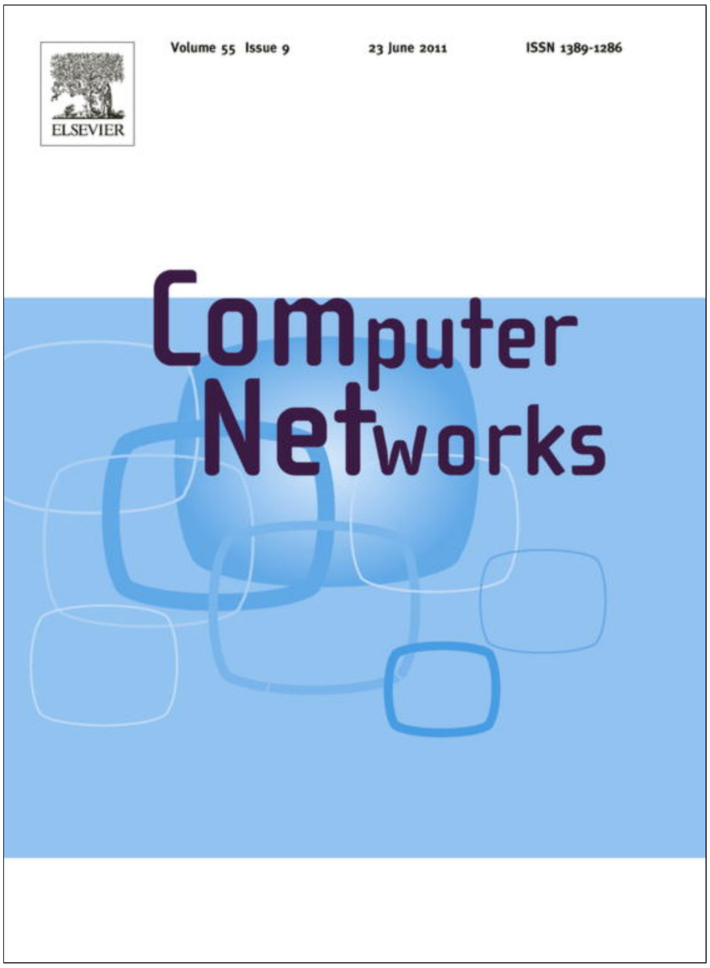


Volume 55 Issue 9 23 June 2011 ISSN 1389-1286

ELSEVIER

Computer Networks

Computer Networks 55 (2011) 2329–2351

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Cross-layer failure restoration of IP multicast with applications to IPTV

M. Yuksel [a,*], K.K. Ramakrishnan [b], R.D. Doverspike [b], R.K. Sinha [b], G. Li [b], K.N. Oikonomou [b], D. Wang [b]

[a] *University of Nevada – Reno, 1664 N. Virginia Street, Reno, NV 89557, USA*
[b] *AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, USA*

ARTICLE INFO

Article history:
Received 3 June 2010
Received in revised form 8 November 2010
Accepted 28 March 2011
Available online 5 April 2011
Responsible Editor. R. Sivakumar

Keywords:
Multicast
Fast-reroute
Cross-layer design
PIM reconfiguration
IPTV

ABSTRACT

Recent applications such as broadcast TV distribution over an IP network require that stringent QoS constraints, such as low latency and loss be met. Streaming content in IPTV is typically delivered to distribution points on an IP backbone using IP multicast, in particular Protocol Independent Multicast (PIM). Local restoration from link failures using MPLS or layer-2 Fast Reroute (FRR) is a proven technique to achieve rapid failure restoration. Link-based FRR creates a pseudo-wire or tunnel in parallel to the IP adjacencies; thus, single link failures are transparent to the Interior Gateway Protocol (IGP) such as OSPF. Although one may choose the back-up path's IGP link weights appropriately to avoid traffic overlap during any single physical link failure, multiple failures may still cause packet loss because of (a) congestion resulting from overlap of an active FRR path with the multicast tree, (b) congestion resulting from overlap of two active FRR paths, or (c) a hit resulting from an OSPF reconvergence after the failure of a link in an active FRR path. We present a cross-layer restoration approach that combines both FRR-based restoration for single link failures and "hitless" (i.e., without loss) PIM tree reconfiguration algorithms to prevent traffic overlap when multiple failures occur. We demonstrate the efficacy of our schemes through simulations. The average recovery time on double failures can be reduced from more than 10s to only approximately 100 ms with our enhancements.

1. Introduction

Distribution of real-time multimedia over an IP backbone has been gaining momentum with content and service providers [1–4]. However, unlike traditional cable-based infrastructures that provide "broadcast" analog-based video (e.g., TV), using an IP backbone for real-time broadcast video distribution imposes stringent requirements for protection and restoration after a failure. Distribution of real-time broadcast (also called "linear") TV requires that the delay experienced by a user viewing the content be limited to less than a few seconds. This limits the size of the playout buffer at the receiving IPTV set-top box. Moreover, loss recovery mechanisms have a limited capability to recover from burst packet losses. Player loss-concealment algorithms and recovery through retransmissions and packet-level redundancy mechanisms such as forward error correction (FEC) are designed to recover from burst losses that are no more than a few tens of milliseconds. The tight QoS constraints of low latency and loss need to be met even under failures. Meeting these constraints while providing high network availability requires a methodology for rapid restoration [3]. In [14], the authors report on diverse measurements from a large commercial IPTV. An analysis of the customer trouble tickets indicates that nearly half are related to performance issues such as video quality. Video quality monitors in the service provider's network indicate almost 79% of the alarms were related to under-runs of the video playout

[*] Corresponding author.
E-mail address: yuksem@cse.unr.edu (M. Yuksel).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.03.015

buffer. Finally, analysis of SNMP and syslog data indicated that almost 55% of the performance-related events were due to layer-1 alarms and IP link flaps.

Use of IP-based protocol independent multicast (PIM) [5–7] to distribute the content from the source to the various distribution points on the IP backbone allows the infrastructure to be cost-competitive with the more mature cable broadcast infrastructure. However, PIM-SSM depends on a "join" and "prune" process to rebuild the multi-cast tree after a network failure. This process, when combined with the Internal Gateway Protocol (IGP) reconfiguration, which may take several seconds or tens of seconds, takes too much time to restore the packet flow to the receivers after a failure. Thus, packet loss concealment and recovery mechanisms alone are not effective in recovering from such a long period of packet loss.

A common approach to providing the needed high availability is link-based Fast Reroute (FRR) [8,9]. If the network is subject to only single link failures, enhancing the IPTV backbone with link-based FRR works well. However, in a long distance network, the probability of multiple link failures is not negligible. Our analysis of failures over a four-month period from a commercial IPTV deployment with more than 2 million customers revealed that in 17% of the cases, at least 2 links were down at the same time, and in 2% of the cases the failures of 3 or more links overlapped. Markopoulou et al. [32] found an even higher occurrence of multiple failures in their analysis: 27% of failures in Sprint's operational IP backbone were multiple failures.

We list three reasons why a network employing FRR may incur significant loss under multiple failure situations (We illustrate these scenarios with examples in Section 3.1):

1. *Overlap of active backup path with multicast tree*. The IGP reconfiguration process is typically unaware of when traffic is rerouted by link-based (layer-2) FRR. Thus, when a second failure occurs before the first failure is repaired, the packets of a given multicast flow (e.g., packetized video for individual IPTV channels) may travel over the same directed link two or more times. We call this phenomenon *traffic overlap*. Congestion may occur during traffic overlap because large-scale, real-time video streams often need 2–3 Gb/s or more in aggregate across all the channels and thus link capacity may not be sufficient to support *two* instances of all these flows.
2. *Overlap of two active backup paths*. Even though one may choose the FRR back-up paths to avoid traffic overlap with the multicast tree [10], there is no guarantee that the backup paths of two different failed links are not going to overlap. Thus if two links fail and are repaired using FRR, congestion may happen on links where the two backup paths overlap.
3. *Extra IGP reconvergence*. A link failure is repaired using FRR but then a subsequent link fails in the failed link's backup path and forces OSPF and PIM reconfiguration resulting in several 10s of seconds of packet loss during the protocol reconvergence.

Because of its stringent QoS constraints, loss from congestion often has the same effect on the customer's perception of service as an unprotected link failure. As failures often take several hours to repair, congestion caused by overlaps due to multiple failures can last for a long time.

We illustrate the issue of overlaps with three example network topologies in Section 6. For those topologies, we verified with nperf tool [26] that 8.4%, 10%, and 27% of the double failure cases result in congestion due to overlap or forced OSPF/PIM reconfiguration. Combined with our earlier observation that multiple failures constitute a significant fraction of all link failures, the potential impact on traffic outlined above must be avoided.

The best way to achieve this is to use FRR only for a short period, then remove (or "cost-out") the failed link from the topology, and run an IGP reconfiguration. However, if it is not carefully executed, in the presence of multiple failures this procedure can cause as many new video interruptions as it removes, due to small "hits" after single failures. This paper describes a careful multicast recovery methodology to achieve the benefits of multicast while avoiding the drawbacks.

1.1. Contributions

The research literature on dealing with multiple failures is primarily concerned with providing connectivity in unicast networks (e.g. [11]). Traffic overlap is generally not considered. Thus, the proposed solutions require that certain links (with traffic overlap) have additional capacity. Our approach does not simply consider connectivity, but specifically targets avoiding traffic overlaps. Further, an IP network carrying high-bandwidth real-time traffic needs to exploit multicast distribution to be cost-effective, and the load on individual links is high enough that there is not substantial capacity left to support the additional traffic that may also be re-routed after a failure. Thus, we seek an efficient multicast tree reconfiguration protocol. Even without using FRR, IGP will reconverge to a new multicast tree and will eventually have no congestion. However, IGP convergence tends to be slow, and production networks rarely shorten their timers enough to allow sub-second failure recovery because of the potential of false alarms [12]. So FRR is needed in order to restore rapidly, and it works very well in a large number of scenarios (e.g., 83% of the failure cases described above, consisting of a single link failure). However in the remaining 17% of multiple link failure cases it may cause traffic overlap.

We try to achieve the best of the both worlds by taking a cross-layer approach of making the IGP routing aware of link failures that are restored by FRR. This allows triggering a reconvergence of IGP routing. Then we allow multicast routing (i.e., PIM-SSM) to see the IGP routing changes immediately, even before full IGP reconvergence, and assure that multicast routing reconfigures without an additional hit. We assume the existence of a combination of video player loss-concealment algorithms, packet-level redundancy mechanisms such as forward error correction (FEC), and retransmission-based recovery, to overcome burst losses that are short, up to about 50–100 ms. The mechanisms we consider here work in a complementary manner with these loss-tolerance and recovery methods.

The multicast reconfiguration protocol that we develop does not have to wait for complete IGP reconvergence over the entire topology (which can take several seconds) before it begins to reconfigure the multicast tree. Thus, it seeks to limit the time the network is exposed to the possibility of a second failure, which may result in traffic overlap. Moreover, the IGP reconvergence typically affects a small subset of nodes, those with a failed link in their shortest path to the root. Thus, most nodes will still have the same shortest path in the new multicast tree after a link failure.

In this paper, we present protocol details and correctness proofs of our protocol, and extensive evaluation of its performance under multiple failures. Our major contributions are:

- Specification of a protocol for reconfiguring multicast trees via "pending joins" without waiting for IGP reconvergence.
- Proof that the proposed protocol consistently reconfigures the multicast tree even in the presence of multiple link failures.
- Implementation of the proposed protocol in a packet-based simulator and evaluation under double link failures.

While our proposed algorithms and protocols may be more generally applicable, we focus on IPTV because it is a prototypical application requiring very low packet loss.

In the next section, we cover related work on multicast reconfiguration. We, then, provide motivation for our multicast reconfiguration using cross-layer information in Section 3. Section 4 outlines the challenges involved in the cross-layer multicast reconfiguration and provides a conceptual description of our approach. Section 5 details our cross-layer multicast reconfiguration (i.e., IGP-aware multicast reconfiguration) protocol and provides correctness proofs. Section 6 presents ns-2 simulation results under various scenarios. We conclude in Section 7.

2. Related work

Much of the recent research in IPTV focuses on architecture design [14], protocol design and selection [31], multimedia stream coding/decoding techniques [15,17,27], as well as potential new applications of multicast [16,18]. There has also been extensive IP multicast research and experiments [13]. The IETF has standardized multiple IP multicast protocols, including PIM-DM [5], PIM-SM [6], and PIM-SSM [7], and has made recommendations for reliable IP multicast [17]. To improve IP multicast performance, many models and techniques have been proposed, including the overlay model [18] and MPLS P2MP [19], resource reservation and admission control to avoid congestion [20], centralized management of PIM over MPLS [21], and priority queueing or fair queueing to guarantee quality of service. These solutions have to be adapted to be usable in a carrier's multimedia distribution service. Approaches such as those proposed for reliable multicast [22] are inapplicable, as they can add latency that is unacceptable in an IPTV environment.

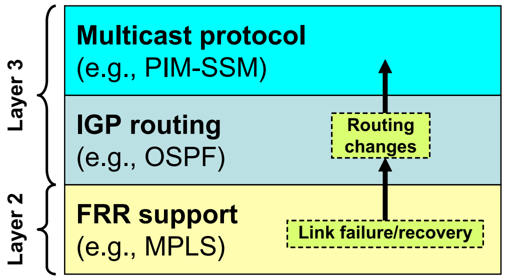


Fig. 1. Cross-layer architecture for multicast failure restoration: The multicast protocol agent is notified about IGP routing changes that are triggered by link failures. The routing changes are exposed to the multicast agent without waiting for full IGP reconvergence, and the agent starts reconfiguring its tree as soon as it thinks (partial) reconfiguration is possible.

