On Investigating Overlay Service Topologies

Zhi Li

Department of Computer Science, University of California, Davis, CA 95616
Email: lizhi@cs.ucdavis.edu

Prasant Mohapatra

Department of Computer Science, University of California, Davis, CA 95616
Email: prasant@cs.ucdavis.edu

Abstract

Recently, a moderate amount of work has been reported on the use of overlay networks to support value-added network services, such as multicasting, Quality of Service (QoS), security, etc. To design an overlay network, the first step is to choose an overlay topology connecting all the overlay service nodes. When considering overlay topologies, several questions need to be answered first: How overlay topologies affect overlay routing performance? Which topologies can provide satisfactory performance? How can we construct efficient overlay topologies connecting all the overlay nodes?

In this paper, we focus on the overlay network topology construction issue. First, we evaluate and compare the performance and overhead of the existing overlay topologies. Second, we formalize the overlay topology construction problem and propose two new heuristic methods to construct efficient overlay topologies. Simulation results have demonstrated the efficiency of the two proposed approaches. It is shown that overlay service performance varies significantly with respect to different overlay topologies. Thus, it is important to choose an appropriate overlay network topology. The knowledge of IP-layer topology information also benefits significantly in constructing efficient overlay topologies as inferred from the results.

Key words: Overlay Service Network, Overlay Routing, Overlay Topology, Resilient Overlay Service Network

⋆ This research was supported in part by the National Science Foundation through the grants CCR-0296070 and ANI-0296034. A preliminary version of this work was presented at IEEE INFOCOM 2004.

1 Introduction

Application-layer overlay technique is an effective way to support new applications as well as protocols without any changes in the underlying network infrastructure. For example, Qbone[4] and Mbone[16] utilize overlay technique to support Quality-of-Service (QoS) and multicast services, respectively, on top of the existing Internet infrastructure.

An overlay network is formed by a subset of underlying physical nodes. The connections between each pair of overlay nodes are provided by overlay links (IP-layer paths), each of which is usually composed of one or more IP-layer links. As the overlay applications are usually built at the application layer, it can effectively use the Internet as a lower level infrastructure to provide advanced services to end users, such as peer-to-peer (P2P) file sharing[1], overlay multicasting[13,12,19,34], resilient overlay network (RON)[6], service overlay network (SON)[15], Quality-of-Service (QRON[21], OverQoS[31]), overlay media streaming[7], etc. An OSN (Overlay Service Network) includes a set of random nodes from different locations of Internet or a group of fixed overlay nodes strategically placed by a third party. In the latter case, the third party can purchase access service for the overlay nodes from one or more different Internet Service Providers (ISPs). These overlay nodes cooperate with each other to provide an overlay service platform, on top of which a variety of application-specific overlays can be constructed, such as multicast overlays, anycast overlays, and end-to-end QoS-aware overlays.

Since different overlays may share IP-layer links with each other or other Internet applications, overlay service nodes usually cannot directly control the underlying IP-layer path resources. Up-to-date overlay link performance information can only be obtained through measurements. With the increase in the number of overlay links, the overlay performance monitoring overhead increases. As the overlay nodes are connected via IP-layer paths (overlay links), theoretically, there exits an overlay link connecting each pair of overlay nodes. Different selections of overlay links (topologies) will affect the overlay service quality and monitoring overhead. A richly connected overlay network composed of high quality paths is necessary for overlay applications to quickly react to performance fluctuation, concurrent multiple path routing, or perform application-specific paths selection.

To connect all the overlay nodes, we have many candidate topologies. For example, we can use several minimum spanning trees to connect all the overlay nodes [33]. We can also leverage the underlying topology information to construct overlay topologies. Alternatively, we can use a full mesh or a random topology to connect all the overlay nodes as in RON [6] and end-system multicast[13]. Even though a moderate amount of research work has been done on overlay service networks, few work has been dedicated to the overlay network topology issue. How will different topologies affect the service performance of overlay networks? Does the performance of overlay routing protocols differ a lot with respect to various overlay topologies? Can any other overlay topologies provide better overlay services? These are some of the questions that have motivated the work of this paper.

In this paper, we focus on the topology design issue for overlay service networks. This type of overlay networks are designed to provide service platform for other application-specific overlays (multicast, etc.). The main function is to provide resilient (or QoS-satisfied) routing service between each pair of overlay nodes. We use resilient overlay routing service as an example to evaluate and explore the impact of different overlay topologies on overlay networks service performance. We believe the same methods can also be applied for other application-specific overlay networks, such as content distribution networks, media streaming overlay services, end system multicasts, etc.

After generalizing and analyzing exsiting overlay topologies, we formalize the overlay service network topology construction problem and propose two heuristic methods to construct efficient overlay topologies. By utilizing IP-layer information and maximizing the number of IP-layer disjoint paths, these two algorithms can effectively facilitate overlay routing networks to achieve fault tolerance with less routing overhead. Using several other overlay network topologies, such as K-spanning tree, mesh-tree, adjacent connection, typology-aware K-spanning tree, we perform extensive simulation studies to evaluate and compare the impact of different overlay topologies on resilient overlay routing services. The simulations are carried on top of two IP-layer topologies: a random topology generated by GT-ITM [2] and a real ISP topology published in Rocketfuel[29]. The simulation results have demonstrated that the topologies have significant impact on the performance of the overlay routing services. In addition, the simulation results also confirmed that the two proposed approaches can construct efficient overlay service network topologies.

The rest of the paper is organized as follows. In Section II, we introduce the resilient overlay service network architecture, base on which we will study the performance impact of overlay network topologies. The overlay service network topology construction problem is formalized in Section III. The existing candidate overlay service network topologies are generalized in Section IV. In Section V, we introduce two new heuristic methods to construct efficient overlay topologies. In Section V, we perform simulations to study and analyze the different overlay topology construction methods. The related works are discussed in Section VII. Finally, the conclusions and the future work are outlined in Section VIII.

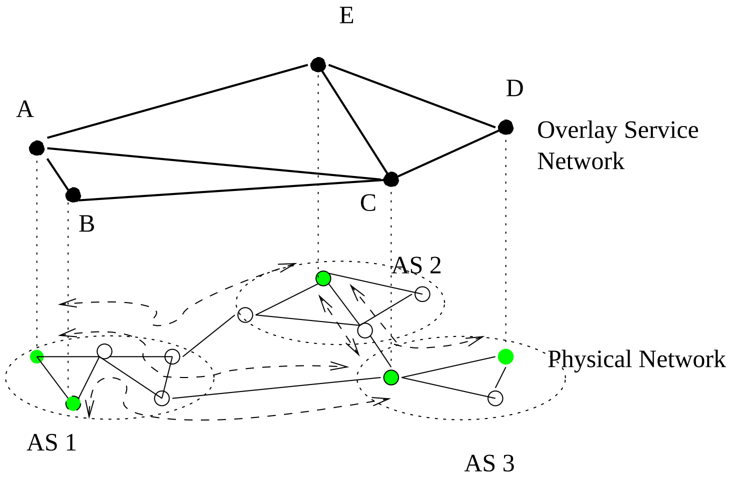

Fig. 1. Overlay Service Network

2 Resilient Overlay Service Network

An overlay service network is composed of a number of specialized overlay nodes that are located at different locations of the Internet or placed in the Internet by a third party (termed as Overlay Service Provider, OSP) to provide generic overlay services support to a variety of applications. Overlay nodes can be placed either at the edge of a domain or in the core.

