Resource Reservation Comparison of Fault Resilient Routing Schemes

Yigal Bejerano and Pramod V. Koppol
Bell Laboratories, Alcatel-Lucent, NJ, USA.

Abstract—**Reliable delivery of network services is critically for numerous real-time applications, such as video conferencing, broadcast TV and content distribution, in which time sensitive content is delivered to a single or multiple destinations. Failure protection in connection-oriented networks can be realized using *local* protection schemes, such as *fast reroute* (FRR), or using *end-to-end* protection schemes, like *redundant trees*. In this paper we study the trade-offs between local and end-to-end protection schemes from several key aspects such as protection availability, management complexity and in particular the resource reservation efficiency of the two approaches.**

Keywords: Multicast, Fast-Reroute, Redundant Trees, Survivable Network Design, Fault Resiliency

I. INTRODUCTION

Reliable delivery of network services is critically dependent on the existence of a fault resilient network that can rapidly restore services in the event of a failure. Network reliability is in particular essential for real-time applications, such as multi-party video conferencing, broadcast TV, content distribution and distance learning classrooms, in which time sensitive content is delivered to multiple destinations by using the network multicast services.

Failure protection in emerging connection-oriented networks can be realized using *local* protection schemes such as MPLS fast reroute (FRR) [1]–[5], or using *end-to-end* protection schemes [6]–[15]. These mechanisms are based on prior provision of alternative paths with guaranteed bandwidth in addition to the primary paths. In both options, sufficient network resources should be reserved for the alternative paths to ensure the bandwidth availability when needed.

This paper studies the trade-offs between local and end-to-end protection schemes from several practical perspectives, with primary focus on resource reservation. To the best of our knowledge, we are the first that compare the intrinsic network resources overhead of these fundamentally different protection solutions. Our study is applicable for both unicast and multicast services, however, we concentrate our discussion on multicast services and consider unicast connections as a special case of multicast connections between two nodes.

A. Local Protection

Given a multicast connection request the network establishes a primary point-to-multipoint (P2MP) tree with the source node and destination nodes being the root and leaves, respectively. In a local protection scheme, such as *fast-reroute* (FRR), a node on a primary path uses an alternate pre-established path known as a *detour* to route traffic from the primary path around the failed link or node. Separate detours are set up to deal with different failures. The main advantage of the local restoration schemes is their short recovery time. However, it has been observed in [16] that local protection schemes can use more bandwidth than end-to-end schemes due to the need to establish multiple detours. Also, for multicast connections, as reported in [17], the bandwidth usage issue gets exacerbated due to traffic overlap that occurs when the same packet has to be sent multiple times on a given link in the same direction. We refer to such traffic overlap as *packet duplication*. Several studies address the bandwidth usage and the packet duplication shortcomings of local restoration. In [17] the authors propose a FRR-based local restoration scheme that eliminates packet duplication by revising the routing mechanism. However, as a result both the primary P2MP tree and the associated detours become considerably longer compared to the shortest possible paths. This may result in high bandwidth usage. Other studies typically propose making efficient use of network resources through sharing of detour bandwidth wherever possible [1], [5]. However, such solutions introduce additional control and management plane complexity.

B. End-to-End Protection

In such scheme, two link or node-disjoint paths (or trees for P2MP connections) are set up between the source and each destination. One of these paths serves as a *primary*, while the other serves as a *standby*. End-to-End protection allows two modes of operation; In *hot standby*, also known as $1 + 1$ protection, the alternative paths carry the protected content. Consequently, a destination node can immediately switch to the backup traffic once a failure is detected. Instead, in *cold standby*, also referred to as $1 : 1$ protection, as soon as the source learns of a failure, it activates the standby paths. For multicast connections, end-to-end protection with two or more trees has been proposed in various studies, [6]–[15]. Among the various solutions, the *redundant trees* (RTs) approach is probably the most common one [8]–[15]. This scheme constructs two trees rooted at the source node that provide two node disjoint paths from the source to every destination node. As a proper representative of his approach we use the scheme described in [15], this scheme maintains near optimal RTs even when new destinations are added to the multicast connection, while other solutions cannot guarantee to maintain valid RTs in the case of dynamic multicast connections.

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

Recently, there has been significant ongoing activity relating to the use of protection modes in MPLS-like transport networks [4], [18], [19]. This includes end-to-end protection and also effective failure detection mechanisms, such as Bi-directional Forwarding Detection (BFD) [20], which substantially reduce the failure detection and notification time. This fact coupled with bandwidth related issues, e.g., packet duplication, and the high management plane complexity related to local protection schemes, makes end-to-end schemes increasingly appealing.

C. *Our Contribution*

This study compares the performance of local *fast-reroute* (FRR) restoration schemes against *redundant-trees* (RTs) based end-to-end protection on several key aspects, which are typically ignored in other studies. Given the rich literature of protection and restoration solutions, this study does not aim to provide a comprehensive survey of the field, but to highlight the fundamental differences between the two approaches. To this end, we consider only a few state-of-the-art alternatives of each approach.

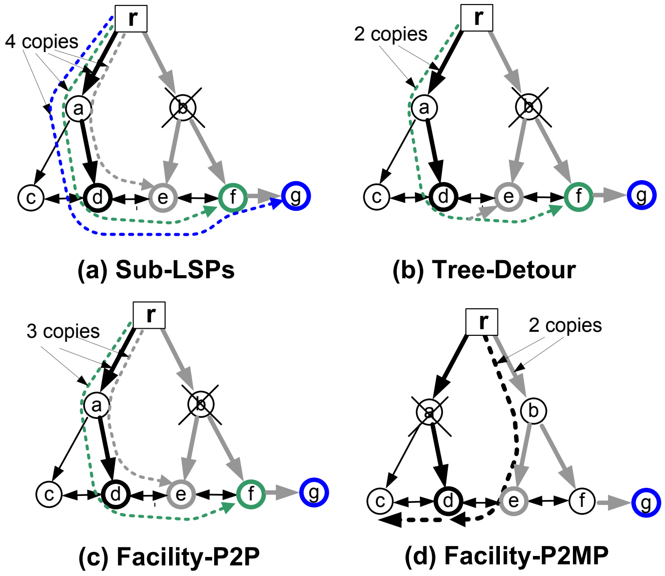
We first provide a short description of the two protection approaches; We described four commonly used FRR alternatives as well as two resource sharing options, which were designed to reduce the resource consummation of FRR. We also introduce the link-coloring scheme in [15], which enables dynamic management of redundant trees. Then we compare the performance of the two approaches. Due to space limitation, we consider only the following few key criteria;