Network restoration has also been an active research area for many decades. In [10], an algorithm for setting IGP link weights intelligently was proposed in an IPTV setting that uses PIM-SSM multicast trees and link-based FRR for failure restoration. The resulting link weights ensure that the multicast tree is disjoint from the backup path of every link, and therefore a single link failure can be recovered using FRR without causing any traffic overlap. However, multiple failures may still cause congestion and traffic overlap, and as noted earlier, these could be significant (as much as 17% of all link failures.)

As video and real-time services migrate to IP-based environments, IP network restoration has also increased in importance, but little work has been done on performance evaluation of different restoration schemes in multimedia backbone design. Although simulations show that large IP networks are able to achieve sub-second convergence for the routing protocol by tuning its timers [23], service providers have not adapted such schemes due to concerns regarding network stability. Other schemes, like failure-insensitive routing [24], apply to unicast routing and are not suitable for multicast. (see Fig. 1)

3. Cross-layer multicast reconfiguration: Motivation

To understand our motivation for using cross-layer failure restoration techniques, we describe in more detail the interaction between multiple failures and FRR. Consider two adjacent routers in an IP network used for IPTV distribution where the routers are part of a multicast tree using PIM-SSM [7]. Assuming the adjacency between these routers is restorable via FRR, as Fig. 2 shows, four different links are visible to the IGP topology: two unidirectional or directed pseudowires (dashed lines between nodes E and C for example), and two unidirectional physical-layer "PHY" links (solid lines between E and C). The pseudowires are associated with a primary path and a backup path, which are typically LSPs. The primary path for each

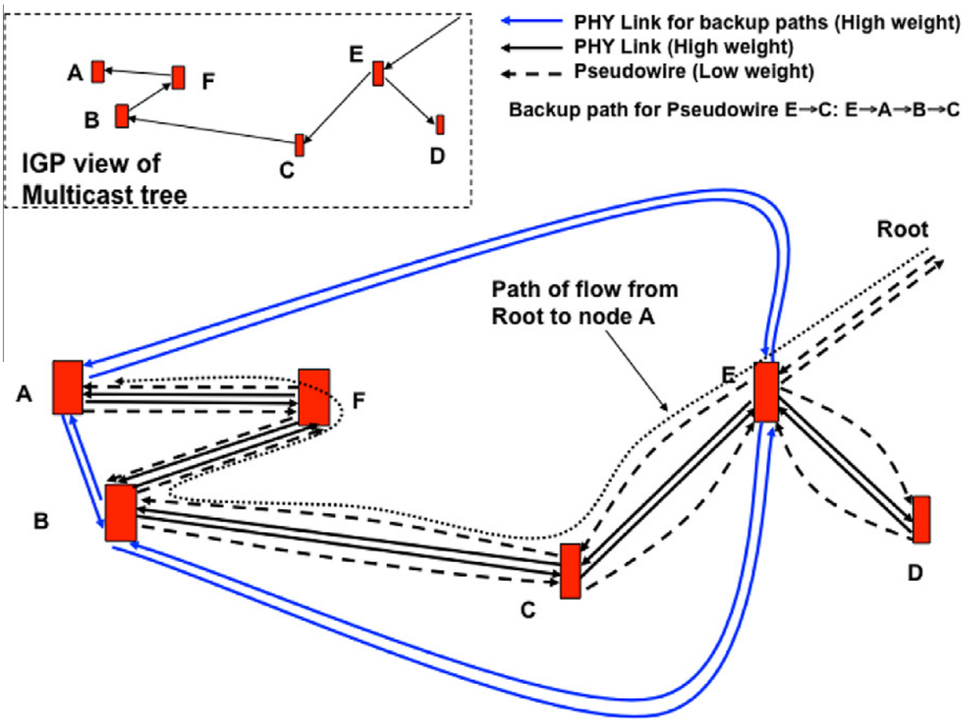

PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
Backup path for Pseudowire E→C: E→A→B→C
IGP view of Multicast tree
A
B
C
D
E
F
Root
Path of flow from Root to node A

Fig. 2. A network segment with pseudowires, PHY links, and a sample FRR backup path.

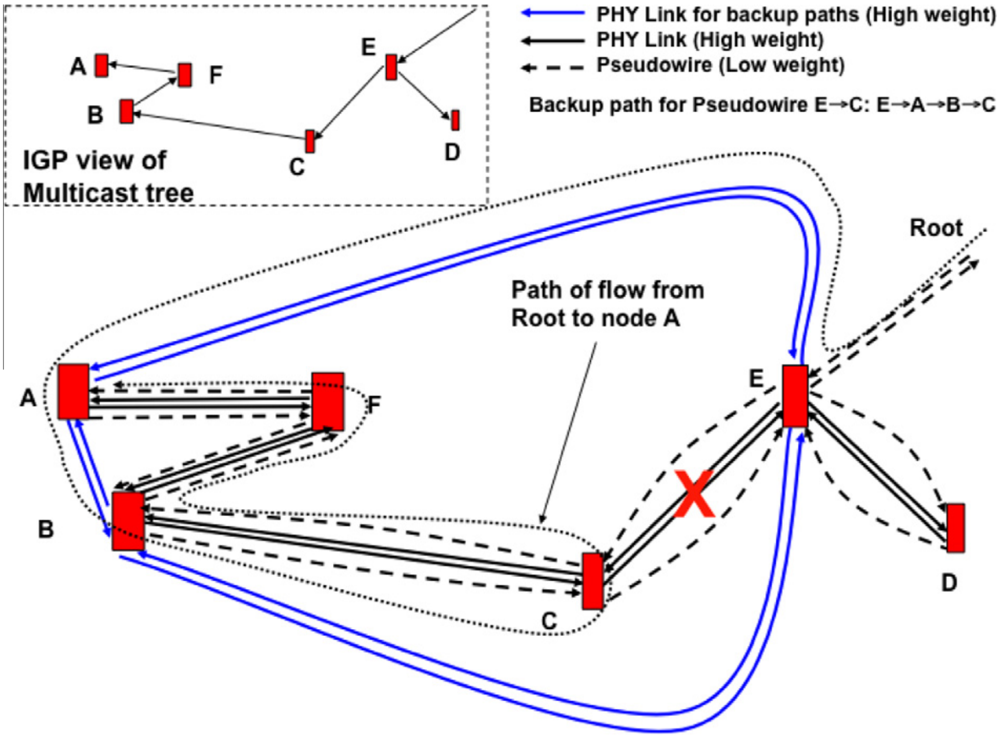

PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
Backup path for Pseudowire E→C: E→A→B→C
IGP view of Multicast tree
A
B
C
D
E
F
Root
Path of flow from Root to node A
X

Fig. 3. Single link failure: FRR backup path is activated and IGP is unaware of the failure. Multicast tree does not change.

pseudowire is its corresponding PHY link, and the backup path routes over other PHY links that are disjoint from the primary path. The IGP costs on the pseudowire links (to which we generically refer as *weights*) are lower than those for the PHY links. This causes the IGP shortest path algorithm to route over the pseudowire links rather than the PHY links in a non-failure state. Thus, when either of the PHY links fails, both links are taken out of service and the two pseudowires are switched from their primary paths to their backup paths, usually with a target switching time of 50 ms or less. This time is sufficiently small that the higher-layer packet loss concealment and recovery

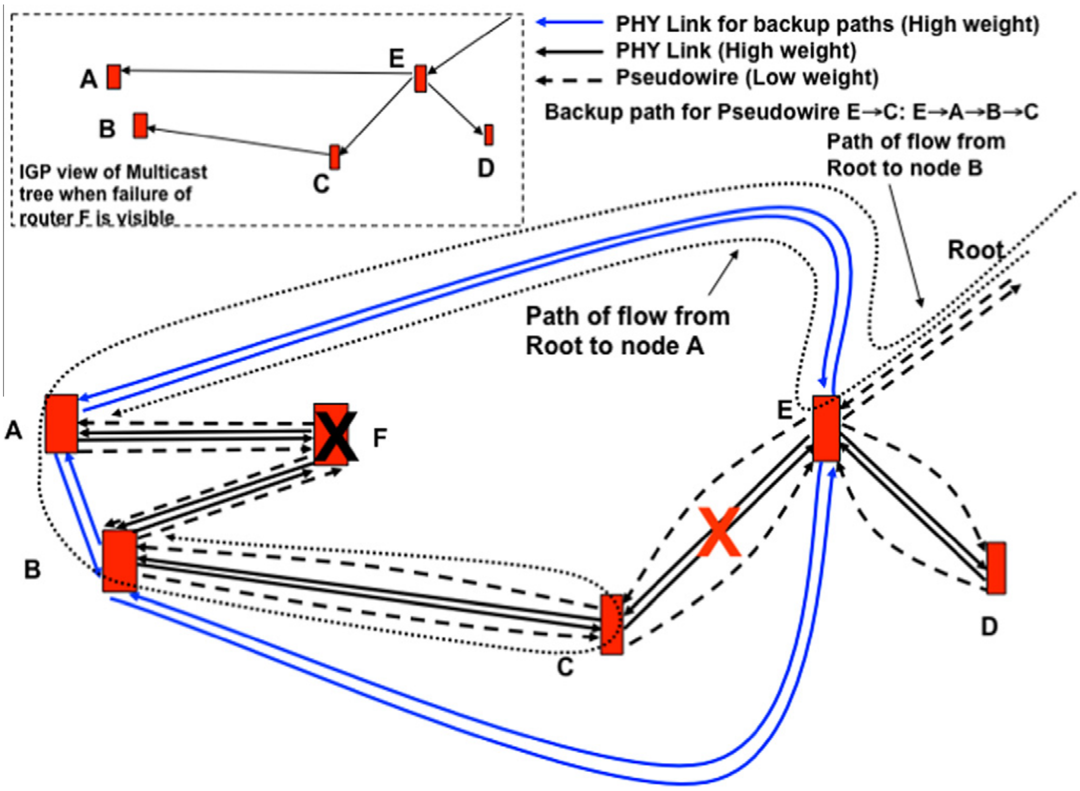

PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
Backup path for Pseudowire E→C: E→A→B→C
Path of flow from Root to node B
Root
Path of flow from Root to node A
IGP view of Multicast tree when failure of router F is visible
A
B
C
D
E
F

Fig. 4. (Scenario 1) Multiple failures with one link and one router failing: FRR backup path overlaps with multicast traffic tree on E → A. Failure of router F is visible to IGP, but failure of link E-C is not.

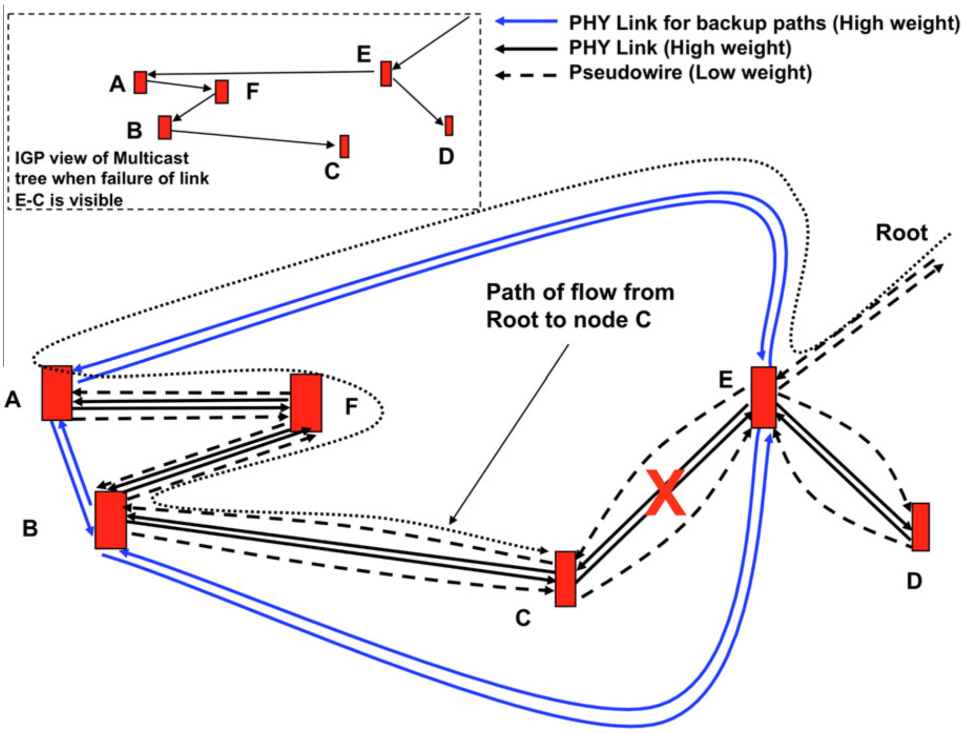

PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
IGP view of Multicast tree when failure of link E-C is visible
Root
Path of flow from Root to node C
A
B
C
D
E
F

Fig. 5. Single link failure with our proposed solution: IGP is made aware of the failure of PHY link E-C, and converges to a new multicast tree that does not use the pseudowire E-C (compare with Fig. 3).

protocols can recover from this failure with little or no perceived video quality disruption. In addition, the backup pseudowire will switch to the backup path well before the IGP timers expire. Therefore, when the IGP link state advertisements (LSAs) are broadcast, although they show that the PHY links are down, the pseudowire link states remain unchanged and there is no change to the IGP shortest path tree and the multicast tree.