Fig. 1 shows an example of overlay service network topology, which is composed of 5 overlay nodes from 3 ASes. The overlay nodes are connected via overlay links, each of which is an IP-layer path connecting an overlay node pair. In this example, the overlay nodes are connected together based on a random topology. From the figure, we observe that some of the overlay links are overlapped at the physical layer even though they are completely disjoint at the overlay service layer. This is one of the special characteristics of overlay networks. In addition, we can see that each of the overlay links is usually composed of several IP-layer links. For an overlay link, other non-overlay traffic or other overlay links may pass through part of or same group of IP-layer links. This means that the overlay links' capacities are not fixed and cannot be controlled by the overlay nodes. To provide satisfactory service to overlay users, overlay nodes need to continuously probe the network to obtain the latest performance of the overlay links.

RON [6] is designed to utilize the Internet path redundancy properties to provide resilient overlay paths. It was shown that the overlay networks provide overlay paths with better performance by passing through one or more intermediate overlay nodes when the default Internet paths are faulty. RON can provide application-specific paths for each applications on top of the overlay service networks, such as delay optimized overlay paths, loss-rate optimized overlay paths, etc. Resilient overlay routing service can not only be seen as an application-specific overlay built on top of overlay service network, but also an important module of overlay service networks, on top of which a variety of application-specific overlays can be built.

Based on the same inference, we can use resilient overlay service networks to provide resilient routing services to a large group of Internet users. The resilient routing service networks work as per the following steps.

- By default, the users always use the IP-layer routing service to send data traffic to destinations.
- When a user realizes that there is a service outage (such as long end-to-end delay, or low throughput) in the default IP-layer path, it will send the subsequent data traffic to its nearest overlay node (source overlay node, the first on-path overlay node).
- The source overlay node will forward the traffic to the destination overlay node (the last on-path overlay node) via a fault-free overlay path.
- The destination overlay node will forward the data to the traffic destination.

The goal of overlay topology construction is to provide resilient overlay paths connecting each pair of overlay nodes under various of IP layer path failures or under degraded performance conditions. In this paper, we will not get into the detailed architecture of resilient overlay service networks. We use it as an example to evaluate the performance of overlay routing protocols for different overlay topologies. Similar overlay topology construction methods can also be applied to other overlay service networks, such as content distribution networks, media streaming service networks, etc.

3 Problem Formulation

In this section, we formalize the overlay topology construction problem. We assume that the locations of overlay nodes are pre-determined: either formed by the participant nodes' locations of overlay service networks or selected by a third party using some specific overlay nodes location selection methods. In addition, we assume that the IP layer uses the least-cost based routing protocols.

The inputs to the problem are listed as follows.
Given:

- IP-layer topology $G_P(V_P, E_P)$ (V_P is the set of nodes while E_P is the set of IP-layer links.).
- A set of overlay nodes V_O, which is a subset of V_P.
- N is the IP-layer topology size (the total number of V_P).

- M is the overlay network size (the total number of overlay nodes, V_O).
- The IP layer topology is represented as $E_P = (e_{p_{ij}})$, which is an N*N matrix representing the IP-layer topology. $e_{p_{ij}} = 1$ if there is a link between node i and j.
- P_{ij}^{mn} is a IP layer path-link indicator, where $P_{ij}^{mn} = 1$ if the IP layer path between m and n passing through IP-layer link ij.

The following **Variables** are necessary for solving this problem

- $E_O = (e_{o_{mn}})$ is an M*M matrix that represents an overlay topology, where $e_{o_{mn}} = 1$ if there is an overlay link connecting overlay node m and n.
- Q_{mn}^{xy} is an overlay path-link indicator, where $Q_{mn}^{xy} = 1$ if there exists an overlay path from x to y passes through overlay link mn.
- $b_{ij}^{xy} = 1$ if the IP link between i and j is broken, there is no existing overlay paths subjecting to above constraints that connects overlay node x and y.

Solving this problem is subject to the following **Constraints:**

- d_m, the maximal number of overlay node neighbors each node can have. That is: $\sum_n e_{o_{mn}} \leq d_m$.
- h_{xy}, the overlay path length (latency) constraints (in terms of IP-layer hops). That is $\sum_{mn} Q_{mn}^{xy} \sum_{ij} P_{ij}^{mn} \leq h_{xy}$.

Suppose the IP-layer link failure states are: $S_1, S_2, S_3, ..., S_n$, each of which will affect the connectivity (or performance degradation) of some sets of IP layer links. This will result in the loss of the IP-layer path connectivity between some pairs of nodes (a subset of V_p).

The **Objective** of the topology construction problem can be formalized as follows:

- Find an overlay topology $G_O(V_O, E_O)$ (i.e. a set of overlay links, E_O) which can achieve $MIN(\sum_{S_k} \sum_{xy} \sum_{ij}) b_{ij}^{xy})$ subject to above two constraints.

That is, we want to construct an overlay topology with node degree constraint that can maintain the maximal connectivity between any two overlay nodes via a distance constraint path under various IP-layer path failures.

4 Existing Overlay Service Network Topologies

To connect all the overlay nodes to form an overlay service network, there are many possible topologies we can adopt. In this section, we first list and generalize several overlay topology construction methods that have appeared in the literature. Each of them could be a viable candidate topology for an overlay service network.

4.1 Full-Mesh (FM)

As overlay service networks run on top of IP-layer network, in normal situation, there is an IP-layer path connecting each pair of overlay nodes. Thus, each pair of overlay nodes could be neighbor with each other at overlay layer. Based on this notion, all the overlay nodes can form a full-mesh topology. We can easily observe that the computational complexity of this approach is $O(V_o^2)$. As mentioned earlier, the overlay nodes cannot directly control and retrieve the overlay link resource information because the unexpected non-overlay traffic may pass through the same IP-layer links. To retrieve the overlay links' performance and detect path anomalies (such as latency, bandwidth), the overlay nodes need to continuously send probing packets to neighboring overlay nodes. RON uses link-state based routing protocol, in which each node sends its local link state to every other overlay nodes. It was shown that for an overlay network composed of 50 nodes, each node will have around 33Kbps routing overhead[6]. Fig. 2 is an example of a full mesh overlay topology. From the figure, we can observe that each pair of overlay nodes are neighbors with each other in the overlay topology. Several overlay links pass through the same common underlying IP-layer links.

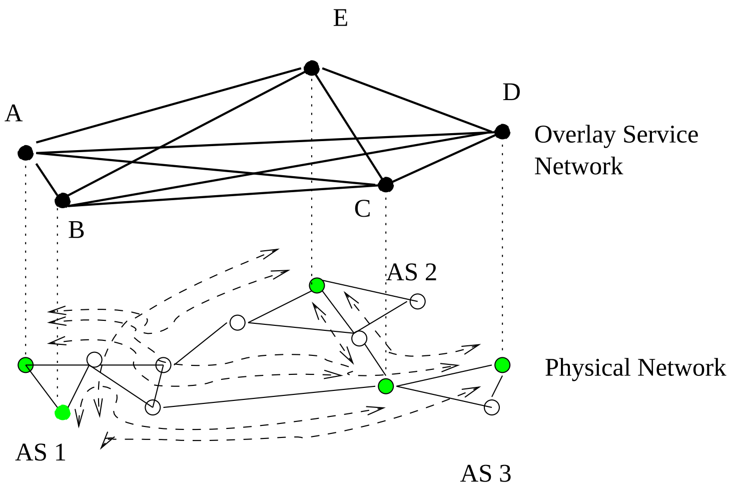

Fig. 2. Full Mesh Overlay topology.