- *Protection Availability*- We examine if each one of the protection approaches can provide protection for any potential failures. We model the network as a directed graph[1] with the only requirement of having two node disjoint paths from the source to each destination, recall that this is prerequisite for providing protection by any protection solution. We show that in such model, FRR may not provide protection against all potential failure, while redundant trees can always be found.
- *Resource reservation* - We evaluate the network resource requirements of the two approaches. Our extensive simulations show that even when aggressive resource sharing method is used fast reroute detours requires significant higher amount of network resources than redundant-trees.
- *Management Complexity* - We discuss the management involvement aspects of the two methods.

We conclude by summarizing the pros and cons of each one of the restoration approaches.

II. NETWORK MODEL

This section describes the network model, which we use for presenting the different protection schemes. We consider a connection-oriented network, such as Multiprotocol Label Switching (MPLS), modeled as a *directed graph* $G(V, E)$, where each node $v \in V$ is a router/switch and E is the set of the directed links between them. A link from node u to node v is denoted by a directed edge $(u, v) \in E$, where node u is termed an *incoming neighbor* of node v, while node v is called an *outgoing neighbor* of node u. Every directed edge $e = (u, v) \in E$ is associated with a positive weight denoted by w_e, which indicates the cost of allocating one unit of bandwidth on this link for the given direction. We allow the weight of (u, v) to be different from that of (v, u).



(a) Sub-LSPs **(b) Tree-Detour**

(c) Facility-P2P **(d) Facility-P2MP**

Fig. 1. Variants of Fast Reroute

Each multicast (or unicast) connection request is characterized by a source node $r \in V$ and a set $D \subseteq V - \{v\}$ of destination nodes. We assume that the required bandwidth for each connection is one unit of bandwidth. The following definitions consider the network as seen from the source $r \in V$. We say that the network is $2 - reachable$ if there are two node disjoint paths from r to very destination in D. Observe that having 2-reachable network is a prerequisite for providing protection against any possible failure. Otherwise, the network contains cut nodes (or links) that their removal disconnect the source from some of the destinations. Throughout the study we assume that the considered network is 2-reachable for the given source node r and a set D of destinations.

III. LOCAL PROTECTION BASED ON FAST REROUTE

A. *Variants of Fast Reroute*

In the case of a local protection scheme, such as *fast-reroute* (FRR), when a multicast connection request arrives, a primary point-to-multipoint (P2MP) tree is provisioned with the source and destination nodes being the root and leaves, respectively. In addition to the primary P2MP tree, the network sets alternative pre-established paths known as a *detours* to route traffic from the primary path around any failed link or node.

---

[1] This is an appropriate network representation which supports various practical scenarios, e.g., where link capacity is only available in one direction but not the other, or where a different cost metric is used for each direction.

Although various variants of fast-reroute have been proposed, we describe here only four common ones. In all the variants, in the event of a failure of any primary-tree link, say link (u, v), (may be due to failure of v), node u, referred to as the *point of local repair* (PLR), switches the traffic to one or more pre-determined detours that bypass the failed link or node. The four variants are different in several aspects;

- *Point-to-point* (P2P) or *point-to-multipoint* (P2MP) detours.
- The *merge points* (MPs) where the detours end.
- *Dedicated detours* or *facility detours*.

Unlike dedicated detours, facility detours are shared among different connections. However, also in this option, each facility detour has a specific PLR and it protects against a specific failure.

The four variants are presented in Figure 1 for a multicast connection with source node r and 4 destination nodes d, e, f and g. For all the detours, we assume shortest path routing. Each sub-figure considers a single FRR variant as well as a failure of either node a or b, and it illustrates the corresponding detour(s) used by this variant for protecting the failed node. The variants are;

- **Sub-label-switching-paths (LSPs) [5]**: This variant uses multiple P2P detours. Each detour starts at the PLR and the merge point is one of the affected destination.
- **Tree-detour [5]**: Each link and node in the primary tree is protected by a P2MP detour that is rooted at the PLR and connects to the immediate downstream nodes of the node being protected.
- **Facility-P2P [3], [4]**: A facility protection approach, which uses multiple P2P detours from the PLR to the immediate downstream neighbors of the node being protected.
- **Facility-P2MP**: A facility protection approach that provides node protection by using a P2MP facility detour from the PLR to the immediate downstream neighbors of the node being protected. Note that in this case, the PLR connects to every outgoing neighbor of the node being protected, which may results with unnecessary resource reservation.

The main advantage of the local restoration schemes is their short recovery time. However, it has been observed in [16] that local protection schemes can use more bandwidth than end-to-end schemes due to the need to establish multiple detours. Also, for multicast connections, as reported in [17], the bandwidth usage issue gets exacerbated due to traffic overlap that occurs when the same packet has to be sent multiple times on a given link in the same direction. We refer to such traffic overlap as *packet duplication*. Notice that all the variants, shown in Figure 1, suffer from packet duplication.

