] focuses on unidirectional shared trees. Second, we examine the topology sensitivity of the path length comparison, evaluating the two trees for a wider variety of topologies.

To explicitly compare the two tree designs (shortest-path and shared tree), we define the distance ratio of an individual receiver to be the ratio of the distance from the receiver to the source on a shared tree to that distance on the shortest path tree. We then choose two performance metrics: the average and the 90th percentile distance ratios across all receivers in the tree.

Because of the way multicast trees are constructed, a receiver’s path to the source or to the shared tree root is independent of the existence of other receivers. That is, for a given topology, these metrics are insensitive to occupancy—the fraction of nodes containing attached receivers. Accordingly, to evaluate our performance metrics, we simply generate a large number of uniform receiver configurations. For each receiver configuration, we uniformly place one source and one shared tree root. We then compute the average and 90th percentile distance ratios. Our final average and 90th percentile distance ratios are averaged over all receiver configurations.

Discussion of Results

Figure 5 shows the average and 90th percentile distance ratio for various topologies. In this bar chart, the distance ratios for the tree and the reduced mesh are not included. For these two topologies, all receivers have a distance ratio of 1—there is exactly one path between a source and a receiver, and both the shortest path tree and the bidirectional shared tree use that path. By comparison, the mesh and the random graph have significantly higher average and 90th percentile distance ratios. As a sanity check, we have verified that for
these canonical topologies the metrics are relatively insensitive to network size. The metrics increase negligibly even with an order of magnitude increase in network size.

Our real topologies have distance ratios intermediate between the tree and the mesh (or the random graph). They all have comparable performance, and their distance ratios are quite small, i.e., less than 1.5 on average and less than 2 for the 90th percentile distance ratio.

The distance ratios of the Tiers and Transit-stub generators also fall in the medium category. The ratio of the Tiers network is comparable to that of the real networks, while Transit-stub network has smaller ratios than all other networks except the tree and the reduced mesh. This latter behavior might be due to the topology construction which uses a single link to connect each stub network to a transit node, yielding a structure similar to a tree. The Waxman network, however, has distance ratio metrics that are similar to that of the random graph.

Summary of Case Study Results

In summary, the distance ratio metrics of all the topologies we studied are relatively small. For the worst case among all topologies we studied, the mesh, the 90th percentile distance on the shared tree is less than 3.5 times that on the shortest path tree. Furthermore, these metrics qualitatively fall into three classes: low, medium and high. This classification is consistent with respect to the metric—that is, a topology that is classified as medium for the average distance ratio also falls into the same category for the 90th percentile. All real and generated networks except Waxman are in the medium category.

To understand which topology metric impacts the distance ratio metrics, we compare the ordering (from high to low) of these protocol metrics with a similar ordering of topologies by their topology metric values. The topology ordering with respect to distance ratio is more consistent with that for distortion (Figure 4) than for expansion or resilience. This,
together with the relative size insensitivity, suggests that distortion exerts the most influence on distance ratios.

This case study then, is our first glimpse at a design question that is impacted by topology. Admittedly, the differences between the performance metrics are small, yet it is interesting that one topology metric—distortion—impacts distance ratios more than other metrics. It is also interesting that, in this case study, real topologies are decidedly not tree-like (their distance ratios are not 1), but fall somewhere between the tree and the mesh.

5 Case Study: Forwarding State Aggregation

In this section, we investigate the topology sensitivity of multicast forwarding state aggregatability. Unicast forwarding tables aggregate well because addresses are, to some extent, assigned topologically [23]. Unfortunately, multicast forwarding information cannot be aggregated using similar mechanisms, because the outgoing interface set of a single entry changes dynamically and is determined by the existence of downstream receivers. Without any aggregation (and assuming a bidirectional shared tree routing protocol) the number of multicast forwarding entries in a router can be proportional to the number of active multicast groups.