It is easy to see that the full mesh topology does not meet the overlay nodes' degree constraint. Some protocols (such as end-system multicast [13]) have used the following method to construct overlay topologies: 1) each overlay node randomly selects a set of nodes as its candidate overlay neighbors; 2) Then, it chooses a subset from them as its neighboring nodes subjective to node degree constraints based on the path distances. We call this method as *Random Connection (RC)*. We can see that this method incurs much less measurement overhead than the full mesh topology and the computational complexity is less than $O(V_o^2)$.

4.2 K-Minimum-Spanning-Tree (KMST)

A minimum spanning tree is the lowest cost tree among all the candidate trees that connect all the nodes. To minimize the state maintenance and overlay link performance measurement overhead, prior efforts[33] have proposed a network topology that is composed of K minimum spanning trees in the full mesh topologies. The K trees have minimal overlaps in the overlay links and compose an overlay service network topology. The cost of an overlay link is defined as the number of IP-layer hops the overlay link passes through. K can take different values based on the different cost-performance tradeoff and node degree constraints. The computational complexity of this approach is $O(K * V + o^2)$. Fig. 3 shows a 2-minimum spanning tree overlay topology, in which the dashed lines and solid lines belong to two different spanning trees, which together compose the overlay topology.

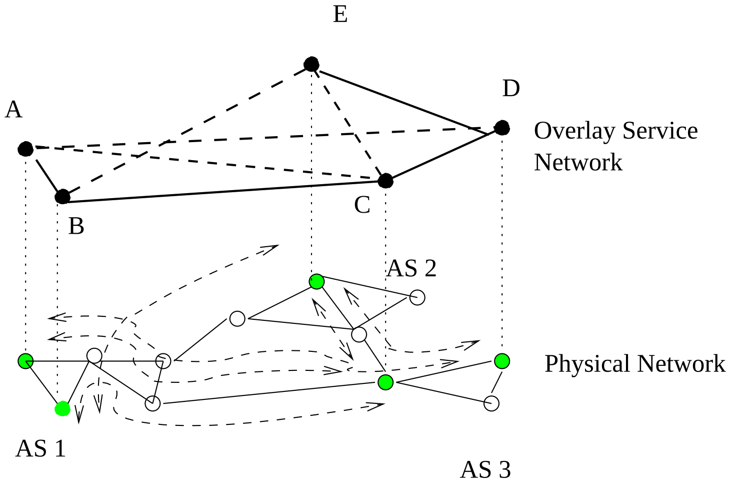

Fig. 3. 2-minimum Spanning Tree Overlay topology.

4.3 Mesh-Tree (MT)

Two similar approaches (we called Mesh-Tree approaches) in [22] and [9] have been proposed to enhance the resilience of the overlay multicast. A Mesh-Tree topology can be constructed as follows: 1) Set up a minimum spanning tree connecting all the overlay nodes; 2) If two overlay nodes have grandchild-grandparent or uncle-nephew relationship in the minimum spanning tree, there is also an overlay link connecting these two overlay nodes. The computational complexity of this approach is $O(V_o^3)$ and has been proved be an efficient method to provide resilient overlay multicasting service.

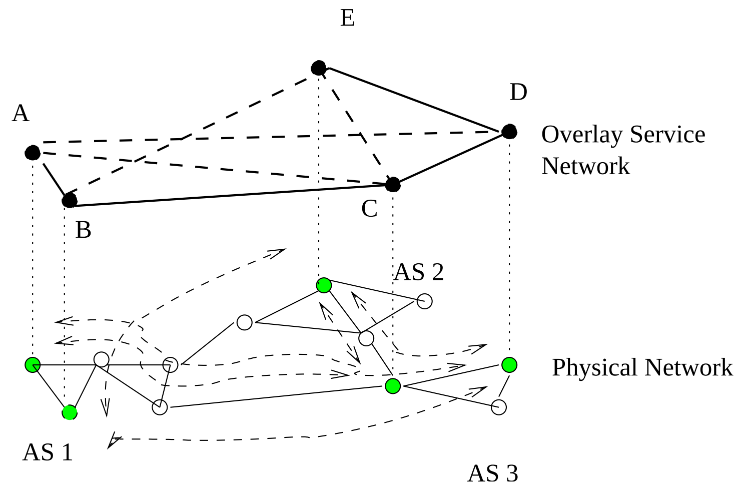

Fig. 4. Mesh-Tree Overlay topology.

Fig. 4 shows an example of MT topology, in which the solid lines compose a spanning tree and the dash lines are the added mesh links.

4.4 Adjacent-Connection (AC)

Adjacent connection topologies use the knowledge of IP-layer topology for the overlay topology construction and assume that the overlay nodes can obtain the IP-layer path information. The overlay topology construction method is: if no overlay node is directly connected to the nodes on the IP-layer path between any pair of overlay nodes, there is an overlay link connecting these two overlay nodes. In [23] and [21], the authors use this method to construct overlay service topologies. If we only have the IP-layer topology information, the computation complexity of finding all the paths connecting all the nodes is $O(V_p^3)$. If the average number of IP layer paths is h, the computation complexity of this method is $O(V_p^3 + h * V_o^3)$.

Fig. 5 shows an example of the AC topology. In this example, no overlay node is on the IP-layer path of the overlay links connecting any other two overlay nodes.

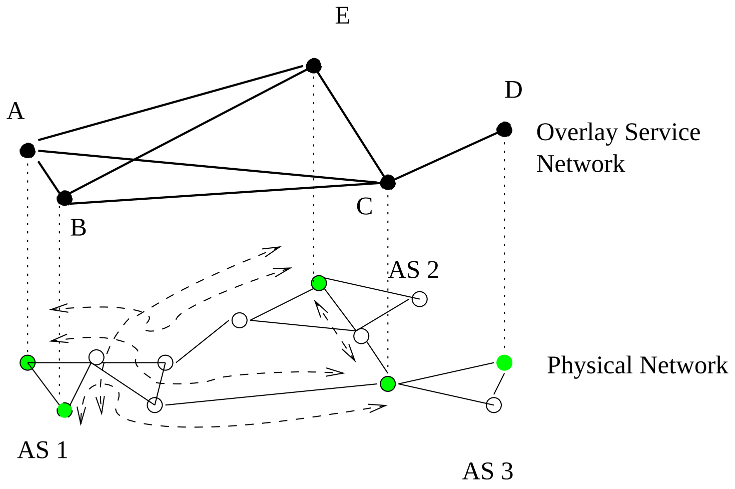

Fig. 5. Adjacent Connection Overlay topology.

5 Two Heuristic Approaches

The overlay topology construction's objective formalized in Section 3 can be restated as : maximize the overlay network connectivity under various situations of IP link failures (or performance degradation) and overlay node degree and overlay path distance constraints. Suppose the IP layer network has W IP links, the total number of failure states (s_i) is 2^W. This ILP problem can be reduced to quadratic assignment problem, which is NP-hard and can not be easily solved for large scale of networks[11].

Based on the problem formulation in Section 3, we can see that the number of b_{ij}^{xy} is highly related to the IP-layer path diversity of all the overlay links. In this section, we are going to propose two heuristic methods to construct efficient overlay topologies. The basic intention is to maximize the connectivity between overlay nodes by considering IP-layer path and topology information.