We give an example of this in Figs. 2–4. Fig. 2 depicts a network segment with 4 node pairs that have pseudowires defined. For example, node pair E-C has a PHY link in each direction and a pseudowire in each direction for restoration (a total of 4 directed links). We also depict the backup path for the pseudowire E → C. Note that certain node adjacencies, such as E-A or E-B, are provided for restoration, hence have no pseudowires defined. From the point of view of the IGP topology, pseudowires can be thought of as virtual links; in fact, the IGP only differentiates the 4 links between a node pair by their weights and interface IDs. Pseudowire E → C (dashed) routes over a primary path, which consists of the single PHY link E → C (solid). If a failure occurs on the primary path, the router at node E attempts to switch to the backup path using FRR. Fig. 3 illustrates such a single layer-1 failure of link E → C. Note that the path from the root to node A, shown in the inset at the top, now switches to the backup path at node E (E-A-B-C), reaches node C, and then continues on its previous (primary) path to node A (C-B-F-A). During the failure, although the path retraces itself between routers B and C, because of the directed links the multicast traffic does *not* overlap. Also, more importantly for this paper, although the IGP view of the topology realizes that the PHY links between E-C have gone down, the shortest path tree, and consequently the multicast tree, remains unchanged because the pseudowire from E → C is still up and has lower weight. The IGP is unaware of the actual routing over the backup path.

3.1. Illustration of traffic losses under multiple failures and our solutions

We demonstrate by detailed examples the three problem scenarios outlined in Section 1. In each scenario, PHY link E-C fails and pseudowire E-C gets repaired using FRR. The multicast tree remains unaffected as in Fig. 3. Then for each scenario, a different failure is the source of the problem. Our proposed solution would have created the network state shown in Fig. 5 and we point out, in each case, how our approach overcomes the problem.

<u>Scenario 1</u>: *overlap of an active backup path with the multicast tree*. Typically, the FRR backup path is kept active until the PHY links are repaired. Once the PHY links are back in service, the pseudowires are switched back to their primary paths rapidly (again with a target switching time of 50 ms or less). This achieves high network availability only if no overlapping link failures occur (by "link" we mean the pair of unidirectional PHY links between a router pair). However, if a second failure occurs while the first failure is being repaired (which may take from a few minutes to several hours), because IGP is unaware of active FRR backup paths, it is possible that traffic overlap may occur. Fig. 4 depicts such a case of multiple failures, where the router at F fails while link E-C is down and E-C's backup path is still active. As a result of node F's failure (which becomes

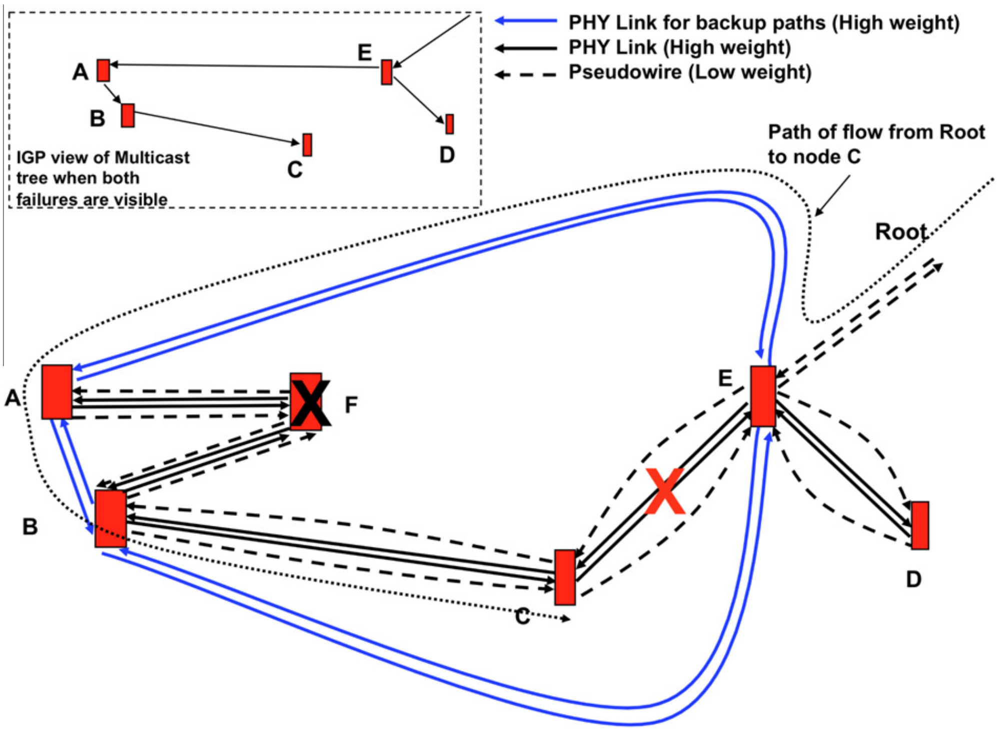

PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
A
B
C
D
E
IGP view of Multicast tree when both failures are visible
Path of flow from Root to node C
Root
A
F
E
B
C
D

Fig. 6. Multiple failures with one link and one router failing: our solutions avoid the overlap in Scenario 1. The overlap is avoided because failures of router F as well as link E-C are visible to IGP (compare with Fig. 4).

visible to IGP), IGP now modifies the shortest path from the root to A, causing this new shortest path and the backup path of E → C to overlap on link E → A.

This traffic overlap can be avoided if we do not leave the backup path active while the PHY link is being repaired. Instead, after the pseudowire is routed over the FRR

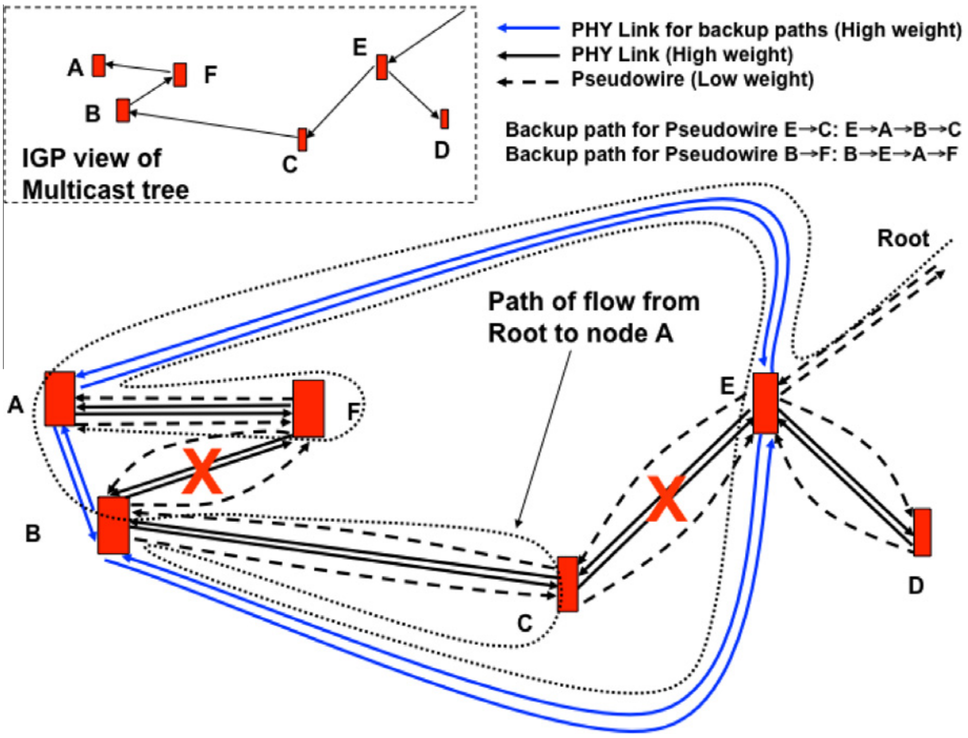

A
F
B
E
C
D
IGP view of Multicast tree
PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
Backup path for Pseudowire E→C: E→A→B→C
Backup path for Pseudowire B→F: B→E→A→F
Root
Path of flow from Root to node A
A
F
B
C
E
D
X
X

Fig. 7. (Scenario 2) Double link failures: failure of B-F starting in state depicted in Fig. 3. The two active backup paths overlap on link E → A.

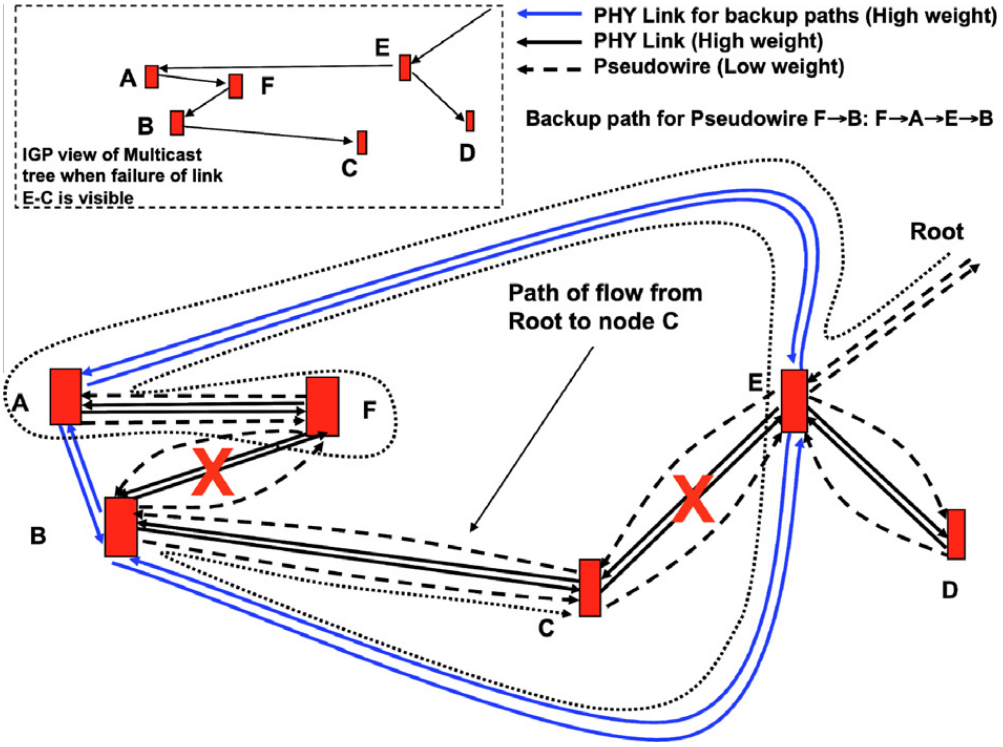

E
A
F
B
C
D
IGP view of Multicast tree when failure of link E-C is visible
PHY Link for backup paths (High weight)
PHY Link (High weight)
Pseudowire (Low weight)
Backup path for Pseudowire F→B: F→A→E→B
Root
Path of flow from Root to node C
A
F
B
C
E
D
X
X

Fig. 8. Multiple link failures: failure of link B-F starting from the state depicted in Fig. 5. Backup path for pseudowire F → B is being used but there is no overlap since the failure of E → C is made visible to IGP. Our solutions avoid the overlap in Scenario 2 (compare with Fig. 7).

backup path upon the first failure, we have IGP reconverge to a new shortest path tree so that the resulting multicast tree does not use the affected pseudowire. Thus, if a second failure occurs, IGP is fully aware of the routing for the tree and can avoid the failed links during reconvergence. In the sample scenario in Fig. 5, shortly after the failure of PHY link E-C, the pseudowires between C and E would be taken down and a new multicast tree would be computed. This tree would not have a branch E → C, but instead would have a path E → A → F → B → C as shown in Fig. 5. Thus, when the second failure occurs, the failure of node F as well as of link E-C would be visible to IGP and consequently node F in Fig. 5 would be pruned from the tree. These two successive tree reconfigurations would result in a multicast tree with no overlaps, as shown in Fig. 6. However, this new approach requires engineering a hitless switch (i.e., with minimal or no packet loss) to the new tree after an FRR reroute. Otherwise, every time a single failure occurs and the backup FRR path is used, there will see a hit associated with converging to the new tree, making the new approach worse than the old one.

Scenario 2: *overlap of two active backup paths*. As shown in Fig. 7, suppose that the pseudowire B → F routes over a primary path, which consists of the single PHY link B → F and the backup path is B → E → A → F. When PHY link B → F fails, the router at node B switches the pseudowire to the backup path, B → E → A → F. The pseudowire in the reverse direction, F → B has the backup path F → A → E → B (not shown in the figure). Then the backup paths of the pseudowires E → C (whose backup path is E → A → B → C) and for B → F overlap on link E → A, which may result in congestion. However, if the multicast tree is reconfigured after the first failure of E → C, then this reconfigured multicast tree will be using link F → B (as per Fig. 5). Then when the second failure (of PHY link F-B) occurs, it fails the multicast link F → B. This results in the use of the backup path F → A → E → B, which has no overlap with the (reconfigured) multicast tree. This situation is illustrated in Fig. 8. Our approach will further expose the second link failure and the tree in Fig. 8 will then be reconfigured again (but not shown in the figure).

Scenario 3: *extra IGP reconvergence due to failure of a link on an active backup path*. In our sample network, A → B is only a PHY link without a pseudowire. Consider the case where the pseudowire E → C fails and its back path E → A → B → C is activated, as shown in Fig. 3. Under that condition, if PHY link A → B fails, it takes down the active backup path E → A → B → C of the first failed pseudowire E → C. This would force an IGP reconvergence since the pseudowire E → C would appear disconnected to IGP. Instead, if we switch the multicast tree to the one shown in Fig. 5 as described in our proposed fix for Scenario 1, this second failure has no effect as A → B is not part of this new multicast tree.