Recent work has proposed an interface-centered multicast forwarding state aggregation technique [26]. This technique essentially represents the entire multicast forwarding table as a collection of per-interface bit strings. These bit strings indicate if, for the corresponding groups, a specific interface belongs to the outgoing interface set. The aggregation technique simply compresses these bit strings using run-length encoding. Such an aggregation technique is strict. By contrast, other work has considered leaky aggregation [22], where, for some groups, outgoing interface sets are not correctly preserved in the aggregation, resulting in traffic leakage in directions where there are no receivers.

Our focus in this section is on the topology dependence of strict aggregation. However, rather than evaluate a particular aggregation technique, we investigate bounds on the aggregatability of multicast forwarding state. To do this, we consider a simple information-theoretic approach. For a given group, let $p$ be the probability that a particular interface of a router is an outgoing interface for that group. Then, the number of bits needed to represent this information is simply the entropy [5]:

$$H(\text{Interface, Group}) = H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)$$

(1)

Our metric for aggregatability, interface entropy, is defined as the average entropy of an interface over all nodes.

To compute our aggregatability metric, we use the following methodology. We first uniformly select a node in the topology to be the root of the multicast distribution tree for a given group. Then, we assume that each node in the network has an identical probability of having an attached receiver (we call this the receiver probability). Tracing the reverse path from the receiver to the source, we can assign the corresponding probability to interfaces along the path. Where there exist multiple equal-cost paths, we assume that each of these paths has an equal probability of being chosen. In this manner, we can compute, for each
router, the average interface entropy for the group. For a given topology, we repeat this procedure for different receiver probabilities and average the results over different choices for tree root.

![Interface entropy graphs](image)

Figure 6: Average interface entropy

**Discussion of Results**

Figure 6 shows the average interface entropy for various topologies, as a function of receiver probabilities. Each of the canonical topologies shows a different behavior (Figure 6(b)). The entropy for both the mesh and the random graph increase monotonically with receiver probability, but for the mesh, unlike the random graph, the entropy approaches 1 in the limit of complete occupancy. This is because of the prevalence of equal-cost shortest paths in the mesh. On the other hand, the tree and the reduced mesh have single-peaked, although completely distinguishable behavior, with zero entropy in the limit.

All real topologies are single-peaked with non-zero entropy in the limit (Figure 6(a)). In this sense, their behavior represents some combination of a tree and a random graph. Of the three, the Mbone topology resembles a tree most. On the other hand, the Internet topology is closest to a random graph.

Among the generated topologies, the Waxman closely approximates a random graph (Figure 6(c)). Tiers and Transit-stub qualitatively resemble the single-peaked, non-zero limiting behavior of real topologies.

We also investigated the worst-case interface entropy, defined as the maximum interface entropy averaged among a number of groups rooted at different nodes. With this metric, the distinctions between topologies disappear because in almost all cases the worst-case entropy is close to 1.
Summary of Case Study Results

These results reveal that, for very low occupancies (about 3%), routing state for a single group can be compressed by up to a factor of 10. For receiver probabilities on the order of 10%, the aggregatability decreases to 3. This qualitatively matches the findings in [26] where the interface-centered technique achieves some, but not dramatic, aggregation.

Also, we notice that the monotonicity vs. single-peaked distinction appears to be more consistent with the distortion (Figure 4) differences than with differences in the other topology metrics. Topologies with high distortion (i.e., the random graph, the mesh, and Waxman) have monotonically increasing interface entropy, while the rest of the topologies that have lower distortion also have a single-peaked average interface entropy.

This case study also reveals the impact of topology on protocol performance. Notice that the methodology employed to study aggregatability is unlike that used in the previous study. Also, unlike in Section 4, we find that there are two facets to the performance of aggregatability that appear to be topology dependent: the monotonicity versus single-peaked distinction, and the limiting behavior of aggregatability. We think distortion differences explain the former, but have no explanation for what aspect of topology influences the latter. This may mean that our topology characterization is not complete. Finally, recall that we chose uniform receiver placement. For this reason, our results may not reflect reality, but, lacking models for receiver clustering, uniform placement was our only available choice.