5.1 Heuristic 1 - Topology-aware K Minimum Joint-Spanning Tree (TKMST)

This method uses K minimal spanning trees connecting all the overlay nodes. However, when considering the disjoint property of two overlay links, we not only consider the overlay layer but also the IP layer path. We define and vary the weight of overlay links based on their IP path information. That is, if two overlay links pass through a common IP-layer link, we also deem them as overlapped and increase the weight of the overlay links based on the number of overlapped IP links. Thus, the resulting K spanning trees have the least overlap with each other at the IP-layer links. Using this method, the overlay network topology can provide each source-destination pair diverse IP layer disjoint overlay paths.

The algorithm is formalized as shown in Algorithm 1.

Algorithm 1

1. Connect all the overlay nodes into a temporary full-mesh topology G_{TO}.

2. Initialize each overlay links' weight in G_{TO} as the number of hops the corresponding IP path passes by.

3. Initialize the resulting overlay topology G_O with 0 overlay link.

4. Initialize the node degree constraint as d_i

5. Find a minimal spanning tree connecting all the overlay nodes based on G_{TO} subjective to each overlay node's degree constraint.

6. For each of the overlay links the spanning tree passes by, increase the overlay link' weight with $h_{xy} * (h - 1)$ (where h is the overlay link's original weight and h_{xy} is the maximal permitted overlay path length) in G_{TO}.

7. For each of the other overlay links of G_{TO}, if it physically shares x hops with the spanning tree, increase the link weight with (x-1)*h.

8. update each node's residual degree constraint by d_i with its increased node degree.

9. Repeat Step 5, 6, 7, and 8 till each overlay nodes meets its degree constraint d_m.

Based on above algorithm description, we can see that for each step, each new minimum spanning tree's links will have least overlap with the existing overlay links. We believe this approach can increase the overlay link path diversity and provide good connectivity under link failure or service degradation. Suppose the average number of hops in the IP paths is h. The computational complexity of the algorithm is $O(V_p^3 + V_o^3 * h^2 * d_m)$, which is a little higher than the above approaches. As our focus is the overlay topologies for overlay service networks, which may again serve large number of overlay applications, we believe the increased complexity is marginal if the service performance can be increased.

5.2 Heuristic 2 - Topology-aware K Random Connection

The goal of overlay topology construction is to provide more feasible IP-layer disjoints paths connecting each pair of overlay nodes. If each overlay nodes' adjacent overlay links are disjoint with each other, there will be high IP-layer diversity in an overlay topology so that the overlay network can provide resilient routing service to each pair of the overlay nodes. Based on this idea, the second heuristic solution follows the steps enumerated in Algorithm 2. The algorithm's computing complexity is $O(V_p^3 + V_o^2 * h^2)$, while h is the average length in terms of number of hops in the IP-layer path between any two nodes in the IP topology.

Comparing to Algorithm 1, we can observe that Algorithm 2 does not require each overlay node to maintain overlay full mesh topology information.

Algorithm 2

```
1. Connect all the overlay nodes into a temporary full-mesh topology $G_{TO}$.
2. Initialize each overlay links' weight in $G_{TO}$ as the number of hops the corresponding IP path passes by.
3. Initialize the resulting overlay topology $G_O$ with 0 overlay link.
4. Initialize the node degree constraint as $d_i$
5. Find a minimum spanning tree under node degree constraint $d_i$ based on $G_{TO}$ connecting all the overlay nodes.
6. Add the overlay links in the minimum spanning tree to $G_O$.
7. update each node's residual degree constraint by $d_i$ with its increased node degree.
8. For each overlay node, $v_o$
9.     For each of its residual outgoing capacity $d_i$
10.        Among all its adjacent overlay links in $G_{TO}$, pick up one with maximal disjoint (in terms of IP layer links) with its existing adjacent overlay links in $G_O$.
```

In contrast, each overlay node only needs to maintain the path performance information to other overlay nodes. As a result, this approach will incur less information exchange overhead and can quickly adapt to the dynamic network performance change.

Both of the above two heuristic methods try to improve overlay link IP-layer path diversity by utilizing IP-layer information. Even though it can incur extra overhead comparing to those IP-layer oblivious approaches, as overlay service networks server large number of overlay applications, we believe that the overhead is marginal comparing the performance gains.

6 Performance Study

6.1 Network Model and Simulation Setup

The simulations are performed on top of two IP-layer topologies: a real ISP topology (an intra-AS topology) and a random topology generated by GT-ITM[2]. The ISP topology we used is a router-level topology with 172 nodes and 763 links published by Rocketfuel [29]. The random topology is Waxman topology[32] generated by GT-ITM[2], which is composed of 200 nodes and connected by 411 links. GT-ITM uses the following approach to generate a sample topology. Network nodes are randomly chosen in a square ($\alpha * \alpha$) grid. An IP-layer link exists between the nodes u and v with the probability $P(u,v) = a * e^{-d(u,v)/(b * \alpha^2)}$, where d(u,v) is geometric distance between u and v, a and b are constants that are less than 1. In the simulation, we take $a = 0.03$ and $\alpha = 200$. All the IP-layer link delays are uniformly set as 2ms. For IP-layer, we assume it always uses the shortest path based routing protocol:

link-state based routing protocol.

For each overlay node, we randomly attach it to one of the physical nodes. During the simulation, we vary the number of overlay nodes and ratio of failure links and obtain simulation results for different performance metrics. In reality, a “link failure” can mean either an IP-layer link is really broken or a link does not meet overlay service users’ QoS requirement, such as delay, bandwidth, etc.

During the simulation, the overlay nodes are connected via one of the following overlay topology construction methods:

(1) Full Mesh (FM).
(2) K Random Connection (KRC).
(3) Mesh-Tree (MT).
(4) K Minimum Spanning Tree (KMST).
(5) Adjacent Connection (AC).
(6) Topology-aware K Minimum Spanning Tree (TKMST).
(7) Topology-aware K Connection (TKC).

For a fair comparison of different methods’ and meet node degree constraints (d_m), we add the following step to MT, KMST and AC’s topology construction algorithm: if an overlay nodes’s degree exceed d_m in the resulting overlay topology, we will only keep the d_m closest neighboring overlay nodes. For comparison, we will also show the simulation results on top of full mesh topologies, which are the best performance an overlay network can provide.

During the simulation, we assume that the overlay networks can detect and recover the IP-layer path failures by using the following mechanism. The overlay routing uses *Link-state based on-demand routing approach* as used in RON. To provide the overlay paths with the least failover time to users, we assume that each overlay node has the global topology information as well as the up-to-date overlay link performance information based on link-state based routing protocol. Thus, the overlay service networks can quickly provide an overlay path connecting to the destination overlay node whenever a fault is detected in an application’s default Internet path. To achieve this, it requires each overlay node to continuously monitor the performance of its adjacent overlay links. In addition, each node also sends the monitoring results to all the other overlay nodes. As a result, the overlay routing overhead not only includes the performance probing traffic but also sending and receiving the link-state message traffic. Suppose an overlay network has n nodes and average node degree is d_m, the overall overlay routing traffic overhead is $n * d_m *$*(number of probing messages) plus* $n * (n-1) *$*(number of link state messages)*. The overhead is directly related to the number of average overlay node degree.

During the simulation, we mainly focus on the following two performance metrics:

(1) **Failure Recovery Ratio**

When an application's default IP-layer path fails or if service performance degrades, resilient routing service overlay should be able to forward the data traffic to the traffic destinations via overlay paths. The failure recovery ratio is an important metric to evaluate the overlay network's service performance. Without further explanation, we assume that the overlay applications (the users of overlay networks) co-exist with overlay nodes and the applications' traffic sources and destinations are restricted to only the overlay nodes. If an IP-layer path connecting two overlay nodes fails, the source overlay node will always try to use the overlay to provide an overlay path connecting to the destination overlay node for the overlay applications. The failure recovery ratio can be defined as follows.