3.2. Need for a hitless switchover of multicast tree

This approach was presented as "Architecture 3" in [10], and as already pointed out, requires a hitless switch to the new tree after an FRR reroute in order to be a real improvement. The primary contributions of the present paper are to propose and describe the algorithms and protocols to perform the switchover to the new multicast tree, and to evaluate their performance via simulation. Furthermore, an advantage of the protocol proposed here is that it reduces the need for the intelligent weight setting algorithm described in [10].

4. Cross-layer multicast reconfiguration: the architecture and switchover protocols

Our main goal in this paper is to attain a multicast protocol that performs effective and efficient switchover from an old multicast tree to a new one. Thus, for the rest of the paper, we will focus on how to achieve a multicast tree switchover given that link/router failures (even though the failed link has an FRR backup path) are notified to the multicast protocol as "IGP routing changes", as shown in Fig. 1. Under normal operation with FRR, such a failure is not visible to multicast protocol and the multicast protocol may not be immediately triggered for a switchover. In our framework, we allow such link failures to be immediately visible to IGP and then to the multicast protocol as a piece of cross-layer information, as shown in Fig. 1. In this section, we will conceptually describe our framework. Since a pseudowire is practically implemented with a primary PHY link and a backup path of PHY links, we will simply refer to the pseudowires and their primary PHY links as "links" to simplify the discussion.

First we describe today's typical carrier IP backbone that may be used for IPTV distribution. We assume that the protocols used for distributing video are PIM-SSM over IGP routing (layer 3), with FRR at layer 2 to recover from failures. This is deployed in at least one carrier network. The underlying changes in the layer 3 topology are first propagated to the rest of the nodes, using standard IGP (e.g., OSPF or IS-IS) techniques. This involves first the detection of the link failure through a lower layer indication, such as SONET or Ethernet alarms, or the lack of HELLOs within a *RouterDeadInterval* [12]; then, the propagation of LSAs via flooding and subsequent topology convergence, which may be a function of the computation of the SPF tree, as well as of the *spfDelay and spfHoldTime* timers [25]. These timers are likely to be set to their default or conservative values in a carrier network, in the interests of stable operation, resulting in an IGP convergence time on the order of several seconds. Subsequently, the PIM-SSM tree has to be reconfigured, which again may take several seconds to tens of seconds (e.g., a new join request is issued after 30 s, as part of the standard process of refreshing the soft-state for IP multicast).

The PIM-SSM tree is typically reconfigured after an IGP reconvergence event in a distributed manner, where each router independently computes the shortest path to the source for each multicast group. After the path recomputation process "settles", routers independently install the new SPF tree and modify the routing and forwarding tables on their line cards. Next, the portion of the tree downstream of the failure is systematically reconfigured step-by-step by each router issuing a join request to attach to

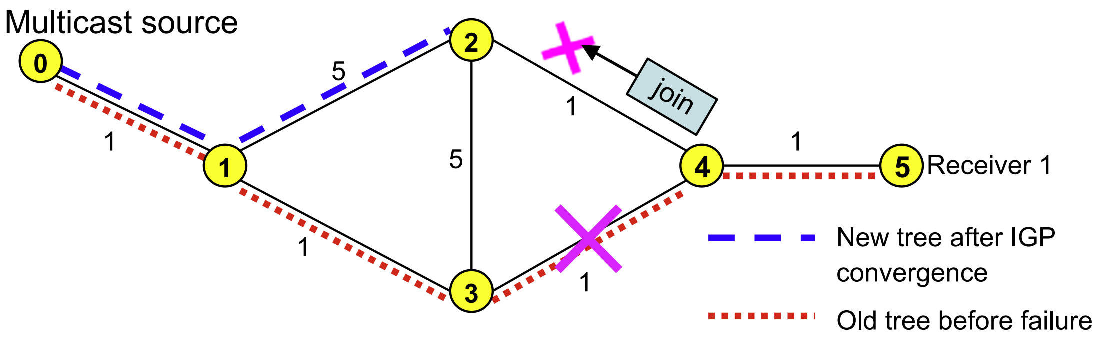


Fig. 9. A sample link failure with join to the new tree is failing due to a lost join message.

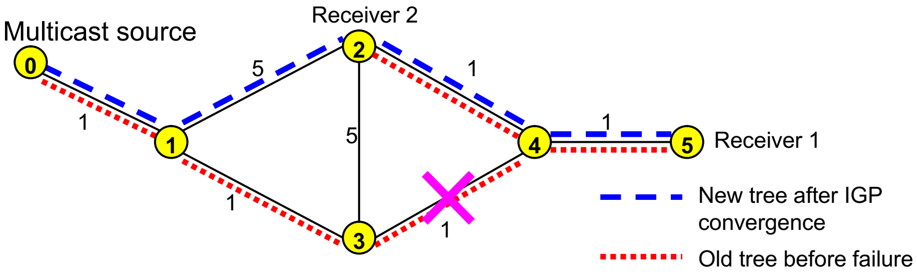


Fig. 10. A sample link failure causing a previously downstream router to be the upstream one on the new multicast tree.

its new parent node, followed by a *prune* message to delete the previous path (a unidirectional link is upstream/downstream of a given router if it points towards/away from the source). During this process, packet loss (a "hit") can occur. We wish to avoid or minimize this potential second hit after a failure and its subsequent FRR rerouting as we cost-out the pseudowire and reconverge to a new tree. Also, the method has to be careful to avoid another hit after the link failure is repaired, e.g. to return to the original PIM tree.

In our method, after a failure the routers coordinate the calculation of the shortest path by implementing *local* decisions on choosing new parent nodes (and the corresponding branches) in the new multicast tree, and then send a *join* request to build a path from its new parent. Before getting into the protocol details in the next section, we describe some of the challenges and our solutions.

4.1. Challenges in designing a robust switchover protocol

The first challenge is that in the current PIM-SSM multicast there is no explicit acknowledgement to a join request. So it is possible that the router stops receiving packets because it sent a prune request to the old parent, but the join to the new parent was not successful. Fig. 9 illustrates this scenario where router 4 attempts to switch from the old tree (red dotted, prior to the failure) to the new blue tree by sending a join message to new upstream router 2. If router 4 sends a prune message to its old upstream router 3 without waiting until the join to the new blue tree is completed (e.g., if the join message to router 2 gets lost), then a significant amount of data packets could be lost. Our solution is a simple modification: the prune message to remove the branch to the previous parent is not sent until the node (router in this case) receives at least one PIM-SSM data packet from its new parent for the corresponding (S, G) group.

Our approach: *prune old upstream interfaces only after a multicast data packet is received on the new upstream interface.*

The *soft state* approach of IP multicast (to refresh the state by periodically sending a join) is also used to ensure consistency. Our approach above does not interfere with transmission of periodic join messages. We use the joins to also guide our tree reconfiguration process at a node in reaction to a failure. Thus, in our approach, routers do not lose data packets during the switchover period. Of course, this primarily works in the PIM-SSM case, with a single source.

The second challenge is that the "link down" LSAs reach different routers at different times. Thus, different routers may have different views of the network topology. As an example, consider Fig. 10 where we are trying to switch from the old tree to the new tree after the failure of link (3, 4). Router 4 needs to send a join to its new parent, router 2. Under standard rules in PIM-SSM, when router 4 sends a join request to router 2, it must stop forwarding packets to router 2, else it would form a loop. However router 2 was receiving packets from router 4 in the old multicast tree and it may not have learned of the topology change yet and has therefore not sent a join to its new parent, router 1. In such a case, router 2, and consequently router 4, will stop receiving packets. The naïve solution is to force every router to wait until enough time has elapsed for all routers to learn of the changes. But this can be quite wasteful. E.g., routers 2 and 4 are not affected by router 5's view of the topology, so they should be able to switch to the new multicast tree without waiting for router 5 to learn the topology changes. This is especially important in a large wide-area network. In order to speed up the convergence to the new tree, router 4 would like to send a join as soon as router 2 learns of the new topology. The key observation is that as soon as router 2 learns the new topology and starts receiving packets from its new parent router 1, it has to send a prune to its old parent, router 4. This prune is the indication to router 4 that it can start receiving packets from router 2. Intuitively, it is easy to see that this is the earliest that router 4 can send a join request to router 2. Further, notice that this decision happens independently at each router; there is no requirement that any other router has to learn of the topology changes. If one were to visualize the old tree transforming into the new tree, each branch changes as soon as it learns of the topology changes. This modification to the protocol is accomplished by introducing a "waiting to send join" state.

The approach described in this paper potentially obviates the need for the intelligent weight setting algorithm described in [10], because any short-lived overlap will be corrected by the switchover. By lowering the priority of the packets flowing on the backup path, as described below, the short-lived overlap will also not cause data loss, as perceived by the receiver.

Our approach: *If router k's new upstream router, j, is on one of the current downstream interfaces for router k, then router k locally designates the upstream interface with the pending "waiting-to-send-join" state. The join does not occur until k has received a prune request from j.*

5. PIM reconfiguration with hitless switchover

After a failure has occurred, our method uses modifications to the PIM-SSM protocol to achieve the goals stated in the previous section. We now describe the assumptions of our multicast reconfiguration protocol and prove its correctness.

We first make the following consistency assumptions (later, we discuss relaxing them).

1. The weight of link (i,k) equals the weight of (k,i). This applies to either the pseudowire or PHY link.
2. Links (i,k) and (k,i) are either both up or both down.
3. No two directed least-weight paths between any two routers share a node or link. This can be always guaranteed by implementing a consistent tie-breaking rule across all routers. Combined with assumption 1, this ensures that the least-weight path from router i to router k is the reverse of the least-weight path from k to i.
4. The IGP reconfiguration process is completed in each router before a join request in response to the topology change is sent, and every router has the same view of a (non-disconnected) SPT. We use this assumption to prove the correctness of our protocol. We then relax this assumption and show that the correctness still holds.
5. The failure is such that no packets are lost immediately after all FRR reroutes are completed (otherwise "hitless" reconvergence has no context in which to be established).
6. If a layer 1 failure causes multiple, simultaneous failures of layer 3 links, receiving the corresponding LSAs at each router gives a complete view of the topology prior to SPT recomputation (i.e., a single multicast tree computation after the LSAs are received reflects the new topology).
7. Multiple (non-simultaneous) failures are separated by enough time to allow our algorithm to settle.

5.1. Major tasks and rules

The following tasks/rules describe our method.

Rule (a): *expose link failure to IGP routing even though FRR backup path is in use.*

Rule (b): *notify multicast protocol about IGP routing changes so that it can reconfigure whenever possible.* A router determines its upstream interface from its most up-to-date SPT as described in Rule (a). If this is a new upstream interface, then this router (say k) sends a join request to the upstream router (say j) *except* if router j is one of the current (i.e., prior to the failure) downstream interfaces for router k. In this latter case, router k locally designates the upstream interface with the pending "waiting-to-send-join" state. Router k will also clear all pending "waiting-to-send-join" states on an upstream interface when it receives a subsequent LSA and computes an SPT that implies a *different* upstream interface.

Rule (c): *prune old upstream interfaces only after data packets are received on the new upstream interface.*

Rule (d): *clear all pending joins when no downstream interface is left, upon reception of prune(s) from downstream routers. When a prune is received on an interface that is marked with a 'pending join', respond by issuing a join on that interface and clear the 'pending join' on that interface.* When router k receives a prune from (downstream) router j, router k prunes the interface to router j and does the following: (case d1) if router k has no remaining downstream interfaces (including receivers), then router k sends a prune message to its upstream interface and clears the "waiting-to-send-join" on any pending upstream interfaces; (case d2) else if router k has at least one remaining downstream interface (to a router other than j or receiver) and is in "waiting-to-send-join" status to router j, it sends a join request to j.

5.2. Description of the algorithm

We now describe the major steps of the protocol via an example. Consider Fig. 11:

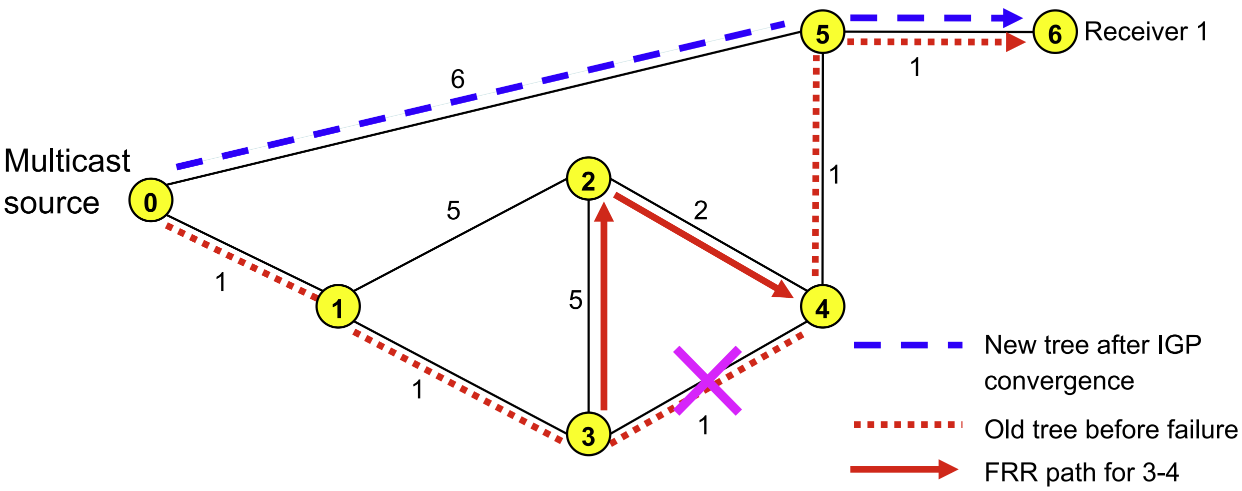


Fig. 11. Protocol convergence example.