6 Case Study: Endsystem Multicast

Recently, some research efforts have proposed endsystem based multicast schemes [12, 16]. These approaches do not require network support for group communication. Rather, the endsystems (hosts) to which participants are attached conspire to set up a distribution tree comprised of point-to-point links between them. In both schemes, the endsystems continually refine the tree in order to improve the overall delivery latency and reduce network overhead. Endsystem multicast schemes are attractive because of their inherent deployability.

In this section, we study the impact of topology on the efficiency of endsystem multicast schemes. To our knowledge, ongoing research has not considered this issue. Our evaluation is meant to inform, but not resolve, the larger architectural debate about the relative merits of such schemes vis-a-vis network-layer multicast. To analyze the efficiency of endsystem multicast, we define two performance metrics, following [16]:

Tree Stretch is the ratio of links in the endsystem tree to that in a native multicast shared tree.

Tree Stress is the maximum number of endsystem tree links that traverse any physical link in the topology.

The stretch and stress metrics depend largely on the particulars of the tree construction methodology. Rather than attempt to faithfully model the methodologies proposed in [12, 16], we consider two simple heuristics. The first heuristic approximately models the initial
tree construction procedure in [12, 16], and the second approximates the result of their continual tree refinement procedures.

**Closest Receiver** This simple heuristic adds new receivers in their join order. Each receiver is connected to the closest node already on the tree whose degree is smaller than some limit. In our evaluation, we choose a limit of 5.

**Minimum Spanning Tree** This assumes that all receivers are known in advance, as are the corresponding inter-receiver communication costs. Receivers are added to the distribution tree using a Minimum Spanning Tree (MST) algorithm.

In this section, we only describe the results for the closest receiver heuristic. The results for the MST heuristic are described in Appendix B.

To compute stretch and stress, we first uniformly select a fraction of receivers in a given topology. We then order the receivers randomly and compute the endsystem tree according to the closest receiver heuristic. To compute the corresponding native multicast shared tree, we randomly select one of the receivers to be the source. We average the stretch and stress thus obtained across a variety of receiver orderings and choices of source. We repeat this steps for different receiver occupancies.

![Graphs showing endsystem tree stretch for different receiver occupancies](image)

**Figure 7:** Endsystenm tree stretch for closest receiver heuristic.

**Discussion of Tree Stretch Results**

Figure 7 plots the tree stretch as a function of occupancy. The canonical topologies exhibit two kinds of behavior (Figure 7(b)). First, in some topologies, the stretch is actually less than 1 at low (about 5%) occupancies. In this category fall the mesh and the random graph, because they, unlike the tree and the reduced mesh, have alternate paths between the receivers. Second, the mesh and the reduced mesh exhibit a dependency of the stretch on occupancy, while the other canonical topologies do not.
For all real topologies, the tree stretch is larger than 1 even at low occupancy (Figure 7(a)). Further, their stretch does not depend on the occupancy. In this sense, the stretch of all real topologies lies between a tree and a random graph. Finally, the stretch for all real topologies is less than 1.6, even given that we have a small, fixed limit on the degree of each node in the endsystem tree. In particular, it is interesting to note that the endsystem tree over the Internet core shows low stretch (approximately 1.4), regardless of occupancy. This is encouraging; endsystem multicast schemes are at least not completely unreasonable from an efficiency perspective.

Among the generated topologies, Tiers and Transit-Stub qualitatively match the behavior of the real topologies (Figure 7(c)). The same cannot be said of the Waxman network, which behaves more like a random graph.

Finally, although we do not present the results here, we considered a variant of the closest receiver heuristic without any limit on the degree of a endsystem tree node. The tree stretch using our heuristic is about 15% higher than that of this variant.

![Graph](image)

(a) Real topologies  (b) Canonical topologies  (c) Generated topologies

Figure 8: Endsystem tree stress for closest receiver heuristic.

Discussion of Tree Stress Results

Figure 8 plots endsystem tree stress as a function of occupancy. Beyond about 10% occupancy, all canonical topologies are insensitive to the occupancy (Figure 8(b)). However, while for the random graph and the mesh the stress is less than or equal to the node degree limit of 5, for the reduced mesh and the tree the stress is noticeably higher.