Failure Recovery Ratio

$= \frac{Num.\ of\ recovered\ failure\ paths\ via\ overlay}{Num.\ of\ failed\ IP-layer\ paths\ between\ overlay\ nodes}$

(2) **Recovery Path Hop Penalty (Path Delay Penalty)** During simulation, we assume that the IP-layer routing protocol always takes the shortest paths connecting the source and destination pairs. This means that the recovered overlay paths may will take higher number of IP-layer paths comparing to the default IP-layer paths. Longer IP-layer path could mean that the path has longer latency or consumes additional network resources. In reality, the IP-layer inter-AS (autonomous system) path is determined by each AS's routing policies, which may result in non-shortest path. In this case, the recovered overlay path may be shorter than the corresponding IP-layer paths. We use recovery path hop penalty to quantify an overlay paths' IP-layer distance compared to the original IP-layer path.

Recovery Path Hop Penalty

$= \frac{Num.\ of\ hops\ in\ recovered\ path\ via\ overlay}{Num.\ of\ hops\ in\ the\ corresponding\ failed\ IP-layer\ path}$

During the simulation, we vary the following variables: number of overlay nodes (overlay network size), IP-layer network size, IP-layer link failure ratio, overlay topology construction method. For each simulation scenario setup, we run the simulator for 1000 times and obtain the average value for each performance metric.

6.2 The Effect of Node Degree

We first investigate the different topology construction methods' performance variations with respect to the different values of node degree constraint d_m. We use different topology construction methods to connect 80 randomly selected overlay nodes on top of the ISP topology and an overlay network with 100 overlay nodes on top of the random topology. We fix the IP layer link failure ratio as 0.02, vary the value of degree constraint (d_m) and obtain the average simulation results for each performance metrics.

The Path Failure Recovery Ratio for the candidate overlay topologies is shown in Figure 6. From the figure, we can observe that the performance comparison among these methods has the same trend on top of the two IP-layer topologies (ISP and random). TKMST and TKC perform better than the rest topology construction methods. MT (Mesh Tree) performs the worst among all the candidate methods. AC (Adjacent connection) method performs worse than KRC (K Random Connection) under degree constraint. TKMST performs better than TKC, both of which can provide similar performance as the full mesh topology (without node degree constraint). When varying the value of node degree constraint (d_m), we observe that with the increase in d_m, the path failure recovery performance increases in the beginning and the performance gain dimmishes after the node degree is above certain threshold. Based on these results, we can conclude that careful design of the overlay network topology can greatly decrease the overlay network monitoring overhead without degrading the service performance.

Figure 7 shows the Recovery Path Hop Penalty for different overlay topology construction methods with the variance of d_m. From the results, we can observe that the TKMST has least penalty cost comparing to KMST, AC, TKC, KR and MT while MT (mesh tree) method has the maximum path penalty cost. The full mesh topologies' average Recovery Path Hop Penalty is around 1.2, which is less than the other approaches. An obvious trend we observe is that the recovery path hop penalty decreases with the increase in the node degree. With fixed node degree of 8, the cumulative distribution function (CDF) of path recovery penalty distribution is shown as in Figure 8. In Figure 8, the Y-axis is the CDF number of recovery paths in terms of path penalty. From the figure, we observe that the different methods' performance is consistent under different path penalty constraint (or overlay path length constraint, h_{xy}). The results show that most paths' recovery path penalty is less than 1.5. In addition, TKMST, KMST and TKC can provide most recovery paths with less path penalty than other methods while providing higher failure recovery ratio as discussed above.

6.3 The Effect of Overlay Network Size

In this section, we investigate whether the results discussed so far are consistent with respect to the overlay network size. With fixed node degree constraint of 6 and IP layer link failure ratio is 0.02, we vary the sizes of overlay networks

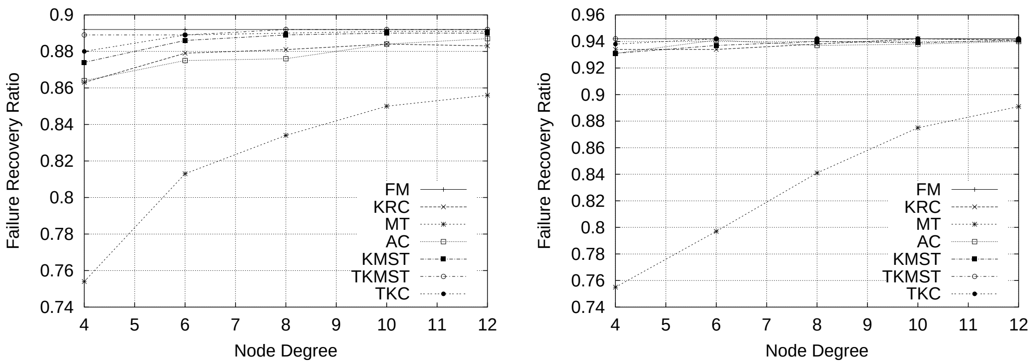

Fig. 6. Node Degree vs. Failure Recovery Ratio (Real ISP Topology & Random Topology)

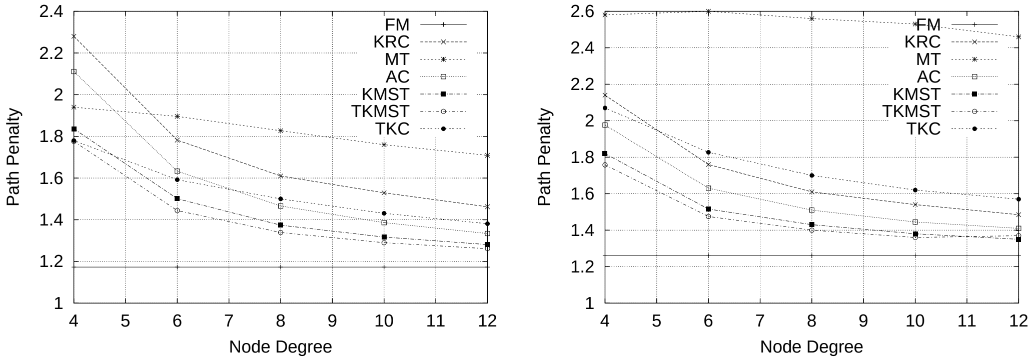

Fig. 7. Node Degree vs. Path Recovery Penalty (Real ISP Topology & Random Topology)