1. Router 4 locally recognizes the failure of PHY links (3,4) and (4,3).
2. FRR re-routes traffic over pseudowire (3,4) to its backup paths; pseudowire (4,3) is similarly re-routed to its backup path (not shown). Note that pseudowire (3,4) is a link of the pre-failure multicast tree.
3. Routers 3 and 4 broadcast LSAs to all their neighboring routers indicating that all the pseudowires and PHY links between (3,4) are down (or assigns them very high link weights). Note, however, that routers 3 and 4 do not take the pseudowires down yet.
4. Upon receipt of the LSA and after a hold-down period, each router with existing downstream interfaces on the old multicast tree locally computes a SPT and determines its new upstream interface on the SPT. Recall that PIM-SSM computes paths in the reverse direction (sink-to-source) of their actual multicast packet flow. This results in router 4 needing to send a join request to router 5. However, since router 5 is also a current downstream interface of router 4, router 4 marks the interface with a "waiting-to-send-join" status. After receiving the LSA and updating SPT, Router 5 sends its join request to router 0. Since other routers with downstream interfaces (on the old tree) do not change their upstream routers, no other join requests are sent in response to the LSA update after the failure.
5. Router 0 receives the join request from router 5, adds its interface as a downstream interface, and begins to send packets to router 5.
6. When router 5 receives the first packet from router 0, it discontinues forwarding packets received from router 4 and sends a prune message to router 4.
7. Since router 4 has no remaining downstream interfaces on the new multicast tree (case d1), it sends a prune message to router 3 and clears the "waiting-to-send-join" status to router 5. However if router 4 had a receiver attached to it, this would require case d2. Router 4 would have to send a join to router 5 only after receiving a prune from router 5 for the old tree once the data starts flowing to router 5 from router 0. Subsequently, router 4 would release the "waiting-to-send-join" state and send the join to router 5. Finally, it would prune the old interface to router 3.
8. Since router 3 has no remaining downstream interfaces on the new multicast tree, it takes down both pseudowires and sends a prune request to router 1.
9. Since router 1 has no remaining downstream interfaces on the new multicast tree, it sends a prune request to router 0.

Without going through the details, once PHY link (3,4) is repaired, OSPF reconvergence is triggered when the LSAs are sent. Then, the multicast tree reconfiguration is triggered by router 5 sending a join request to router 4 and pruning router 0 once packets flow down (on the original tree) from router 4.

Some of our key changes to create a hitless version of PIM-SSM [7] are contained in Rule (b) that implements the "waiting-to-send-join" and Rule (c) that sends prunes only after it receives packets from the new upstream interface. Simple as these changes may seem, the rules have been carefully constructed to guarantee that the process converges and is hitless. For example, consider router 4 in Fig. 11. Router 4 must continue to send packets to router 5 until router 5 has established its new upstream path to the source; otherwise, receiver 1 experiences a hit. However, router 4 wants to send a join request to router 5 according to the new SPT. Under standard rules in PIM-SSM, when it sends a join request, it must stop forwarding packets to router 5, else it would form a loop. Our rules ensure that such conditions do not occur and that the receiver continues to receive packets throughout the convergence process.

Algorithm 1: Pseudo-code for our proposed non-standard multicast behavior

```
# $M_{1\times N}$ – list of $(S_i,G_i)$s the node is subscribed to
# oiflist$[S,G]$ – list of downstream interfaces for $(S,G)$
# current_iif$[S,G]$ – list of current upstream interfaces
# new_iif$[S,G]$ – list of new upstream interfaces for those $(S,G)$s on transient state
# uplink_changed$[S,G]$ – true for $(S,G)$ in transient state
# pending_join$[S,G]$ – true when waiting to send a join
# pending_join_iif$[S,G]$ – interface on which a pending join to be sent

void notify( ) # IGP routing interfaces have changed
    for $i = 1$ to $N$
    $(S,G) \leftarrow M_i$
    # obtain the new iif for $S$
    new_iif$[S,G] \leftarrow$ IGP$(S)$
    if current_iif$[S,G]$ != new_iif$[S,G]$ then
# if the new upstream node is among the current downstream ones, then delay the join
        if new_iif$[S,G] \subset$ oiflist$[S,G]$ then
            pending_join$[S,G] \leftarrow$ true
            pending_join_iif$[S,G] \leftarrow$ new_iif$[S,G]$
            return
        end if
        # indicate the transient state on $(S,G)$
        uplink_changed$[S,G] \leftarrow$ true
        join$(S,G)$
        end if
        end for

void handle-wrong-iif$(S,G,iif)$ # Data packet for $(S,G)$ was received on a wrong upstream interface iif
  # if the node is on transient state for $(S,G)$ and you have been expecting data packets on iif
  if uplink_changed$(S,G)$ AND new_iif$[S,G]$ = iif then
  uplink_changed$[S,G] \leftarrow$ false
  # send prune to the old upstream interface
  send-prune(S, G, current_iif$[S,G]$)
  # install the new upstream interface
  change-iif(S, G, current_iif$[S,G]$, new_iif$[S,G]$)
  current_iif$[S,G] \Downarrow$ new_iif$[S,G]$
  end if

void recv-prune(S, G, iface) # Data packet for $(S,G)$ was received on a wrong interface
  if pending_join$[S,G]$ AND iface = pending_join_iif$[S,G]$ then
  pending_join$[S,G] \leftarrow$ false
  uplink_changed$[S,G] \leftarrow$ true
  join$(S,G)$
  end if
# Continue the regular processing of prune
```

5.3. Pseudo code

Our cross-layer failure restoration algorithm, presented as pseudo code in Algorithm 1, involves three additions to the default operation of multicast routing.

The first addition is the **notify( )** procedure which is invoked when the underlying IGP routing protocol recalculates its SPF, which could be triggered either by detection of a failure local to the node, or reception of a link state announcement from a neighbor IGP router. Once **notify( )** is called, it checks if there are any (S,G) multicast sessions for which the upstream interface has changed as a result of the IGP recalculation. For those multicast sessions with a new upstream interface, the tree reconfiguration procedure starts by trying to join the tree via the new upstream node. If the new upstream interface is one of the current downstream interfaces, then the node moves to the *pending_join* state and must wait for that downstream interface to receive a prune message first.

The second addition is the **handle-wrong-iif (S, G, iif),** which simply checks if a data packet for multicast session (S,G) was received on the correct upstream interface or not. If a data packet was received on a wrong incoming interface, then it must be due to the fact that the node has been waiting for the data packets to arrive on the new upstream interface to which a join was sent in **notify( )**. If **handle-wrong-iif( )** indeed detects that the new data packet was received on the new upstream interface for the multicast session (S,G), then it sends a prune message to the old upstream interface for (S,G). This completes the switchover to the new tree.

The last addition is **recv_prune (S, G, iface)** which is called when a prune message is received on the interface *iface* for the multicast session (S,G). The purpose of **recv_prune( )** is to check if the multicast session (S,G) has been in the *pending_join* state due to the fact that *iface* has been among the current downstream interfaces while the algorithm wanted to send a join to it. If this is the case, then **recv_prune( )** simply moves the node from the *pending_join* state to the normal state and sends the pending join message onto interface *iface*.

5.4. Correctness proofs

There are three major claims that establish the correctness of the algorithm described.

Claim 1. *After a failure, eventually the protocol converges and the set of all links of the actual multicast tree are equivalent to the computed multicast tree (CMT). The computed multicast tree is defined as the tree theoretically computed from the new SPT and the actual multicast tree is defined to be the tree established by the routers working independently according to our rules.*

Claim 2. *Except for packet loss due to potential FRR path overlap and packets "in flight", there is no packet loss due to convergence from the old tree to the new tree.*

Claim 3. *Packet loss will be reduced due to potential FRR path overlap using a packet QoS prioritization scheme where packets sent over a pseudowire backup LSP path have lower priority than other packets.*

Proof of Claim 1. We will show that every link in the computed multicast tree (CMT) also appears in the actual multicast tree. Assumption 4 guarantees that all routers eventually calculate the same (new) SPT. Let us start at an arbitrary leaf node contained in the CMT with a receiver interfacing on it. Rule (a) guarantees that any actual upstream link is contained in the new CMT, and rule (c) guarantees that it prunes any upstream links not contained on the CMT. Recursively, as we follow the path of actual upstream links, if we encounter any upstream link not on the CMT, then this would again violate rules (a) and (c). If we repeat this process of following the path on all leaves of the CMT, we will eventually encounter every node contained in the CMT. Thus, we have shown that for every node contained in the CMT, its actual upstream links are also contained in on the CMT.

Now let us assume that there exists an actual link not contained in the CMT. We will show a contradiction in all cases. Let us follow any path of actual links downstream (starting with this link) until we encounter the first router that is either (Case 1) contained in the CMT or (Case 2) is not contained in the CMT, but has no actual downstream links. Let us designate this last downstream link on the path as x and its downstream router as k. If Case 1 holds, we have contradicted the previous paragraph by finding an upstream link, x, to router k, that is not on the SPT. If Case 2, then either (Case 2.1) link x was established by the protocol after the failure or (Case 2.2) link x existed prior to the failure. For Case 2.1, at some time after the failure, there must have existed a downstream link from router k, but that was subsequently pruned. Otherwise, router k must have a receiver interfacing on it and thus contained in the CMT. If the last downstream link was subsequently pruned from router k, this contradicts rule (d) that prunes the upstream link once all the downstream links are pruned. If Case 2.2, then as in Case 2.1, either router k must have a receiver interfacing on it and thus contained in the CMT or must have originally had a downstream link that was subsequently pruned by our protocol, which contradicts rule (d). Thus we conclude all actual links are contained in the CMT. This concludes the proof. □

Proof of Claim 2. Given assumption 5, all receivers continue to receive packets before the first join or prune request is sent. Thus, if we examine a receiver and its path at any point in time from source to receiver and then examine how each join or prune request affects this path, we see a join request along this path will not cause packet loss because of rule (c): the intermediate router must receive packets from a new upstream interface before turning off the old upstream. Rules (c) and (d) also guarantee that a prune will not be sent upstream unless (1) packets are arriving from a new upstream interface or (2) there are no more downstream interfaces. This concludes the proof. □

While we do not prove Claim 3 in a formal manner, we evaluate the performance improvement through simulation given the capability for such QoS mechanisms in routers. Fig. 12 is an example where the new tree might overlap on links (we call these “common links” (CL)) of the FRR path of the failed pseudowire. The dotted red tree is the original tree before the failure of link 3–4. When link 3–4 fails, the pseudowire switches to FRR backup path 3–2–4, which now overlaps with link 2–4, which is part of the reconfigured tree. The inefficiency is because the same multicast packets will be carried over CL 2–4 more than once.

A possible way of solving the congested CL problem is to assume that routers can prioritize packets. In Fig. 12, when link 3–4 fails, the IGP is then informed about the failure. Router 3 forwards packets on the backup path and marks them as lower priority packets. As the join process evolves, router 4 will send a join request to router 2 and then router 2 sends a join request to router 1. When router 1 begins to transmit to router 2, these packets will eventually arrive at router 2 as will packets from the backup path 3–2–4. However, router 2 does not “see” any multicast packets from router 3 at layer 3, because the FRR path is tunneled through at Layer 2. However, Layer 2 forwarding protocols will recognize that the packets from router 3 have lower priority than those from router 1. If the link (2–4) has insufficient capacity to handle twice the rate of the multicast flow, then the packets from router 3 (that are marked at a lower priority at router 2) will experience loss. However, because of rule (c) described earlier, the first packet from router 2 that is received at router 4 will cause the packets from router 3 (via the 3–4 pseudowire that routes over the backup FRR path 3–2–4) to be ignored. This will

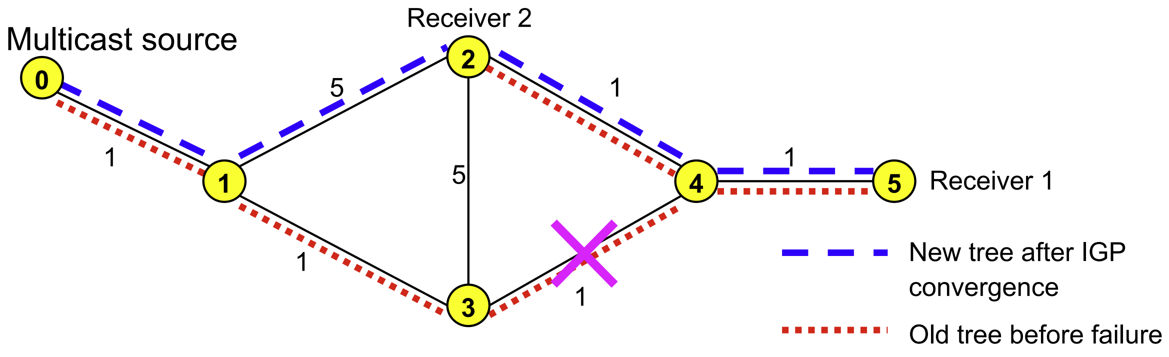


Fig. 12. Example of FRR overlap.

result in little, if any, packet loss: any such loss will be due to the switching mechanism to implement rule (c). Therefore, as stated earlier, this is a function of router implementation, rather than a mathematical proof.