All real topologies have similar stress, which is close to the degree limit. Compared to the canonical topologies, this places them between a tree and a mesh (Figure 8(a)). Among the generated topologies, Tiers and Transit-stub, like the real topologies, are between the tree and the mesh, while Waxman is more closer to the random graph (Figure 8(c)).
Summary of Case Study Results

Which topology metric determines stretch? At low occupancy (on the order of 10%), the ordering of tree stretch values by topology more closely matches the ordering of topologies by distortion (Figure 4) than by resilience or expansion. This correspondence with distortion is inverted: topologies with high distortion have low stretch and vice versa.

Furthermore, there is some prior work that suggests that the dependence on occupancy may be a function of expansion [13]. Our canonical topologies reveal this. The mesh and the reduced mesh, with non-exponential expansion, exhibit this dependency, whereas the other two topologies do not. However, some of the real and generated networks which exhibit algebraic expansion do not reveal this dependence. We do not understand why this is, but one possible explanation is that edge effects, such as those described in Appendix A, may be masking this dependence.

On the other hand, it is a little unclear which topology metric affects stress. It may be that our limit on the node degree suppresses topology effects. For all real networks, the closest receiver heuristic results in a stress that is insensitive to occupancy and is close to the bound on node degree. Finally, for this case study as well, we emphasize that our uniform receiver placement methodology may be unrealistic.

7 Case Study: Alternate Path Routing

Alternate path routing has been applied to call routing in telephone networks, where it has proved successful in reducing call blocking rates and increasing network utilization [17]. Recently, alternate path routing has been proposed for IP multicast routing with a similar intent [30]. That work discusses a receiver-driven alternate path routing protocol to route around congested links in a multicast tree. It describes a mechanism called APM for discovering alternate, uncongested paths from receivers to sources. The efficacy of APM is demonstrated by simulating these on several network topologies chosen to stress the discovery mechanism. It also specifies a protocol for installing these alternate paths in a loop-free manner.

This paper asks a broader question not addressed in [30]: How does topology impact the alternate path routing problem? Specifically, we are interested in how well alternate path routing works if we assume omniscience in the discovery of such paths, as well as if we use a local, scalable discovery method such as APM. Omniscient path discovery is achievable in practice using link-state routing information. We emphasize that our exploration is not intended to address the issue of whether alternate path routing mechanisms should be deployed on the Internet.

To understand the impact of topology on alternate path routing, we define two protocol metrics. The success rate is the probability of finding an uncongested path to the source. The extra path length is the additional length of the uncongested path, as a fraction of the shortest path. These two metrics can be studied from two perspectives. The success rate and extra path length from the perspective of a single receiver are relevant for multicast lectures. On the other hand, the success rate and extra path length from the perspective of the entire tree (i.e., all receivers) are interesting for multicast conferences.
Because this design space is rather large, in this section we only present results for omniscient path discovery from the receiver perspective. Appendix C discusses how omniscient path discovery performs from the tree perspective for different sized trees. Finally, Appendix D describes the performance of the APM mechanism from the receiver perspective.

To compute success rate and extra path length from the receiver perspective, we randomly mark as congested a certain percentage of links in the topology. Then we uniformly select a receiver and a source. If the shortest path between them is affected by a congested link, we try to obtain an alternate path. This process is repeated for several times, each time with many uniformly selected sender and receiver placements. We repeat each of these simulations for all of our ten topologies, for varying fractions of congested links.

[Graphs showing success rate with increasing percentage of congested links for different topologies]

(a) Real topologies  (b) Canonical topologies  (c) Generated topologies

Figure 9: Success rate with increasing percentage of congested links for the omniscient scheme, receiver perspective.

Discussion of Results for Success Rate

Figure 9(b) shows the success rate of omniscient path discovery as a function of the fraction of congested links for our canonical topologies. These topologies exhibit two distinct behaviors. The tree and the reduced mesh have zero success rate: in these topologies, there is exactly one path between source and receiver. The success rate in the mesh and the random graph depicts a phase transition: with few congested links, receivers are able to find an alternate path to the source. However, beyond a certain fraction of congested links—and this threshold is different for the mesh and the random graph—the success rate abruptly drops to zero.