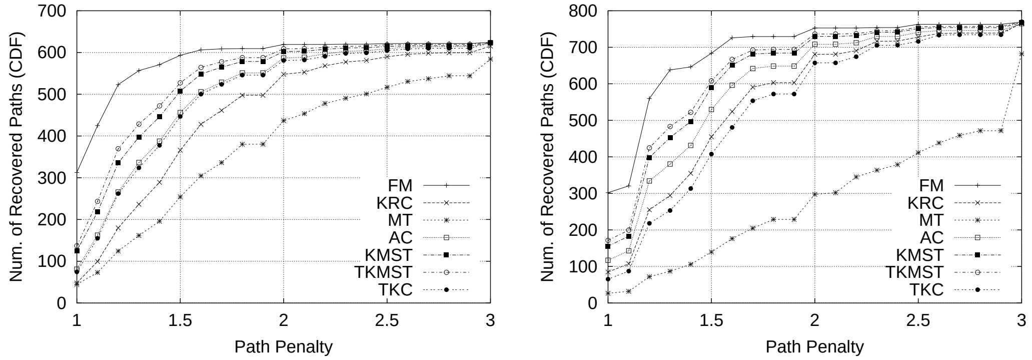

Fig. 8. Distribution of Path Recovery Penalty (Real ISP Topology & Random Topology)

and obtain the simulation results as shown in Figure 9 and Figure 10. From the figures, we observe that the performance comparison among the candidate topologies remains the same as described in previous sections. In addition, the recovery path penalty drops with the increase in the overlay network size while the failure recovery ratio more or less remains stable. This is because when the number of nodes increases, more IP-layer paths connecting these nodes will be affected during failure occurs. As a result, because we restrict the overlay applications' source and destination nodes to the overlay nodes, the increase in the overlay network size cannot increase failure recovery ratio very much. However, larger size overlay network can benefit more overlay service customers, as shown in Figure 11. In this figure, the source and destination overlay nodes of the overlay applications can be any node in the IP topology as comparing to only the overlay nodes as the setup for other simulation results. As a result, the failure recovery ratio is all the IP-layer nodes' failure recovery ratio (not only the overlay nodes').

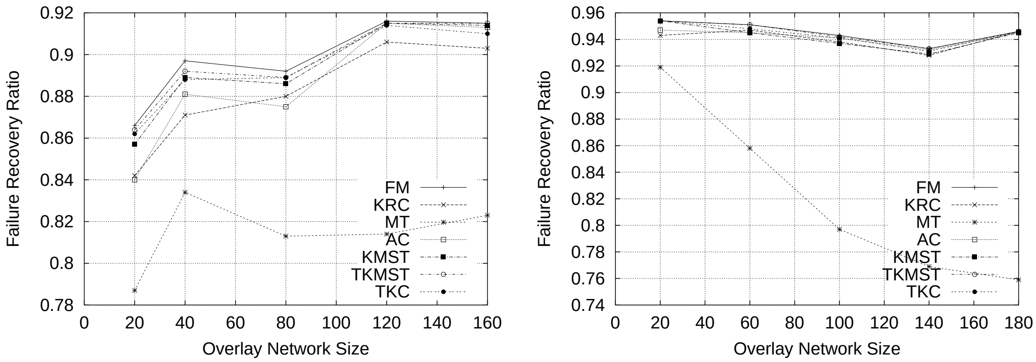

Fig. 9. Overlay Network Size vs. Failure Recovery Ratio (Real ISP Topology & Random Topology)

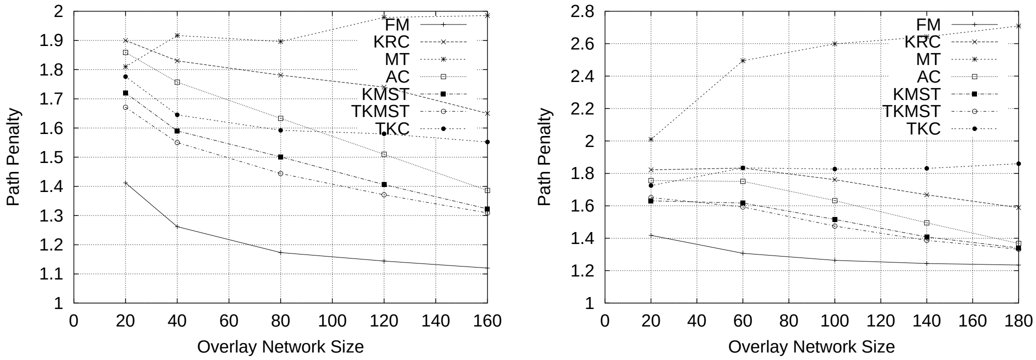

Fig. 10. Overlay Network Size vs. Path Recovery Penalty (Real ISP Topology & Random Topology)

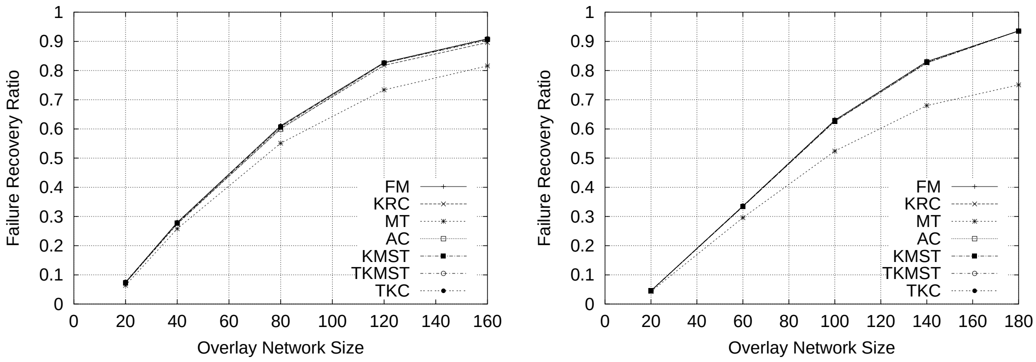

Fig. 11. Overlay Network Size vs. Overall Failure Recovery Ratio (Real ISP Topology & Random Topology)

6.4 The Effect IP-Layer Link Failure Ratio

In this section, we will focus on IP-layer link failure ratios' impact on the performance comparison among all the candidate overlay topologies. For the ISP topology, we fix all the overlay topologies with 80 nodes and degree constraint (d_m) as 6. For the random topology, all the overlay topologies have 100 nodes with degree constraint (d_m) as 6. The simulation results are shown in Figure 12 and Figure 13. The results prove that the performance comparison is not changed with the increase in the IP-layer link failure ratio. When IP link failure ratio increases, failure recovery ratios gradually decrease and recovery path hop penalties increase for all the candidate topologies. Moreover, performance gaps between different methods become larger as we increase the IP link failure ratio. We also observe from the figure that MT's failure recovery ratio drops much faster than other approaches. This is due to the topological characteristics of mesh tree topology construction method. As the mesh links are set up between uncle-nephew or grandfather-grandson nodes, these links will more or less share the same group of IP-layer "risk" links with the minimum spanning tree. As a result, when the number of failure links are increased, there will be higher chance that the overlay topologies will also loss connection and can not provide desirable failure recovery performance.

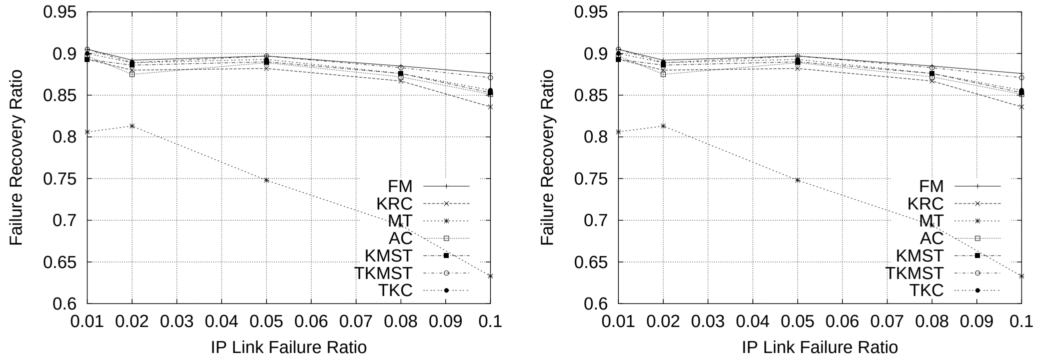

Fig. 12. IP Link Failure Ratio vs. Failure Recovery Ratio (Real ISP Topology & Random Topology)