Several studies address the bandwidth usage and the packet duplication shortcomings of local restoration. These studies typically propose making efficient use of network resources through sharing of detour bandwidth wherever possible [1], [5]. In this study, we consider two types of *resource sharing* that may be applied with all the FRR variants;

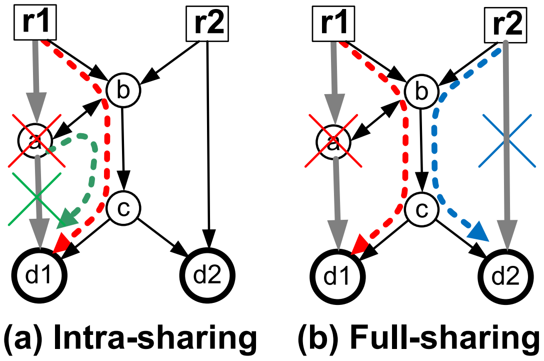
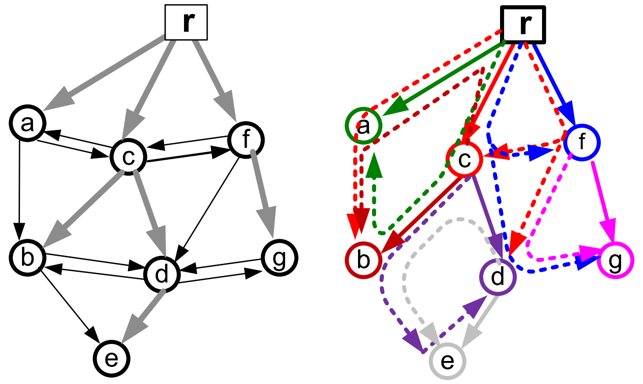


Fig. 2. Variants of resource sharing

- **Intra-sharing [5]** – Detours of *a given connection* may share bandwidth resources if they are not active at the same time, but detours of two different connections do not share bandwidth. Figure 2-(a) shows a scenario where intra-sharing reduce the resource reservation. In this example, two detours marked with red and greed dashed lines, protect the connection from r_1 to d_1 and they both use the link (b, c). Since they protect the primary path against different failures the two detours are not active at the same time, so they may share the restoration resources allocated on the link (b, c).
- **Full sharing [1], [5]** - Detours protecting a node or a link may share their bandwidth resources with *any* other detours that protect against failures of other nodes or links, regardless of the connections associated with the detours. Figure 2-(b) illustrate the advantages of full sharing. In this example, two detours, marked with red and blue dashed lines, of two different connections may share the link (b, c). Since they protect from different failures, these detours are not active simultaneously (assuming the network provides a protection only against a single failure at a time). Recall from Figure 2-(a) that also the green detour uses the link (b, c). Thus, only a single bandwidth unit is needed and it can be shared between the three detours. This approach provides the best possible sharing (no other sharing scheme can do better using shortest path detours). Thus we consider the full sharing option as a lower bound for the resource consumption required by each FRR variant.

B. Management Complexity of Fast Reroute Protection

As mentioned above, each connection is associated with several detours. The number of required detours is at least as the number of links along the primary tree, when assuming tree detours. In such settings, each detour protects against the failure of a single link (u, v), which may result from the failure of node v, where node u is the PLR. As an example, Figure 3 illustrates all the tree detours that are required for protecting all the links and nodes of the multicast connection shown in Figure 3-(a). Figure 3-(b) shows the detours, where each



(a) The input graph and primary tree. **(b) All Tree-Detours for protecting the primary tree.**

Fig. 3. An example of all the tree detours of a single multicast connection.

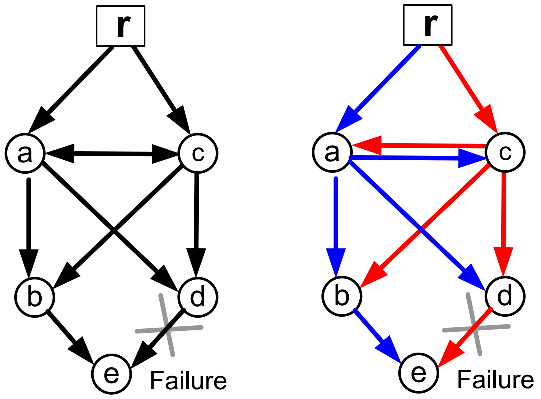
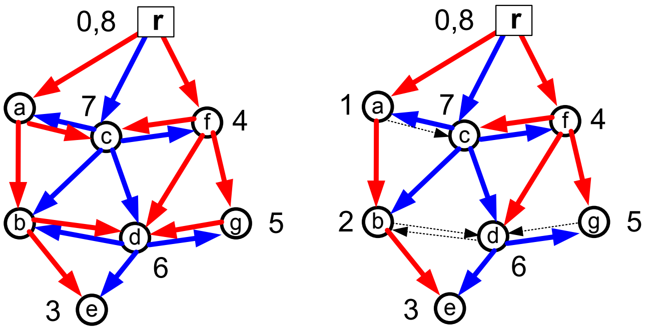


Fig. 4. An example where FRR cannot provide maximal possible protection.



(a) The red and blue RDAGs **(b) Two Redundant Trees**

Fig. 5. A directed graph with a pair of redundant trees.

primary link and its protected end-point has a unique color and they are marked with solid line, while the corresponding tree detour is marked with dotted line with the same color. For instance, the primary link (r, a) is colored green and its detour is the path r, c, b, a. Since a is a leaf node of the primary tree it is not protected by a detour. Now consider the primary link (r, c) and node c which are colored red. These components are protected by a single detour. Since c is both a detonation and branch node of the primary tree, the corresponding (red) detour ends at three merge-points; nodes b, c and d, and it contains the links $(r, a), (a, b)(r, f)(f, c)$ and (f, d). This scenario illustrates the bandwidth requirements for protecting just a single link or node. Figure 3-(b) also visualizes the packet duplication issue. For instance, the link (c, b) is included in the primary tree and it is used by three different detours. Thus, unless resource sharing is used, 4 units of bandwidth are required to be allocated on this link.

The requirement to provision so many detours increases the management complexity of the network. Consequently, facility detours are essential for reducing the overall number of required detours. They are instrumental for simplifying the provisioning and management of the detours. However, as we show later, they may increase the resource consumption.

C. *Availability of Protection Detours*

We now check the availability of FRR protection in any circumstances, in which the network is 2 reachable (i.e., the network contains two node disjoint path from r to every destination in D). As illustrated in Figure 4-(a), there are 2-reachable directed graphs, where FRR cannot provide restoration for some links or nodes. Consider any path from r to node e in the graph depicted in Figure 4-(a). Regardless of the selected path, it must traverse either node b (or d). Since node b (and d) has a single outgoing link (b, e) (or (d, e)), there is no detour that can start at node b (or d) and protect the link (b, e) (or (d, e)). Notice that this is an inherent limitation of FRR regardless of the used FRR variant. While Figure 4-(b) shows that there are two RTs rooted at node r and provide protection against any link or node failure.