Unlike the mesh and random graph, the success rate for real networks decreases gradually (Figure 9(a)). However, it is interesting that the Internet topology has high success rate, like the mesh and the random graph, in the regime with few congested links. While the shape of the Internet and Mbone curves are similar, the AS exhibits a markedly more gradual falloff.

Not surprisingly, the Waxman network shows a random graph-like success rate behavior with a marked phase transition (Figure 9(c)). Tiers and Transit-stub topologies model the gradual falloff of real networks well. In fact, the success rate of the Tiers network is almost
identical to that of Mbone. However, the Internet and AS topologies have higher success rates over a wider regime, a behavior that Tiers and Transit-stub are unable to capture.

![Graphs showing percentage of congested links vs. average path length increase.](attachment:graphs.png)

(a) Real topologies  
(b) Canonical topologies  
(c) Generated topologies

Figure 10: Extra path length with increasing percentage of congested links for omniscient path discovery, receiver perspective.

**Discussion of Results for Extra Path Length**

Figure 10 shows the extra path length of omniscient path discovery as a function of the fraction of congested links. Of the canonical networks, the extra path length increases both for the random graph and the mesh (Figure 10(b)). Corresponding to the threshold at which these networks exhibit a phase transition in their success rate, they also show a marked increase in extra path length. At extreme congestion levels, the alternate path of the random graph is nearly three times the shortest path.

Because real networks are not susceptible to phase transitions in the success rate, their extra path length increases much more gradually throughout the regime of path availability (Figure 10(a)). Even though the AS topology shows a different success rate behavior than the Internet, their extra path lengths closely match each other. Furthermore, at even relatively high congestion levels (60%), the average alternate path in the Internet is less than 50% longer than the congested shortest path.

The Waxman network shows a more pronounced increase in path length than the random graph (Figure 10(c)). Tiers and Transit-stub match the gradual increase in extra path length of the real networks. For these two networks, the average path length of the alternate path is within a factor of two of the congested shortest path.

---

8In Figure 10, we excluded the extra path length for the regime with less than 0.5% success rate. Those data points tend to have less samples and therefore exhibit more erratic behavior.
Summary

To determine which, if any, topology metric affects the success rate, we consider the ordering of topologies when their success rates drop to zero. We observe that this ordering is approximately the same as the ordering of resilience (Figure 3), and it is distinct from the ordering of topologies with respect to expansion and distortion (Figures 2, 4). This seems to indicate that success rate is most significantly affected by resilience, as one would expect.

The behavior of the extra path length seems well correlated to that of success rate. By this we mean that topologies that incur phase transitions in their success rate show marked increase in extra path length. However, it is less clear which, if any, topology metric influences this phase transition behavior. Perhaps the high distortion of the mesh and random graph contributes to this behavior. On the other hand, it may just be the structural uniformity of these graphs, but not their high distortion, that determines this behavior.

While the generated and canonical topologies we have studied show widely varying behavior, real topologies—and the Internet in particular—exhibit desirable structural properties for alternate path routing. Specifically, the Internet has a high success rate over a wide variety of congestion levels, but only gradually increasing average alternate path lengths. Two topology generators, Transit-stub and Tiers, qualitatively capture the same behavior, but over a markedly smaller range of congestion levels.

What does this case study reveal? As with the aggregatability study, there exist aspects of performance—the phase transition behavior of the success rate in the mesh—that are not captured by our topology metrics. Unlike previous studies, here it is resilience that appears to maximally influence performance. Finally, in this study, none of the canonical topologies form even approximate models for the performance of real networks.

8 Conclusions

In this paper, we examined characteristics of real and generated topologies, and also examined the impact of topology on protocol design and performance. Several of our results are worth emphasizing:

- In general, generators fail to match the topology characteristics of real networks.
- There doesn’t exist one canonical topology that consistently resembles real topologies, or a pair of them that consistently bounds the performance of real topologies.
- Distortion is not a topology property discussed much in the networking literature, but appears to most influence many of our protocol case studies.
- The Internet topology has tree or random graph like expansion, mesh like resilience, and a distortion that is intermediate between the mesh and the tree.