6.5 Effect of Physical Network Size

To show that the stability of the performance comparisons with respect to different sizes of IP layer topologies, we pick two other random topologies with size 100 and 500 which are generated by GT-ITM using the same method as discussed above. For the first experiment, we fix the degree constraint (d_m) as 4 and 6, the overlay size as 50, and the IP-layer link failure ratio as 0.05. For the second case, we fix the overlay network size as half size of the corresponding IP layer topology. The failure recovery ratio for the different simulation

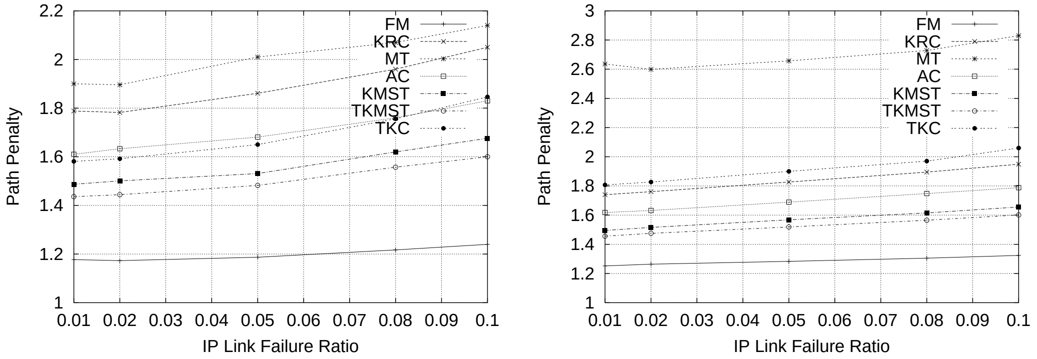

Fig. 13. IP Link Failure Ratio vs. Path Recovery Penalty (Real ISP Topology & Random Topology)

setups is shown in Figure 14 and Figure 15. The results prove that the previous performance comparisons among different methods are consistent with different sizes of IP layer topology and the different ratios between overlay network size and IP Network. That is, TKMST and TKC show similar performance as Full Mesh (FM) and better than the rest candidate overlay topologies while MT performs the worst among all the candidate overlay topologies based on the simulation setup.

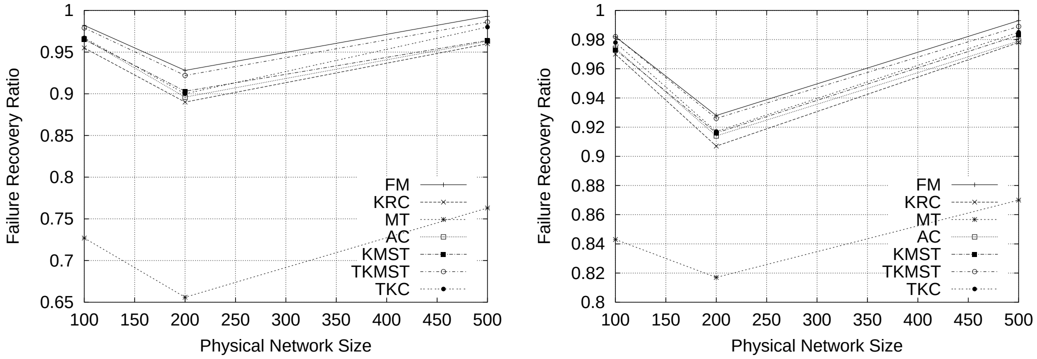

Fig. 14. Physical Network Size vs. Failure Recovery Ratio (Random Topology d_m=4 & d_m=6))

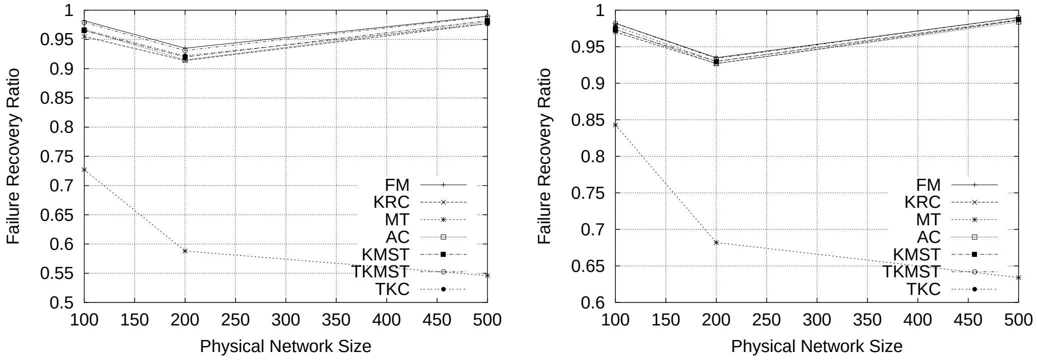

Fig. 15. Physical Network Size vs. Failure Recovery Ratio (Random Topology d_m=4 & d_m=6))

6.6 Summary of Results

From the above simulation results, we can draw the following conclusions.

(1) **The performance of overlay routing service highly depends on the construction of overlay topologies.** It is very important to carefully select an overlay topology when designing an overlay network. The two proposed new approaches (TKMST and TKC) perform better than existing candidate topologies in terms of failure recovery ratio and recovery path hop penalty. The performance comparison among all the candidate topologies scales to physical network size, overlay network size, and IP layer link failure ratio.

(2) **IP-layer topology aware based overlay topology can achieve better performance in providing resilient routing service.** Among the simulated topologies, adjacent connection (AC), topology-aware K spanning tree (TKMST) and topology-aware K-connection (TKC) are IP-layer connectivity-aware overlay topology construction methods. These approaches consider the IP-layer topology (or path information) when constructing the overlay topology. The resulting overlay topologies can always provide us with similar performance as that of full mesh and higher failure recovery ratio compared to the other methods. At the same time, they incur moderate path recovery penalty and routing overhead.

(3) **Mesh tree is not a good choice as an overlay service topology.** Even though the mesh-tree topology can provide us good resilience service when considering overlay multicast protocols, its performance in general overlay service network (providing generic routing services to other applications) is not good. For example, it provides the lowest average failure recovery ratio among all the candidate approaches even though it incurs similar routing traffic overhead (same node degree constraint) as the other approaches.

7 Related Work

There has been a moderate amount of work in the area of overlay networks. The effort on application-specific overlay networks has targeted on widely usable applications such as multicasting[13,34,19,12,8,28] and peer-to-peer file sharing[30][27][1]. Several other work has been dedicated to proposing a general overlay service networks that can be used to provide value-added service for a variety of application-layer services, such as QRON[21], SON[15], Opus[10], YOID[17], OverQoS[31]. Another research effort is the Planet-lab[3] whose goal is to build a global testbed for developing and accessing new network services. It not only utilizes overlay network technique to provide service but also provides overlay networks with a desirable test platform. X-Bone[5] is a system for the automated deployment of overlay networks. It operates at the IP-layer and bases on IP tunnel technique. The main focus is to manage and allocate overlay links and router resource to different overlays and avoid resource contention among the overlays.