IV. REDUNDANT-TREES BASED PROTECTION

A. *Redundant-Trees*

Consider a 2-reachable network $G(V, E)$, a *source node*, r, and a set D of destinations. *Redundant trees* (RTs) based protection is achieved by constructing two trees rooted at r that induce two node disjoint paths from r to each destination in D. Consequently, the source remains connected to all the destinations in the event of single link or node failure. An example of an RT-pair is illustrated in Figure 5-(b) for the network shown in Figure 3-(a). An RT-pair is considered *optimal* if its total cost is *minimal*, The problem of finding optimal RTs is known to be NP-hard when only some of the nodes are destinations [14], however, some algorithms produce near optimal results [14], [15].

B. *The Link-Coloring Scheme*

We consider in this study the RTs calculation algorithm based on the *link-coloring* scheme presented in [15]. This scheme has several advantages over the other alternatives. Unlike other solutions, this algorithm enables dynamic addition of destinations to existing multicast connections while ensuring two node disjoint paths from the root to each destination, the algorithm ensures maximal protection also when the network contains cut nodes or cut links, it graceful deals with topology changes and it produces near optimal RTs.

The scheme is based on a preliminary graph-partitioning of the network. Consider a network modeled as a directed graph $G(V, E)$ and a source r. The scheme performs link coloring that logically partitions the network into two *redundant directed acyclic sub-graphs* (RDAGs), referred to as the *red and blue RDAGs*. The two RDAGs share all the nodes but have disjoint sets of directional links, and each RDAG contains a path from r to any other node. The RDAGs are constructed in such a manner that preserves the following property:

Property 1: For any given destination node, say d, any path from r to d in the blue RDAG is node disjoint from any path from r to d along the red RDAG.

Consequently, for any multicast connection request, red and blue trees can be independently provisioned using any tree selection method at each of the two RDAGs and together induce low cost redundant trees. An example of such two RDAGs is given in Figure 5-(a). For the sake of completeness, we provide a brief description of the link-coloring algorithm in [15], which contains two steps;

Node Arrangement - First, the nodes are arranged in a list, denoted by L, such that each non-source node is included only once in L and it has incoming edges from neighbors both before and after it in L. Only r is included twice in L as the first and last nodes. The algorithm starts with an empty list L and it iteratively inserts all the nodes into L. We refer to any node in L as *marked*, otherwise it is termed *unmarked*. Initially, the algorithm inserts r twice into L as the first and last nodes. It also inserts all the outgoing neighbors of r to L between the two instances of r, in any arbitrary order. After the initialization stage, the algorithm iteratively finds an unmarked node, say u, with two marked incoming neighbors. Then, it inserts u to L between these two neighbors. The algorithm terminates when all the nodes are included in L (marked) and they are numbered according to their locations in L. Note that during the iterative process the algorithm may not find such unmarked node u with two marked incoming neighbors. These are challenging situations and they require special treatment, as described in [15].

Link Coloring - After calculating L, the algorithm colors the links according to their orientation. Consider any link $(u, v) \in E$. If u appears before v in L, (u, v) is termed a *forward link* and it is colored red. Otherwise, it is called a *backward link* and it is colored blue. Since r is placed as the first and last node in L, a special treatment is given to its outgoing links. Consider an outgoing link (r, v) from r to one of its neighbors v. Since G contains two-node disjoint path from r to any other node, node v must have at least one more incoming link (u, v) from a non-source node, say u. If the link (u, v) is a red forward link than the links (r, v) is considered as backward link and it is colored blue. Otherwise the link (r, v) is considered as forward link and it is colored red. In the case that node v already have both red and blue incoming links than the link (r, v) can be colored either red or blue.

As an example, we consider the node arrangement process of the graph depicted in Figure 3-(a). Initially, $L_0 = \{r, a, b, c, r\}$. Then, the algorithm iteratively, inserts nodes g, h, i to L. The final list is $L = \{r, a, g, i, b, h, c, r\}$. Figure 5-(a) shows the obtained RDAGs after performing the link coloring. The number near each node indicates its location in L, besides the source node r, which has the first and last indexes, 0 and 8, respectively.

C. Availability of Protection

We now show that link-coloring scheme can always find two RTs for any given 2-reachable graph even if new destinations are dynamically added. A formal proof of this property is given in [15]. Recall that the link coloring guarantees Property 1 to any non-source node u. Consider any red path from r to u. Since this path contains only forward links each non-source node along this path appears before u in L. Now consider any blue path from r to u, similar to the red path, every non-source node in the blue path is placed after u in L. This ensures that the two paths are node disjoint. Since each of the RDAGS contains a path from r to each other node, it follows that any two trees with root r, which are provision on the red and blue RDAGs are redundant trees, and provide full protection to the multicast connection.

D. Management and Routing Aspects

Unlike FRR, if RT-based protection is used only two trees are provisioned for each multicast connection. This significantly simplifies the provisioning and management of multicast connections. However, each tree is established according to one of the RDAGs, which implies that the primary tree may not be the shortest path tree to the destinations, even if shortest path routing is used.

V. RESOURCE RESERVATION EVALUATION

This section presents several typical observations from our extensive resource reservation evaluation of the fast-reroute (FRR) and the redundant tree (RT) based restoration approaches.

A. Evaluated Schemes

We evaluated the four FRR variants described in Section III; *Sub-LSPs*, *Tree-detour*, *Facility-P2P* and *Facility-P2MP*. For each one of the variants, we calculated its resource reservation assuming the two resource sharing methods, *Intra-sharing* and *Full sharing*. Since full sharing provides the best possible sharing (no other sharing scheme can do better using shortest path detours), we consider this option as a lower bound for the resource consumption required by each FRR variant.