Our work represents only an initial attempt at understanding network structure and how it affects performance. Much work remains. In particular, understanding the physical intuition behind network structure can help explain why topology generators fail to capture real
networks. In addition, exploring other metrics for characterizing topologies might explain some aspects of protocol performance that our metrics did not account for.

Beyond the obvious relevance of our work to real topologies, these results are also relevant to virtual topologies formed by server infrastructures (e.g., Web caches). For these, it might be possible to engineer the topologies along the lines of some of our canonical topologies in order to obtain desired performance.

References


A Expansion in the Internet: Exponential or Power Law?

Figure 11: Expansion for various Internet maps

Figure 12: Expansion for various random graphs
In order to resolve the expansion characteristics of the Internet, we computed the expansion for a variety of Internet topologies. The $L_r$ topology was obtained from traceroutes to destinations chosen from BGP routing tables [2]. To the $L_r$ topology, we applied alias resolution [14] to obtain the $L$ topology. The $S$ topology is obtained as described in [14]. Finally the $SL$ topology is a merged version of the $S$ and $L$ topologies. Figure 11 plots the expansion of these Internet topologies on log-linear and log-log scales. These Internet topologies are qualitatively consistent with each other in both representations. However, these figures do not conclusively indicate whether Internet expansion is exponential or algebraic.

To understand Internet expansion better, we plotted the expansion of random graphs, which are known to exhibit exponential expansion asymptotically. Figure 12 shows the expansion of random graphs of different sizes with average degree 10 as well as their asymptotic expansion bound $10^h$ on the log-linear scale and log-log scale. Notice that these figures qualitatively match the corresponding figures for the Internet map. For example, the log-log figures for both exhibit the same pronounced non-linearity for low $h$.

This leads us to conjecture that the dominant behavior of the Internet is indeed exponential expansion. However, for $h$ greater than some small value, the curves diverge rapidly from the exponential due to what we call edge effects. As $h$ increases, a ball of radius $h$ is more likely to encounter the edge of the map, resulting in smaller $E(h)$ than the exponential. This effect is more apparent in irregular topologies such as the random graph, and less so in, for example, $k$-ary trees.

### B Endsystem Multicast with Minimum Spanning Tree Heuristic

![Graphs showing tree stretch and stress results for different topologies](image)

**Figure 13: Endsystem tree stretch for minimum spanning tree heuristic**

In Section 6 we presented the tree stretch and stress results for the closest receiver with limiting node degree heuristic. Here we describe the results for the Minimum Spanning Tree (MST) algorithm. Space precludes a detailed discussion, so we only briefly highlight some
of the results.

Figure 13 shows the tree stretch as a function of occupancy. At an occupancy of 1, the number of links in the MST is $N - 1$, as is that in the IP multicast tree. In the limit, then, stretch is 1. The mesh and random graph have stretch less than 1 at low occupancy such as mesh and random. On the other hand, the tree and the reduced mesh have stretch higher than one, as do the real topologies.

If we compare the MST stretch with the closest receiver tree stretch on Figure 7, we can see that for low occupancy (20% or less) the results are qualitatively the same (e.g., the topologies with low closest receiver tree stretch have also low MST stretch, and vice versa). Quantitatively, the closest receiver tree stretch is about 30% higher compared to the MST. For higher occupancy this difference increases. It is interesting to note that for all real topologies the difference in stretch between the closest receiver and the MST heuristic for low occupancy is remarkably low (of the order of 20–30%).

![Graphs showing tree stress for different topologies and occupancy levels.](image)

(a) Real topologies  
(b) Canonical topologies  
(c) Generated topologies

Figure 14: Endsystem tree stress for minimum spanning tree heuristic

Figure 14 plots the MST stress as a function of occupancy. All real topologies (Figure 14(a)) have very high stress, and this depends significantly on the occupancy. At first glance, it may seem that the real topologies are in the same category as the tree and the reduced mesh. However, after some more careful investigation, we found that in all cases the high stress of the real topologies was because of some node with very large node degree.