5.5. Relaxation of assumptions for practical operation

Now that we have established the basic validity of our approach, we discuss potential relaxations of the assumptions 1–7. We generally do not recommend relaxing assumptions 1 through 3. In particular, without assumption 3, the CMT becomes ambiguous. Finally, assumptions 6–7 are mostly provided to give clear context for the proofs. However, if one examines these proofs, they demonstrate that we can start from any actual multicast tree, even one that is in transition to a target state, and the process will converge to the eventual CMT once the views of the SPT by all routers settle. Thus, we feel assumptions 6–7 can be relaxed without negative impact.

The strongest assumption is 4. In reality, there may be situations where we need to relax this assumption when the variation of completion time of the IGP reconfiguration process across the routers is large. Then, it is possible a router may send a join request to an upstream interface before the upstream router has computed its new SPT. This would result in the process possibly proceeding up the wrong tree. However, it is easy to see that any router that had chosen the wrong upstream interface will, after receiving the LSA, updating its SPT and, following application of rule (a), recalculate its upstream interface per the new SPT and then send a join request to the correct upstream interface; therefore the process will eventually converge to the CMT. As demonstrated in Claim 2 earlier, no packets should be lost as this tree building process converges.

Another complication of relaxing assumption 4 is that because of potential “waiting-to-send-join” states, we need to demonstrate that the protocol does not “deadlock”. Suppose we enter a state where a cycle of routers, $(i_1, i_2, \ldots, i_n, i_{n+1})$, is deadlocked, and router i_{k-1} is “waiting to send a join” to router i_k and $i_{n+1} = i_1$. That is, router i_k is upstream of router i_{k-1}. Once the SPT has been updated consistently in all routers, rule (b) implies that these upstream links are chosen so that they are contained in the SPT in a path towards the source. However, they form a cycle, which contradicts the definition of an SPT.

Another option to is to impose a timer to insure the SPTs have all been updated before the first join is sent (thus ensuring assumption 4 is met); however, because there so many complex timers and interactions among timers and protocols already in router-based networks (and in multicast networks) we suggest that this latter option be avoided.

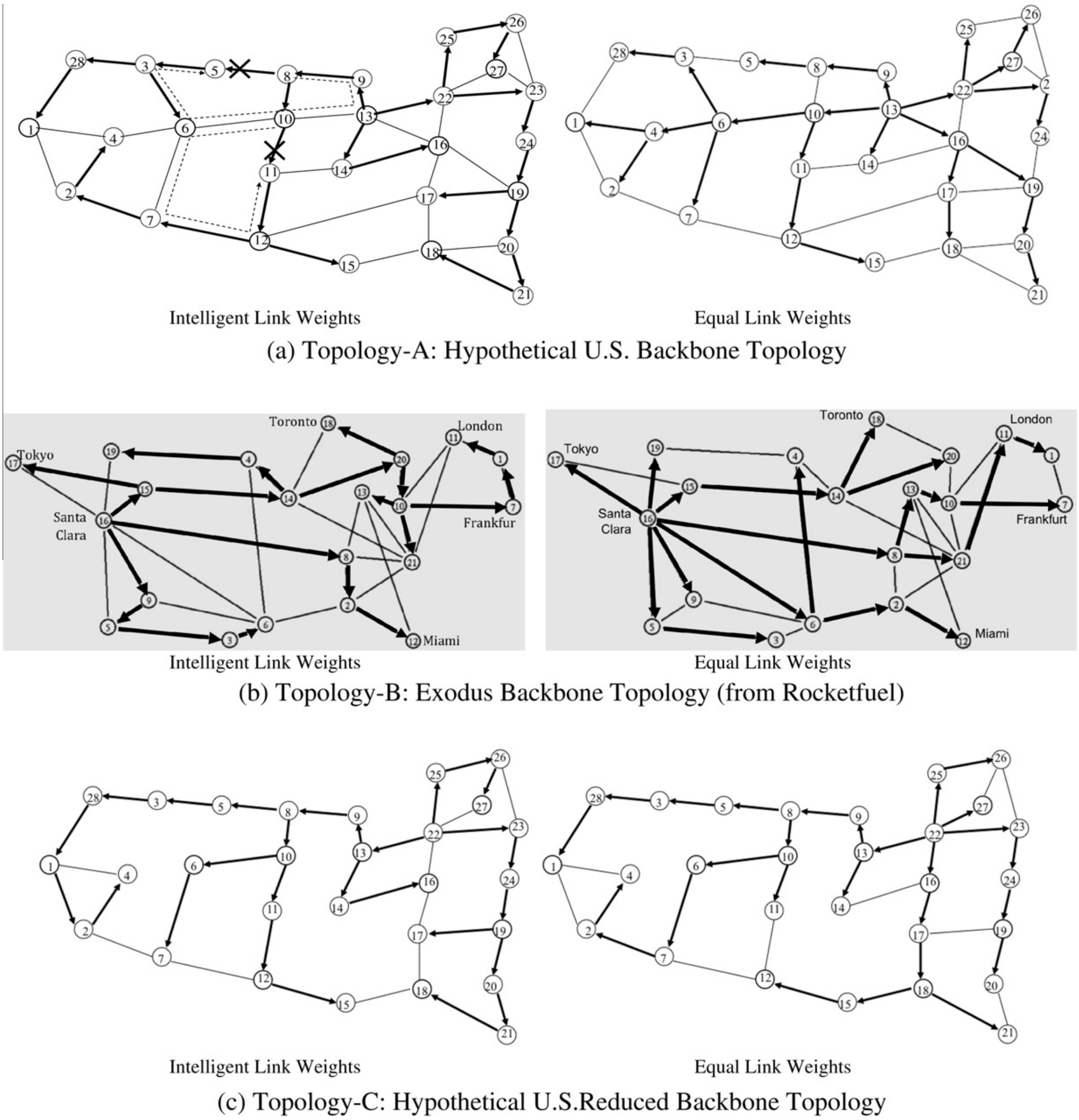
Another issue regarding relaxing assumption 4 is that due to inconsistent completion of SPT computations among the routers, it is possible for a router to receive a join request from a downstream router but then needs to send an (upstream) join request to the same router. The current PIM-SSM protocol does NOT admit the join request from the downstream router if it is not in its current SPT. Reconfiguration of the SPT or a soft-state refresh message is used to eventually correct this situation. This requirement can be accommodated in our modifications and convergence to the CMT can be established. However, we make the observation that with our new rule (b), indeed the join request from the router can be allowed (even if not on the SPT of the upstream router), but the upstream join request back to that router is put into the “waiting-to-send-join” state. Once the router updates its SPT and the correct upstream interface is calculated, the “waiting-to-send-join” state will be cleared and it will converge to the correct configuration. Strictly speaking our new protocol does not require soft state refreshes to rectify this situation. However, as various error conditions occur and violate our assumptions, it is recommended that current PIM-SSM soft-state rules be retained to handle these situations.

6. Simulation results

We evaluated our cross-layer failure restoration framework using ns-2 [28]. Specifically, our experiments aim to reveal how much *packet loss* occurs during switchover after failure. We used MPLS [29] to provide FRR support at the routers and implemented PIM-SSM over MPLS in ns-2. Regular ns-2 implementations of multicast routing and MPLS do not allow them to be used simultaneously. So, we made significant revisions to the packet classifier structure of ns-2 (both for multicast and MPLS). We also revised the PIM-SSM implementation of ns-2 to make it a fully distributed protocol, rather than the centralized computation of multicast tree in the existing PIM-SSM implementation. We implemented our hitless switchover protocol in this distributed PIM-SSM implementation, and made it optional to be aware of IGP routing changes, which we use in our comparative evaluation. For the IGP routing, we used an OSPF implementation in ns-2 [12]. We made the necessary changes to the protocols so that OSPF is informed about link failures (even though MPLS forwards the data packets on the FRR path immediately after the failure) and PIM-SSM is informed about OSPF routing changes.

6.1. Experimental setup

We performed simulations with three different topologies: (1) Topology-A is a hypothetical US backbone topology (with 28 routers and 45 links) shown in Fig. 13(a), with the multicast source set at router 13 (approximately the center of the network), (2) Topology-B is the “Exodus” topology from Rocketfuel [33] (with 21 routers and 36 links) shown in Fig. 13(b), with the multicast source set at router 16, which corresponds to its Point-of-Presence (PoP) in Santa Clara, and (3) Topology-C, a reduced version of Topology-A (with 28 routers and 36 links) shown in Fig. 13(c), with the multicast source set at router 22. We assigned equal capacities to the links but set their propagation delays proportional to approximate physical distances. The multicast source generated UDP traffic with a packet size of 500B. The rate of the multicast traffic was 70% of the capacity of the links. We budgeted for 120 ms of buffer time, i.e., a link with a 100 Mb/s capacity had a buffer size of 3000 packets. We used default OSPF timer



Intelligent Link Weights

Equal Link Weights

(a) Topology-A: Hypothetical U.S. Backbone Topology

Intelligent Link Weights

Equal Link Weights

(b) Topology-B: Exodus Backbone Topology (from Rocketfuel)

Intelligent Link Weights

Equal Link Weights

(c) Topology-C: Hypothetical U.S.Reduced Backbone Topology

Fig. 13. Experimental topologies.

settings, e.g., *spfDelayTime* = 5 s, and *spfHoldTime* = 10 s, and a relatively short rejoin interval of 30 s for PIM-SSM. This timer is normally set to several minutes, which would make the improvements resulting from our protocol better than is reported here.

Our simulation scenarios consisted of failing link(s) in the topology and observing the multicast data traffic during re-convergence of IGP and the multicast tree. We measured (a) packets lost at each receiver until the failure restoration is complete, and (b) the time taken to complete failure restoration. These two metrics, (a) and (b), quantify the ''hit'' caused by each protocol. We examine both the average and the maximum for the hit time and packet loss over the length of each experiment.

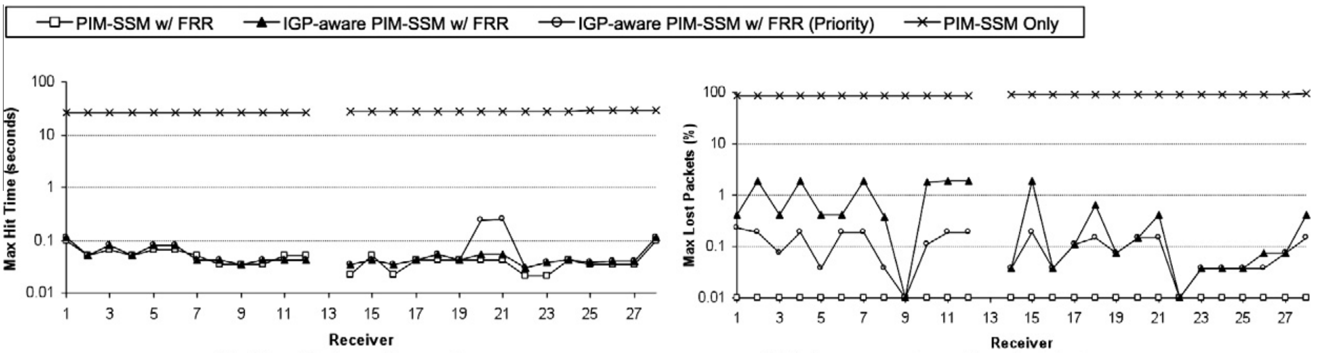
We compared four scenarios: (i) PIM-SSM only (i.e., no FRR support and no IGP-aware tree reconfiguration), (ii) PIM-SSM w/ FRR (i.e., FRR support exists but no IGP-aware tree reconfiguration), (iii) IGP-aware PIM-SSM w/ FRR (i.e., both FRR support and IGP-aware tree reconfiguration are used – **our proposal**), and (iv) IGP-aware PIM-SSM w/ FRR with data over the backup path being forwarded at higher priority (as per Claim 3 of Section 5.4, an enhancement to our proposal). For ''IGP-aware PIM-SSM w/ FRR'' with priority, we divided the link buffer equally between the two priority classes. For single failure cases, we evaluate the evolution of the multicast reconfiguration protocol for a period equal to the time for the PIM-SSM rejoin timer (i.e., 30 s) after a failure occurs. As described below, for multiple failure cases, we evaluate the performance until the failed links are repaired and IGP and multicast protocols have re-converged completely. Note that the PIM-SSM convergence time is normally 3–4 times longer than the rejoin timer, as all unnecessary downstream branches will have to be pruned. However, we are conservative and use only the rejoin time in our comparison, since PIM-SSM guarantees that multicast traffic will flow to all receivers after the timer expires. We assigned OSPF link weights using two approaches: (a) by using the algorithm in [10] which assures that single failures do not cause any traffic overlap for the "PIM-SSM w/ FRR" case ("**Intelligent Link Weights**" in the figures), and (b) by assigning identical weights to all links ("**Equal Link Weights**" in the figures). When selecting the FRR paths, we choose the least-cost non-overlapping path that would restore the failure. This experimental setup favors the "PIM-SSM w/ FRR" case and our comparison of the four cases reveals the extra improvement our multicast reconfiguration protocol attains.