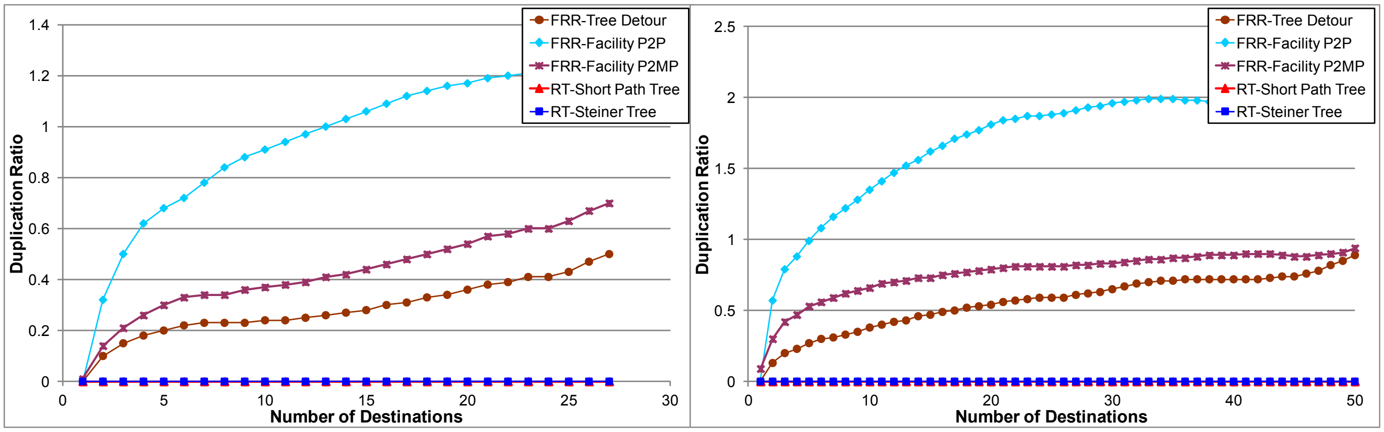
We also simulated the RT-based protection scheme by using the link-coloring algorithm described in Section IV and [15]. We consider two variants of this scheme based on two different tree-calculation algorithms at each RDAG;

RT-Short Path Tree – The RTs are the shortest path trees between the source and destinations at each RDAG.

RMT-Stiener Tree – Each one of the redundant trees is a Steiner Tree (minimal cost tree) between the source and the destination nodes at each RDAG.

TABLE I
TOPOLOGY INFORMATION

Instance	Nodes	Links	Node degree				Link weight			
			min.	avg.	std. dev.	max.	min.	avg.	std. dev.	max.
USLD [12]	28	45	2	3.21	0.94	5	11.0	29.1	12.9	62.0
Metro [22]	51	60	2	2.35	0.62	5	10.0	25.3	8.5	30.0



(a) USLD – Duplication ratio.

(b) Merto – Duplication ratio.

Fig. 6. Duplication Ratio for the USLD and Metro networks.

We emphasize that in our evaluation sharing is only applied to the FRR variants. No sharing is used in the RT-based schemes. For the later approach, the resource consumption is calculated as the summation of the individual bandwidth consumed by each one of the redundant trees.

B. Simulation Tool

For our evaluation we built a custom-made C++ simulator, which leverages the graph representation and algorithmic capabilities of the Boost [23] Library. Our simulator accepts as input a network topology, including the link costs, a source and destination nodes, as well as a specific protection scheme. The provided scheme may be either a FRR variant or a variant of the RT approach. Then the simulator calculates the primary and backup paths from the source to each destination for the given protection scheme. Recall that for FRR the primary paths are along the shortest path from the source to each destination, while the backup paths are the detours as specified by each FRR variant. For the redundant tree approach the simulator calculates red and blue RDAGs. Then, it calculates either the shortest-path-tree or the Steiner tree at each RDAG according to a given RT scheme. At the end, it returns the selected paths.

In addition to the simulator, we implemented two additional auxiliary programs. The first emulates a network with large number of unicast and multicast connections and invokes the simulator for calculating the primary and backup paths for each connection. The second receives as input the routes of the different connections (as provisioned by our simulator) and calculates the overall resource consumption for primary and restoration traffic according to a given resource sharing option.

C. Evaluated Network Topologies

We use multiple tier-1 and 2 network topologies for evaluating the different protection schemes. Due to space limit, we describe only two topologies which are commonly used for evaluating protection schemes;

US Long Distance Network (USLD) [12]– This is a nationwide network with considerable path diversity. It contains 28 nodes and 88 links.

Metropolitan network (Metro) [22]– This is an access network with lesser path diversity and can be view of a collection of connected rings. It contains 60 nodes and 66 links.

The network specifications are given in Table I.

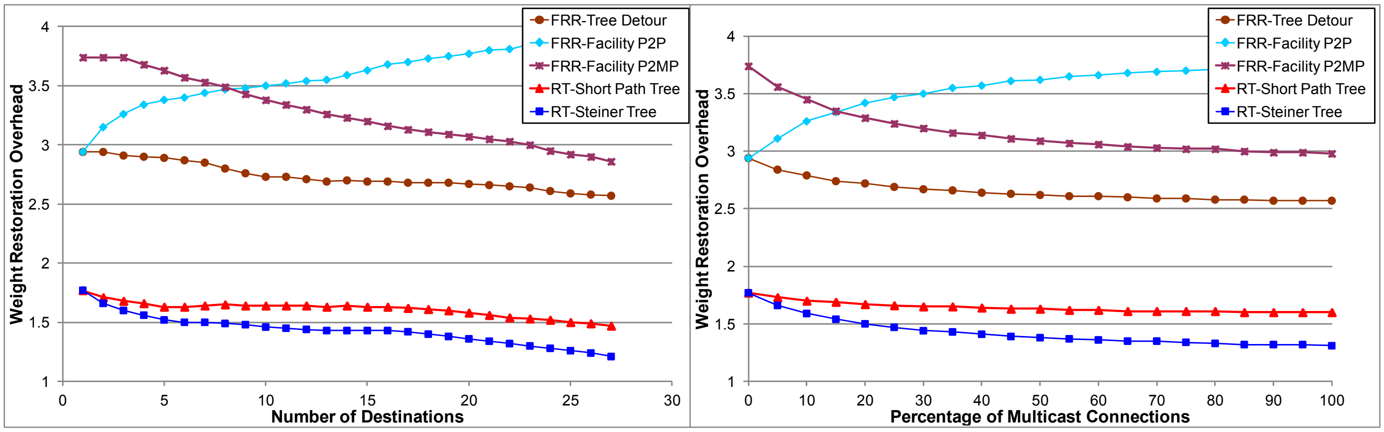
D. Simulation Settings

For a given network topology, each of our simulation runs involved setting 1000 connections generated according to a *connection spec*. Each connection requests one unit of bandwidth. We used two types of connection specs;

Multicast – At each simulation run, the number of destinations of each one of the 1000 multicast connections is fixed. Our simulation runs generate 1000 multicast connections, each with the specified number of randomly selected destinations and originated at a randomly selected source node. Every source node has the same probability of being selected.