In summary, the MST stretch improvement over the closest receiver heuristic is relatively small for low occupancy, and is of the order of 20–30%. However, because for some real topologies the MST heuristic can induce high stress, this makes it impractical.

---

9The highly overloaded link was between such node and some of its physical neighbor-receiver, because such neighbor is potentially closer to many other nodes, and therefore it will have much higher degree on the endsysten tree.
C Alternate Path Routing: Tree Perspective

In this section we study the success rate and extra path length of the omniscient alternate path routing scheme from the tree perspective (Section 7). To compute these metrics, we fix the fraction of congested links to be 20% of the total links in a topology, then vary the number of receivers in the topology as a fraction of the number of total nodes.

![Graphs](image)

Figure 15: Success probability with increasing occupancy for omniscient alternate path discovery, tree perspective (20% links congested).

Figure 15 shows the success rate as a function of receiver occupancy. We observe two major trends in this figure. First, the behaviors of the canonical topologies fall into two distinct classes. The mesh and the random network (as well as the Waxman) exhibit a phase transition in success rate. Second, unlike the receiver perspective, now the real and the generated topologies (Tiers and Transit-stub) lose their gradual transition. Instead, they all show similar behavior, and their success rates drop sharply at almost the same occupancy.

Here is a possible explanation to this rather interesting phenomenon. The success rate from the tree perspective depends on two factors: the success rates of individual receivers from the receiver perspective, and the sharing of paths in the multicast tree. In the extreme case of no sharing, the success rate of the entire tree is the product of the success rate from each receiver. On the other extreme of complete sharing, \( i.e., \) when the multicast tree degenerates to a string, the success rate of the tree is equal to that of a single receiver. \( [20] \) indicates that the degree of sharing (\( \bar{L}(n)/n\bar{u} \)) in the Internet and AS is much higher than that of Mbone. The higher the degree of path sharing, the more slowly the success rate drops. Our results seem to confirm this observation. Unfortunately, because none of our topology metrics measure the degree of path sharing, this phenomenon cannot be explained using our topology taxonomy.

All topologies that we have studied exhibit similar behavior for extra path length from the tree perspective (Figure 16). They drop sharply as occupancy increases, then become constant. We speculate that this drop is due to the path sharing in the alternate multicast tree. In other words, when the second receiver joins the alternate multicast tree, to this
receiver the sender’s location is no longer random, thus its extra path length is affected. However, we were not able to quantitively confirm this conjecture.

Quantitively, the Mbone has the highest extra path length. Other real networks, as well as the Tiers network, fall into the same class as the random and the Waxman networks. The mesh has the lowest extra path length. This ordering clearly does not match that of any of the topology metrics.

D Alternate Path Routing using APM

In the previous sections, we focused on an omniscient version of alternate path routing protocol using link state information. Here we turn to another design of alternate path multicast routing protocol, APM [30]. It uses local information to discover alternate routes, thus is more scalable than a link state approach. In the following we only discuss success rate and extra path length from the receiver perspective, partially because it is difficult to reason about the effects of path sharing (Appendix C).

Figure 17 shows the success rate as a function of the percentage of congested links in the topology. Comparing this with Figure 7, we found that most topology have slightly decreased success rate, but their behaviors are qualitatively similar. Because APM uses local information to infer alternate path, it may not be able to discover alternate paths in all circumstances. The mesh, on the other hand, exhibits dramatically different behavior. It still exhibits phase transition, but its success rate drops sharply even when the fraction of congested links is well below 10%. This turned out to be an artifact of our route computation algorithm, which always choose a fixed route when there are multiple equal-length paths.

Figure 18 shows the extra path length as a function of the percentage of congested links. Comparing this with Figure 10, we found that except the Mbone, all topologies experience slightly increased extra path length due to the local path discovery of APM.
Figure 17: Success rate with increasing percentage of congested links for APM, receiver perspective

Figure 18: Extra path length with increasing percentage of congested links for APM, receiver perspective.