Resilient Overlay Network (RON)[6] is closely related to this paper. It is proposed to quickly detect and recover path outages and degraded performance. RON can better cope with the inter-domain path re-convergence problem than border gateway protocol (BGP), which usually takes longer time to converge to a new valid route. A similar work was proposed in [35], which is based on, a prefix based routing approach, Tapestry[36]. It provides resilient overlay routing services by dynamically switching traffic to pre-computed alternate routes. In addition, the messages can be duplicated and multicasted around the network congestion and failure hotpots with rapid re-convergence to drop duplicates. Reference [25] and [18] deal with dynamic topology construction to adapt to network dynamics.

The inter-domain topology's impact on routing performance and the process of delayed Internet routing process is examined in [20]. In [26], the authors characterize the real and generated physical topologies. Their focus is the difference between generated topologies and real Internet topologies. The physical topologies' impact on four different multicast design issues were also studied in this paper. In [24], the authors use a game-theoretic approach to investigate the performance of selfish-based routing (overlay routing or source based routing) in Internet-like environments.

8 Conclusions

In this paper, we formalized the overlay topology construction problem and investigated the effect of overlay topology on the performance of overlay routing service. The contribution of this paper is two-folds. First, we conducted extensive simulation studies of different overlay topologies' performance in terms of failure recovery ratio and recovery path hop penalty. The results show that the performance of the candidate overlay topologies differ a lot, and it is necessary to carefully design overlay topologies when constructing overlay networks. Second, we proposed two new overlay topology construction methods: Topology-aware K Minimum Spanning-Tree (TKMST) and Topology-aware K Connection (TKC). The simulation results show that these two methods outperformed other existing approaches. In addition, it is observed that the underlying IP-layer network information can benefit a lot in constructing efficient overlay topologies.

For the future work, we will focus on investigating how we can optimally select overlay links with best performance in addition to "topology-aware" approach. Another part of future work is to conduct large scale simulation to verify the different overlay topologies' performance on top of the real Internet inter-AS topology.

References

[1] Gnutella[online]. http://www.Gnutella.com.

[2] "GT-ITM: Modeling Topology of Large Internetworks,"[online]. http://www.cc.gatech.edu/projects/gtitm/.

[3] Planet Lab[online]. http://www.planet-lab.org.

[4] Qbone[online]. http://qbone.internet2.edu/.

[5] Xbone[online]. http://www.isi.edu/xbone.

[6] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, R. Morris, "Resilient Overlay Network," In Proc. ACM SOSP'01, pp.131-185, Oct. 2001.

[7] J. Apostolopoulos, T. Wong, W. Tan and S. Wee, "On Multiple Description Streaming with Content Delivery Networks", In Proc. IEEE INFOCOM'02, June 2002.

[8] S. Banerjee, B.Bhattacharjee, C.Kommareddy, "Scalable Application Layer Multicast, " in Proc. of ACM Sigcomm 2002, Pittsburgh, Pennsylvania, August 2002.

[9] S.Banerjee, S. Jee, B.Bhattacharjee, C.Kommareddy, "Resilient Multicast using Overlays", in Proc. of ACM Sigmetrics 2003, San Diego, CA, June 2003.

[10] R. Braynard, D. Kostic, A. Rodriguez, J. Chase, and A. Vahdat, "Opus: An Overlay Peer Utility Service," In Proc. IEEE OpenArch'02, June 2002.

[11] E.Cela, "The Quadratic Assignment Problem: Theory and Algorithms", Kluwer Academic Publishers, 1998.

[12] Y. Chawathe, S. Mccanne, E. A. Brewer, "RMX: Reliable Multicast for Heterogeneous Networks," In Proc. Infocom'00, pp.795-804, March 2000.

[13] Y. Chu, S. G. Rao, H. Zhang, "A Case for End System Multicast," In Proc. ACM SIGMETRICS 2000, pp.1-12.

[14] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms," McGraw-Hill Book Company 1990.

[15] Z. Duan, Z. Zhang, and Y. T. Hou, "Bandwidth Provisioning for Service Overlay Networks," In Proc. SPIE ITCOM Scalability and Traffic Control in IP Networks (II) '02, July 2002.

[16] H. Eriksson, “MBone: The Multicast BackBone,” Communications of ACM, vol.37, no. 8, pp.54-60, Aug. 1994.

[17] P. Francis, “Yoid: Extending the Internet Multicast Architecture [online],” http://www.aciri.org/yoid/docs/index.htm.

[18] J. Han, D. Watson and F. Jahanian, “Topology Aware Overlay Networks“, IEEE INFOCOM 2005.

[19] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O’Toole, Jr. “Overcast: Reliable Multicasting with an Overlay Network, ” In Proc. 4th USENIX OSDI, Oct. 2000.

[20] C. Labovitz and A. Ahuja, “The Impact of Internet Policy and Topology on Delayed Routing Convergence,” In Proc. IEEE INFOCOM 2001.

[21] Z. Li and P. Mohapatra, “QRON: QoS-aware Routing in Overlay Networks,” in Proc. of IEEE Journal on Selected Areas in Communications (JSAC) special issue on Service Overlay Networks, Jan. 2004.

[22] Z. Li and P. Mohapatra, “HostCast: A New Overlay Multicasting Protocol,” In Proc. IEEE Int. Communications Conference (ICC) 2003.

[23] A. Nakao, L. Peterson, A. Bavier, “A Routing Underlay for Overlay Networks,” to appear in Proc. ACM SIGCOMM 2003.

[24] L.Qiu, R.Y.Yang, Y.Zhang and S. Shenker, “On Selfish Routing in Internet-Like Environments,” in Proc. ACM SIGCOMM 2003.

[25] Z. Ma, H. Shao and C. Shen, “A New Multi-path Selection Scheme for Video Streaming on Overlay Networks“, in Proc. IEEE ICC 2004.

[26] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, D. Estrin, “On characterizing network topologies and analyzing their impact on protocol design,” USC-CS-TR-00-731, March 2000.

[27] S. Ratnasamy, P. Francis, M. Handley, R. Karp and Scott Shenker, “A Scalable Content Addressable Network,” In Proc. of ACM SIGCOMM 2001, Aug.2001.

[28] S. Y. Shi, and J.S.Turner,“Routing in Overlay Multicast Networks,” In Proc. IEEE Infocom’02, June 2002.

[29] N. Spring, R. Mahajan, and D. Wetherall, “ Measuring ISP topologies with Rocketfuel,” In Proc. ACM SIGCOMM’02, Aug. 2002.

[30] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications,” In Proc. ACM SIGCOMM 2001, August 2001, pp. 149-160.

[31] L. Subramanian, I. Stoica, H. Balakrishnan and R.H.Katz, “OverQoS: Offering Internet QoS using Overlays”, In Proc. HotNet-I Workshop, October 2002.

[32] B. M. Waxman, “Routing of Multipoint Connections,” IEEE Jornal on Selected Areas in Communications”, Dec. 1988.

[33] A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson and R. Y. Wang, “Overlay Mesh Construction Using Interleaved Spanning Trees,” In Proc. IEEE Infocom’04, March 2004.

[34] B. Zhang, S. Jamin, L. Zhang,“Host Multicast: A Framework for delivering Multicast to End Users,” In Proc. Infocom’02, June 2002.

[35] B. Y. Zhao, L .Huang, A. D. Joseph,and J. D. Kubiaotowicz, “Exploiting Routing Redundancy via Structured Peer-to-Peer Overlays” in Proc. of the 11th IEEE International Conference on Network Protocols (ICNP’03).

[36] B. Y. Zhao, L .Huang, A. D. Joseph,and J. D. Kubiaotowicz, “Tapestry: A Resilient Global-scale Overlay for Service Deployment,” IEEE Journal on Selected Areas in Communications (JSAC), January 2004, .