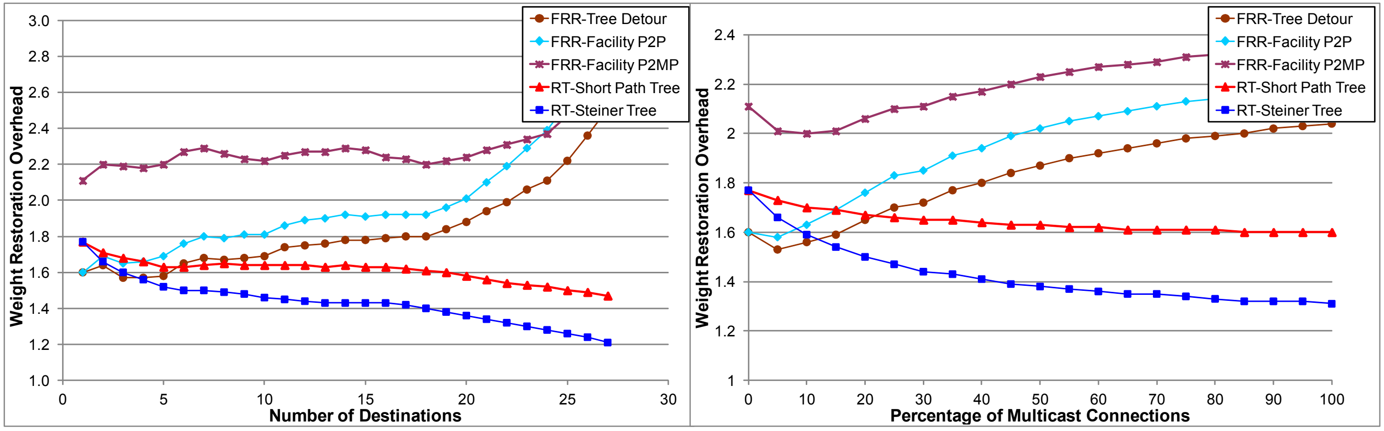
MIX – Such specification considers a mixture of unicast and multiast connections. Each run generates the specified number of unicast connections and the rest of the 1000 connections are multicast connections with number of destinations ranging from 70% to 90% of the total number of nodes.

In both specs, the source node and the destinations of each one of the multicast connections are randomly selected.



(a) USLD Multicast – Restoration Overhead for Intra-sharing.

(b) USLD Mix – Restoration Overhead for Intra-sharing.



(c) USLD Multicast – Weight Restoration Overhead for Full-sharing.

(d) USLD Mix – Restoration Overhead for Full-sharing.

Fig. 7. Weight Restoration Overhead evaluation for the USLD network.

E. Evaluation Metrics

We considered two metrics for evaluating the performance: **Duplication ratio (DR)**: For a given connection, DR is the ratio of the number of links in the network where two or more bandwidth units are reserved and the number of links used by the primary tree. For each of our simulation runs, DR is the average of the DRs for each of the 1000 connections.

Weight Restoration Overhead (RO): For a given connection,

$$RO = \frac{TW - PW}{PW}$$

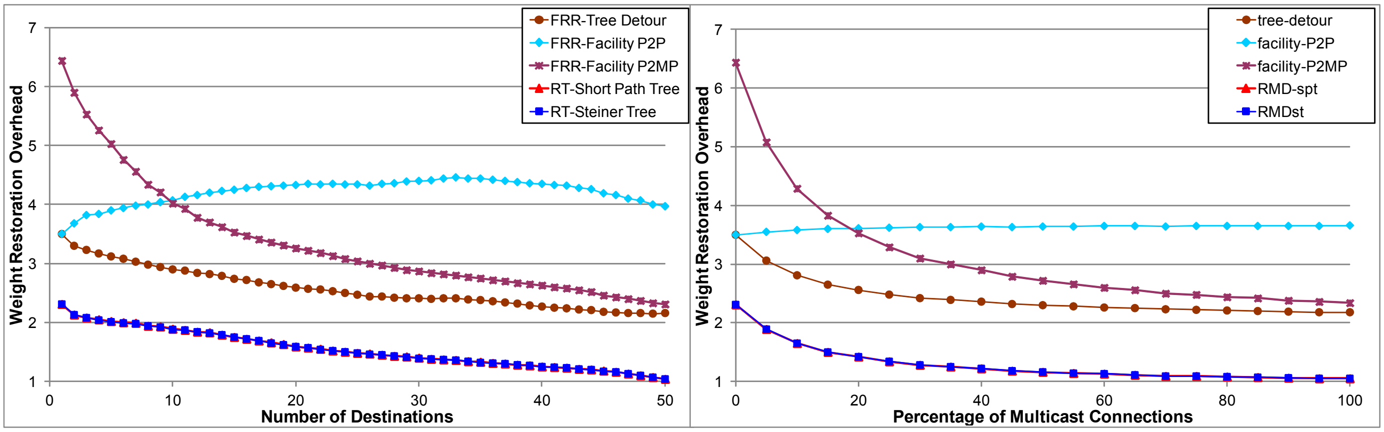
TW is the *total weight* of the allocated network resources for primary and backup paths. Recall that if multiple units of bandwidth are allocated on a link, then the links contribution to TW is its weight times the number of allocated resources on this link. PB is the *primary tree weight* when it is routed as a Steiner tree. Note that for the case of FRR variants, RO coincides with the *restoration ratio* metric used in previous work [1], [5]. We use RO to provide a common basis for comparing RTs based schemes with the FRR schemes and also with previous related work. RO also provides a good evaluation of the restoration overhead relative to the possible lower-bound. Since PW is the cost of the Steiner tree between the source and the destinations, the total weight of all the detours of a given connection must be at least as high as PW. Hence, $RO > 1$ and it shows how close the restoration overhead of a scheme to the possible lower bound.