6.2. Single failures

To compare the four protocols for a single link failure, we failed each link in the multicast tree one by one and measured the maximum time that each of the receivers experience a hit (i.e., disconnect) as well as the maximum percentage of packets lost for each receiver during a 30 s time interval, which is the maximum failure restoration time based on the PIM-SSM rejoin timer of 30 s. Figs. 14–19 show the results of these single failure experiments.

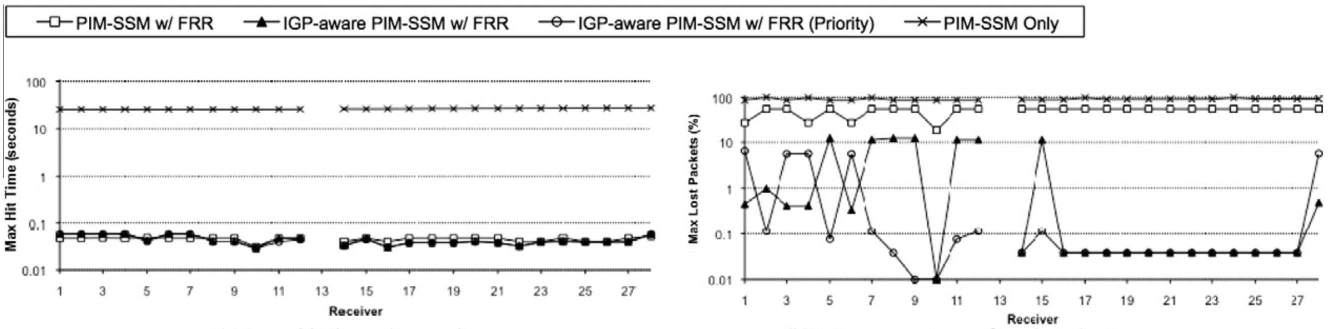
Figs. 14, 16 and 18 show the results for the cases when link weights are set intelligently, and we observe with just PIM-SSM the maximum time that a receiver experiences a hit can be quite large (10 s of seconds). When the "IGP-aware PIM-SSM w/FRR" schemes (both with and without prioritization of FRR traffic) are employed, the receivers experience a significantly smaller (around 100 ms) maximum hit time, which is also the case for "PIM-SSM w/ FRR". In these simulation scenarios, the OSPF link weights are intelligently set, so it is expected that PIM-SSM w/FRR would perform the best since overlaps are avoided due to such a link weight setting. However, the main point of these simulation experiments in Figs. 14, 16 and 18 is to show that our cross-layer mechanisms (i.e., IGP-aware PIM-SSM) do not add any notable impairment to received data traffic at the receivers.

We also examined the case where all the links have equal weights. Figs. 15, 17 and 19 show the results for the equal link weights cases for Topology-A, Topology-B and Topology-C, respectively. As shown in Figs. 15(a), 17(a) and 19(a), the relative performance of the four protocols does not change significantly with respect to the maximum hit time. But, the maximum of the percentage of lost packets grows dramatically (Figs. 15(b), 17(b) and 19(b)) for "PIM-SSM w/FRR" since the link weights are not set to avoid any overlaps. For the two protocols using IGP-aware cross-layer reconfiguration, the maximum



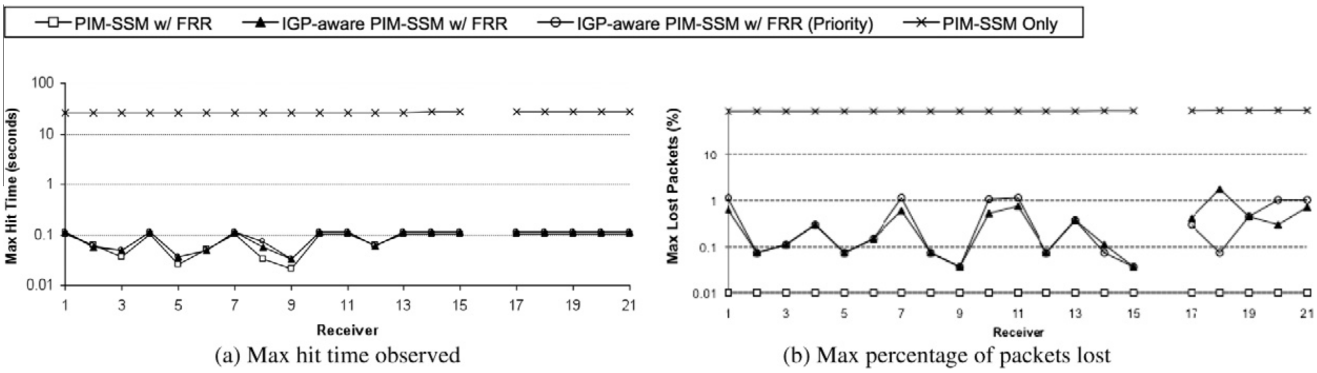
(a) Max hit time observed (b) Max percentage of packets lost

Fig. 14. Topology A with intelligent link weights: results for single failure restoration.



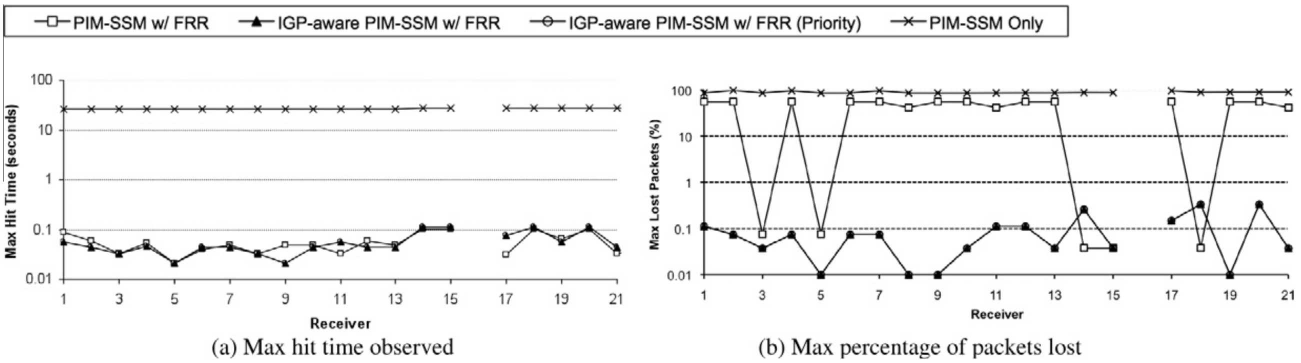
(a) Max hit time observed (b) Max percentage of packets lost

Fig. 15. Topology A with equal link weights: results for single failure restoration.



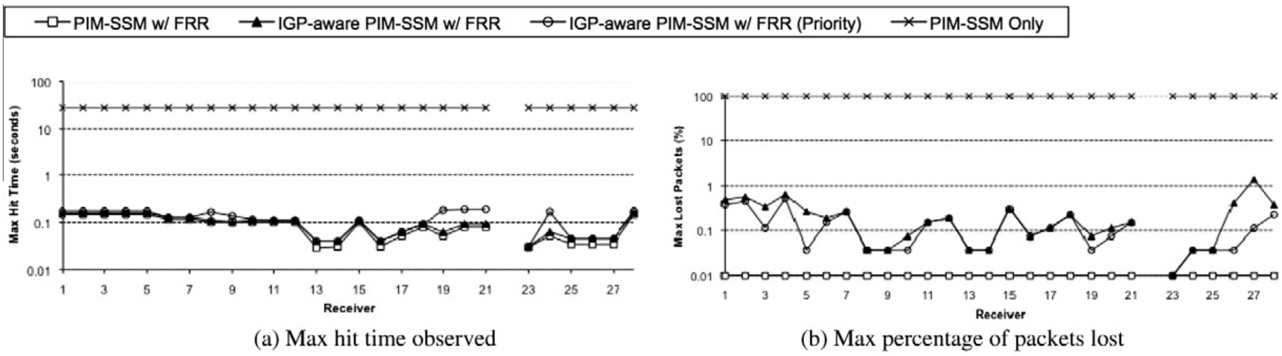
(a) Max hit time observed (b) Max percentage of packets lost

Fig. 16. Topology B with intelligent link weights: results for single failure restoration.



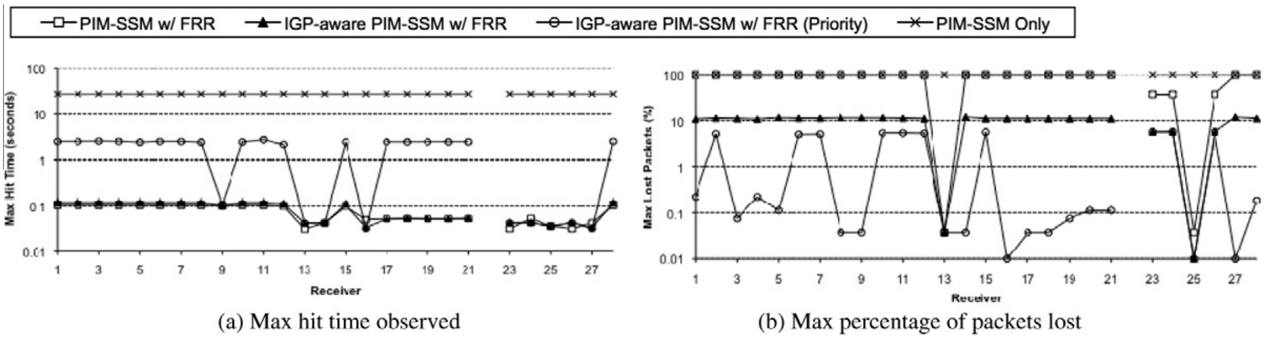
(a) Max hit time observed (b) Max percentage of packets lost

Fig. 17. Topology B with equal link weights: results for single failure restoration. The effect of prioritized FRR traffic is not noticeable.



(a) Max hit time observed (b) Max percentage of packets lost

Fig. 18. Topology C with intelligent link weights: results for single failure restoration.



(a) Max hit time observed (b) Max percentage of packets lost

Fig. 19. Topology C with equal link weights: results for single failure restoration.

percentage of packets lost stays roughly the same in comparison to the intelligent link weights cases. Further, prioritization of the FRR traffic helps reduce the packet loss for most receivers. As shown in Fig. 19(b), the benefit of the prioritization of the FRR traffic is more pronounced for Topology-C, which has fewer backup links.

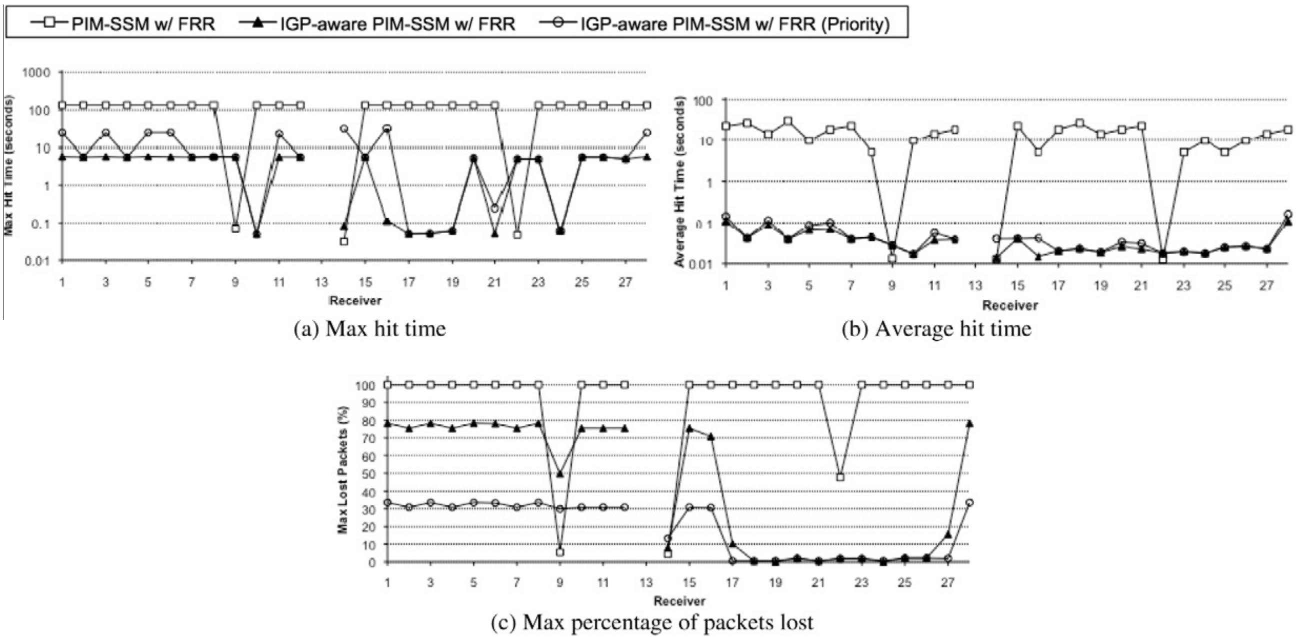
Overall, the results for single failures show that by using IGP-aware PIM-SSM, there is less of a requirement for the intelligent weight setting algorithm. However, when the link weights are set equally, the IGP-aware mechanisms help significantly in remedying the performance loss due to temporary overlaps during the period of switchover to the new tree. Further, the IGP-aware mechanism to reconfigure PIM-SSM after FRR consistently achieves little or no packet loss beyond what PIM-SSM with FRR achieves. Indeed, the only packet loss that takes place in our scheme is due to the packets lost in transit on the failed link, and the potential congestion on the FRR path during the transient period while our scheme reconfigures the PIM tree. During this transient period, we may observe the common congested link (CL) situation, where lowering the priority of the traffic sent over the backup path reduces loss further.