F. Evaluation Results

The results of our simulations are presented in Figures 6,7 and 8. Every point on each of the charts is computed as the average of 10 simulation runs each using an identical connection spec. We made the following observations:

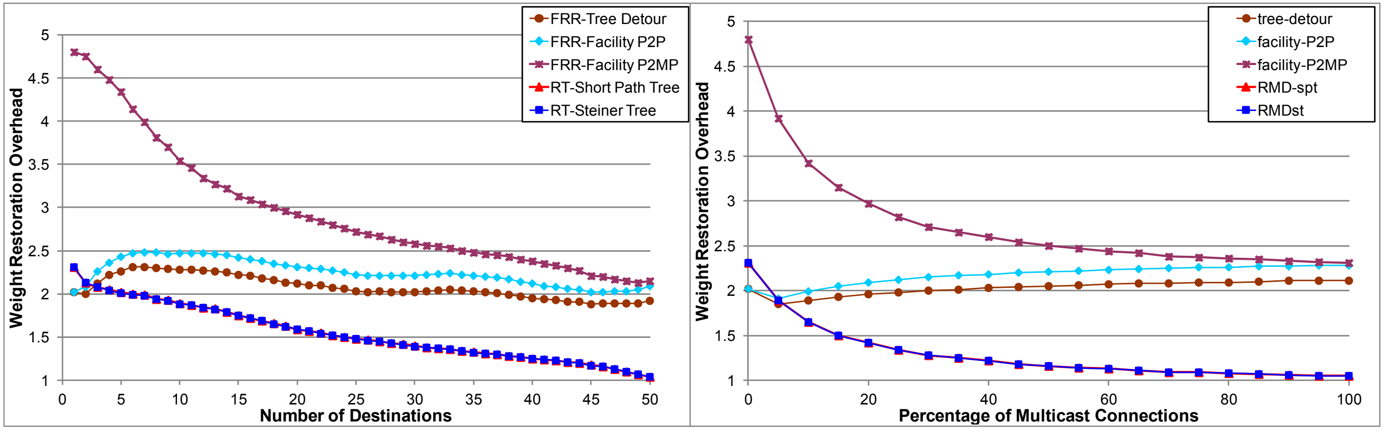
We observed that the Sub-LSPs FRR variant requires substantially higher amount of resource reservation than all the other variants and the gap is enlarged as the number of destinations increases. Therefore, for clear visual comparison of the other options, it has been removed from our charts.

Figure 6 shows that *packet duplication* (DR) is a substantial issue for all variants of FRR, it becomes increasingly significant as the number of multicast destination nodes increases. In particular, the DR is very high when the FRR-Facility-P2P scheme is used. For this variant, with just 2 or 3 destinations, DR already exceeds 1 for the Metro network, meaning there are at least as many links with duplication as there are on the primary tree. Such situation occurs only if some of the non-primary-tree links are used by multiple detours. For the two FRR tree-detours variants, DR is also very high but by



(a) Metro Multicast – Restoration Overhead for Intra-sharing.

(b) Metro Mix – Restoration Overhead for Intra-sharing.



(c) Metro Multicast – Weight Restoration Overhead for Full-sharing.

(d) Metro Mix – Restoration Overhead for Full-sharing.

Fig. 8. Weight Restoration Overhead evaluation for the Metro network.

definition it is smaller than 1. We notice that overall the DR values are higher for the Metro network than the USLD network, since the former has lower path diversity. Also, the FRR-Facility P2MP variant has higher DR than the FRR-Tree-Detour variant, since in this variant P2MP detours may end at merge points (MPs) that are not on the primary tree and are not affected from the failure. Finally, as expected, *the RT-variants dont suffer from packet duplication.*

Figures 7-(a),7-(b),8-(a) and 8-(b) show that the FRR variants produce high restoration overhead (RO) even when Intra-sharing is used. The RO of the USLD network is between 2.5 to 4 for both the multicast and MIX specs, while the RO of the Metro network is between 2.2 and 6.5. These values are much higher than the RO of the RT schemes.

The RO values of the FRR schemes are significantly improved when Full-Sharing is used. For unicast connections the values are between 1.6-1.8 for USLD and around 2 for the Metro network, as depicted by Figures 7-(c),7-(d),8-(c) and 8-(d). However, the RO values increase with the number of destinations or portion of multicast connections.

Although the RT-variants do not benefit from resource sharing, they perform on par with the FRR schemes for unicast connection when Full-sharing is used. As the number of destinations increases we observe that RO values of the RT-variants are decreasing (while the RO valued of the FRR scheme are increasing), and they converge to values near 1, which is the RO lower bound. This means that the *RT-schemes not just outperform the FRR-variants in the case of multicast connections, their restoration overhead is almost optimal, without any sharing.* These observations are supported also by the simulation results presented in [15].

We also note that our FRR results are consistent with the results presented in [5], which used the USLD topology.

We also note that our results for the case where the number of destinations is 10% of the nodes in $USLD$ is consistent with the results presented in [5] which uses the same network and data points.

VI. CONCLUSION

This study explored the trade-offs between fast-reroute (FRR) and redundant-tree (RT) based protection schemes for multicast and unicast connections. FRR is de-facto the most commonly used local restoration scheme in many networking technologies, such as MPLS and MPLS-TE, while the RT method is the predominant approach for end-to-end protection. Although we mainly focused on the resource reservation efficiency of these approaches, we consider other key parameters as well. We observed that the two approaches have different merits and weaknesses, which make them appropriate to different requirements. Below we summarize our key observations.

A. Fast-Reroute

There is no doubt that the main advantage of the FRR scheme is its ability to provide very fast recovery in the event of a node or link failure. It also benefits from the ability to route the primary traffic along the shortest paths to the destinations. However, these advantages come with high price. FRR requires to provision a detour for every link (and internal node) along the primary paths. The large number of detours causes not just considerable management complexity, but also inefficient consumption of network resources. Several resource sharing techniques that have been proposed in the literature achieve noticeable reduction of the FRR resource consumption with the cost of additional signaling and management complexity. Yet for multicast connection even with the best possible resource sharing method, the restoration overhead is still high. Furthermore, we have shown that in some situations FRR cannot provide protection against all failures although the network has two-disjoint paths from the source to every destination.

B. Redundant-Trees

RTs are simpler and easier-to-manage protection alternative. In this method each connection is associated with only two trees that together ensure two node disjoint paths from the source to each destination. This simplicity yields also efficient resource utilization, which achieves near optimal restoration overhead for medium and large multicast connections. We also proved that RTs can be found if two node-disjoint paths exist between the source and each destination. However, to ensure instantaneous recovery, hot standby is required in which the traffic is sent on both trees. If slightly longer recovery time is allowed, e.g., in the case of *delay tolerance networks*, then cold standby can be used[2]. Recent activity on effective failure detection mechanisms, such as Bi-directional Forwarding Detection (BFD) [20], substantially reduce the failure detection and notification time. This fact coupled with bandwidth and management related issues, make RT-based end-to-end protection increasingly appealing.

C. Summary and Future Work

To the best of our knowledge, this is the first study that compares local-protection based on Fast-Reroute with Redundant-Trees based end-to-end protection. Our comparison considers multiple key aspects, including recovery time, protection availability, resource reservation and management complexity.

We observed that, in one hand, Fast-Reroute allows very fast recovery time with the expense of high resource reservation (even when resource sharing methods are used) and complicated network management. On the other hand, Redundant-Trees offer a much simpler and resource efficient protection scheme, however, hot-standby is required to ensure very fast recover time.

These contradicting attributes of the two approaches raise the fundamental question *if a hybrid scheme can be designed that benefits from the advantages of the two approaches without suffering from their drawbacks*.

---

[2] RTs with cold standby allow further reduction of resource reservation by utilizing resource sharing techniques, however, such reduction will come with the cost of higher management complexity. Hence, this option should be carefully evaluated.

REFERENCES

[1] M. Kodialam and T.V. Lakshman, “Dynamic routing of bandwidth guaranteed multicasts with failure backup,” *ICNP*, Nov. 2002.

[2] A. Raj and O. C. Ibe, “A survey of IP and multiprotocol label switching fast reroute schemes,” *Comput. Netw.*, vol. 51, no. 8, 2007.

[3] P. Pan, G. Swallow, and A. Atlas, “Fast Reroute Extensions to RSVP-TE for LSP Tunnels,” RFC 4090 May 2005.

[4] Katherine Zhao, Renwei Li and Christian Jacquenet, ”Fast Reroute Extensions to Receiver-Driven RSVP-TE for Multicast Tunnels”, Internet-Draft draft-zlj-mpls-mrsvp-te-frr-00.txt, July, 2012.

[5] G. Li, D. Wang, and R. Doverspike, “Efficient distributed MPLS P2MP fast reroute,” *INFOCOM 2006.*, April 2006.

[6] Aiguo Fei, Junhong Cui, Mario Gerla., Dirceu Cavendish ”A Dual-Tree Scheme for Fault-Tolerant Multicast” in *IEEE ICC 2001*

[7] W. Lau, S. Jha, and S. Banerjee, “Multicast resilience with quality of service guarantees,” Tech. Rep. UNSW-CSE-TR-0408, Univ. of New South Wales, Sydney, Australia, 2004.

[8] A. Itai and M. Rodeh, “The multi-tree approach to reliability in distributed networks,” in *IEEE FOCS*, 1984.

[9] M. Medard, S. G. Finn, and R. A. Barry, “Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs,” *IEEE/ACM Trans. on Networking*, vol. 7, no. 5, 1999.

[10] G. Xue, Li Chen, and K. Thulasiraman, “Quality-of-service and quality-of-protection issues in preplanned recovery schemes using redundant trees,” *IEEE JSAC*, 21:8, Oct. 2003.

[11] W. Zhang, G. Xue, J. Tang, and K. Thulasiraman, “Linear time construction of redundant trees for recovery schemes enhancing qop and qos,” *INFOCOM 2005.*, March 2005.

[12] L. Kong, M. Ali, and J.S. Deogun, “Building redundant multicast trees for preplanned recovery in wdm optical networks,” *Journal of High Speed Networks*, vol. 15, no. 4, 2006.

[13] Y. Bejerano and P.V. Koppol, “Optimal construction of redundant multicast trees in directed graphs,” in *INFOCOM 2009.*, April 2009.

[14] Y. Bejerano, S. Jana and P. Koppol, “Efficient Construction of Directed Redundant Steiner Trees” *LCN’12.*

[15] Y. Bejerano, S. Jana and P. Koppol, “Link-Coloring Based Scheme for Multicast and Unicast Protection” *HPSR’13.*

[16] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, “Routing bandwidth guaranteed paths with local restoration in label switched networks,” *ICNP*, vol. 0, pp. 110, 2002.

[17] R. Doverspike, G. Li, K. Oikonomou, K.K. Ramakrishnan, and D. Wang, “IP backbone design for multimedia distribution: Architecture and performance,” *INFOCOM 2007*, May 2007.

[18] M. Bocci, S. Bryant, and L. Levrau, “A Framework for MPLS in Transport Networks,” Internet-Draft draft-ietf-mpls-tp-framework, IETF, July 2009, Work in progress.

[19] I. Busi and B. Niven-Jenkins, “MPLS-TP OAM Framework and Overview,” Internet-Draft draft-ietf-mpls-tp-oam-framework, IETF, Mar. 2009, Work in progress.

[20] R. Aggarwal, K. Kompella, T. Nadeau, and G. Swallow, “BFD for MPLS LSPs,” Internet-Draft draft-ietf-bfd-mpls, IETF, June 2008, Work in progress.

[21] R. Aggarwal, D. Papadimitriou, and S. Yasukawa, “Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs),” RFC 4875, May 2007.

[22] E.B. Basch, R. Egorov, S. Gringeri, and S. Elby, “Architectural tradeoffs for reconfigurable dense wavelength-division multiplexing systems,” *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 12, no. 4, July-Aug. 2006.

[23] Boost C++ Libraries, *http://www.boost.org/.*