6.3. Double failures

The real benefit of IGP-aware multicast reconfiguration is when there are multiple overlapping failures in the network. To understand the significance of this benefit, we simulated the case where there are overlapping failures of two links. The first link fails at the 10th second and the second link at the 50th second of the simulation. Then, the first and second links are repaired at the 200th and 250th s, respectively. This experiment setup ensures that the time between the two failures is larger than the typical time taken by PIM-SSM (30 s) to have a branch of the multicast tree that was disconnected by the first failure rejoined to the multicast tree. We further set failure recovery times such that there is enough time for the PIM-SSM to prune all unnecessary branches, which might have created during the reconfiguration process. As a result, we have set the time between a link failure and its repair to be larger than 120 s to ensure completion of the convergence of the protocols. We measured performance from the time of the first failure until both the links are repaired and IGP and multicast protocols have re-converged completely. The parameters chosen for the simulation experiments were selected such that the time of the overlap of failures was larger than the minimum required to assure protocol convergence. If the time that failures overlap in practice is longer than these values, the improvement with our IGP-aware techniques will be even more significant than shown in our results.

We simulated every possible combination of 2-link failures involving any of the link pairs causing at least one reconfiguration of the PIM tree. Fig. 13(a) shows an instance of the multicast tree for a particular double link failure. Some of these link pairs actually cause some receivers to become disconnected. So, to compare the protocols, we examine the "average" hit time and packet loss in addition to the "maximum" hit time and packet loss. We do not evaluate the "PIM-SSM only" case under double failures as it performs much worse than the other three cases even for single failures.

Figs. 20–25 show the impact of double failures for various intelligent or equal link weight settings on the topologies. These results clearly show that IGP-aware multicast reconfiguration handles multiple failures much better than the FRR-only approach. Figs. 20(a), 22(a) and 24(a) show that the IGP-aware cases can restore double failures within 5 s while PIM-SSM with FRR can take more than 100 s (based on the *maximum* hit time) when the link weights are set intelligently. The link weight setting method does not change this relative performance of the alternatives, as observed in Figs. 21(a), 23(a) and 25(a). IGP-aware protocols manage to keep maximum hit time under 5 s in

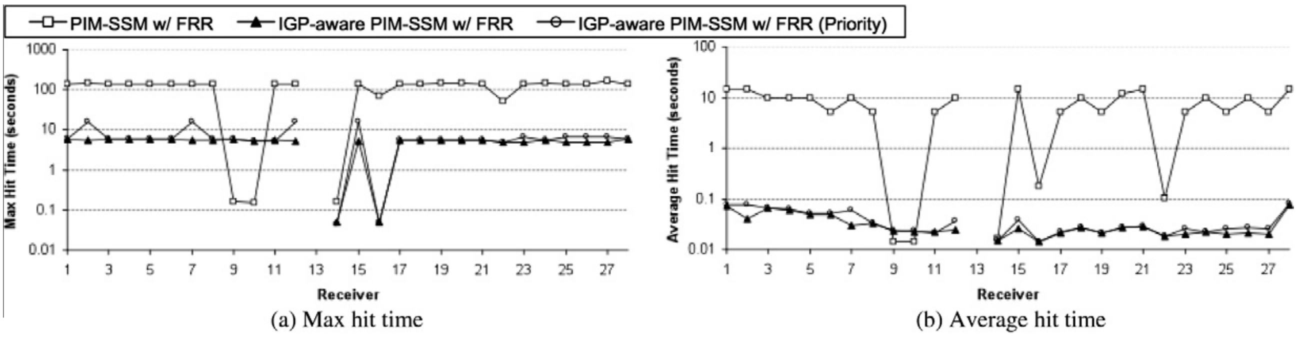


(a) Max hit time

(b) Average hit time

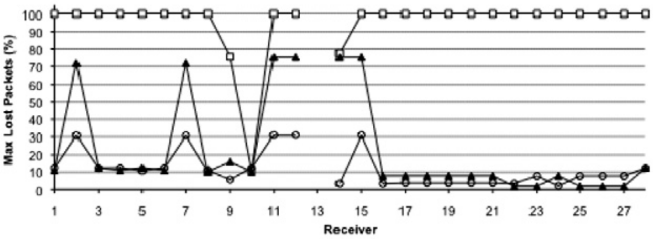
(c) Max percentage of packets lost

Fig. 20. Topology A with intelligent link weights: results for double failure restoration.



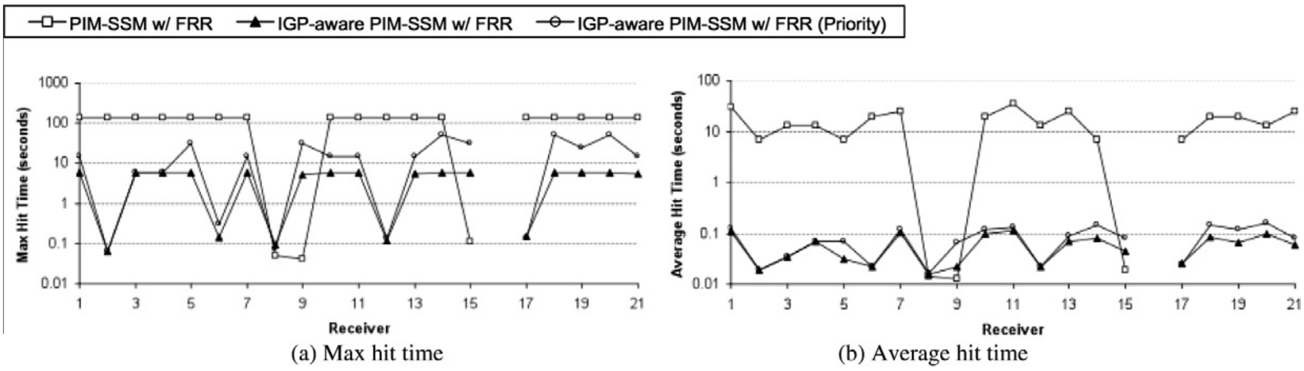
(a) Max hit time

(b) Average hit time



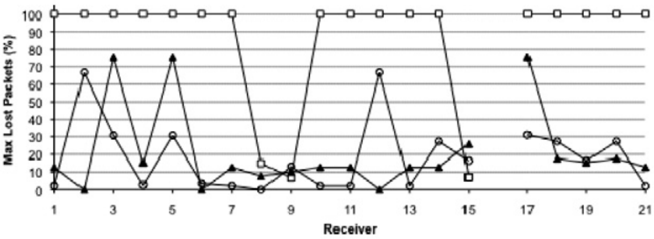
(c) Max percentage of packets lost

Fig. 21. Topology A with equal link weights: results for double failure restoration.



(a) Max hit time

(b) Average hit time

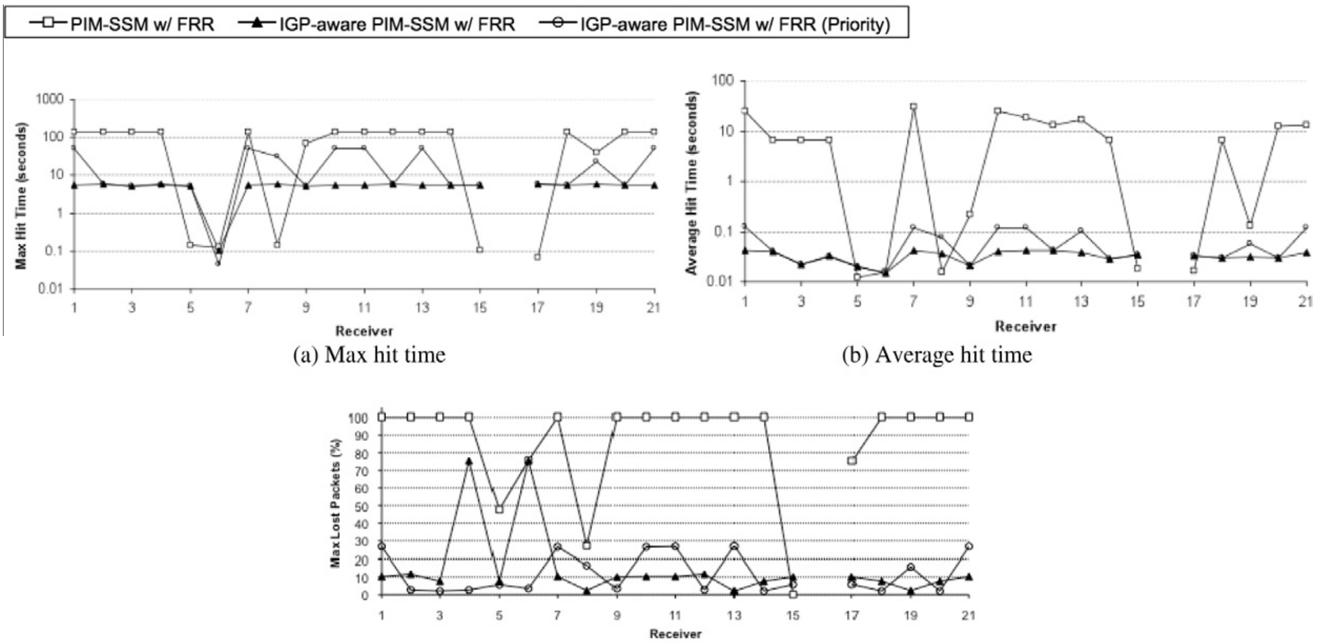


(c) Max percentage of packets lost

Fig. 22. Topology B with intelligent link weights: results for restoration from double failures.

Topology-C as seen from Figs. 23(a) and 25(a), even though it has significantly fewer backup links in comparison to Topology-A. Note that the reason why IGP-aware protocols still take up to 5 s to restore is because the second failure may have an FRR path that utilizes the first failed link. This may only be restored by repairing the link or using a dynamic reconfiguration of the FRR paths that may be impacted by the failure of the first link, as suggested in [30]. Without such a dynamic reconfiguration, control packets (e.g., prune) may also be lost, resulting in longer restoration times.

Figs. 20(b), 21(b), 22(b) and 23(b) show the *average* performance impact of double failures on each receiver. The IGP-aware protocols achieve less than 100 ms average recovery time while PIM-SSM with FRR spends more than 10 s for most of the receivers. Even when there are a very limited set of backup links available, as in Topology-C, the IGP-aware protocols can keep the average hit time under 500 ms as shown in Figs. 24(b) and 25(b). These results on average hit time clearly show that IGP-aware protocols provide significantly superior performance in handling double failures.

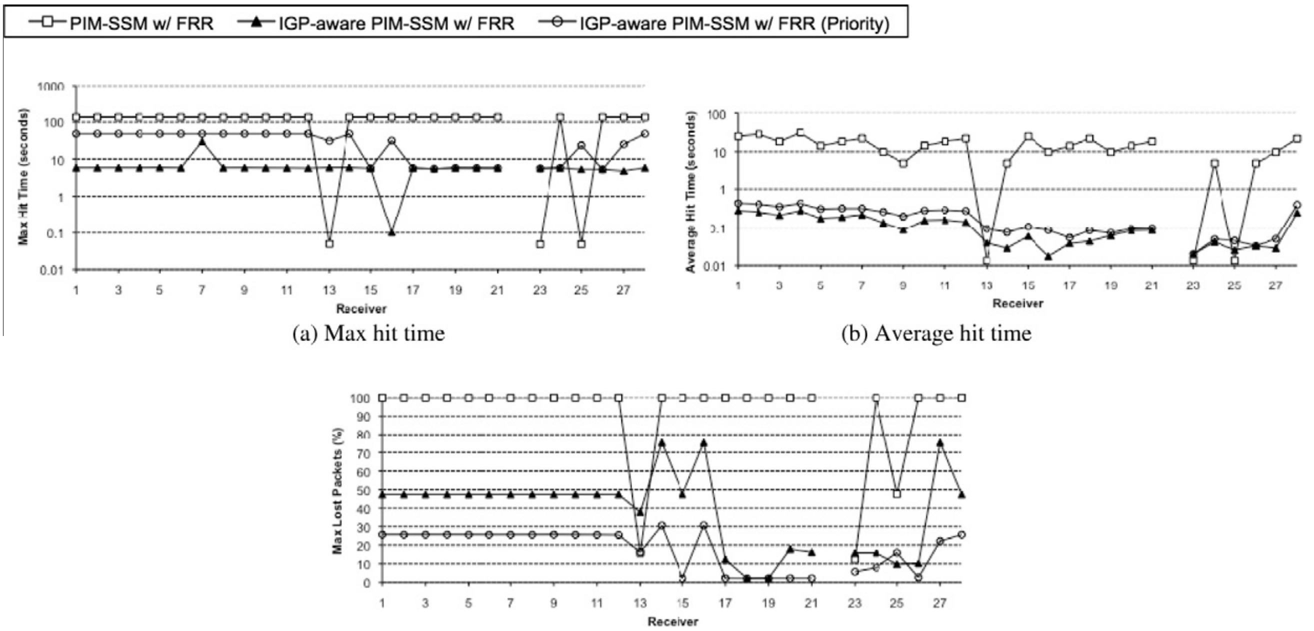


(a) Max hit time

(b) Average hit time

(c) Max percentage of packets lost

Fig. 23. Topology B with equal link weights: results for restoration from double failures.



(a) Max hit time

(b) Average hit time

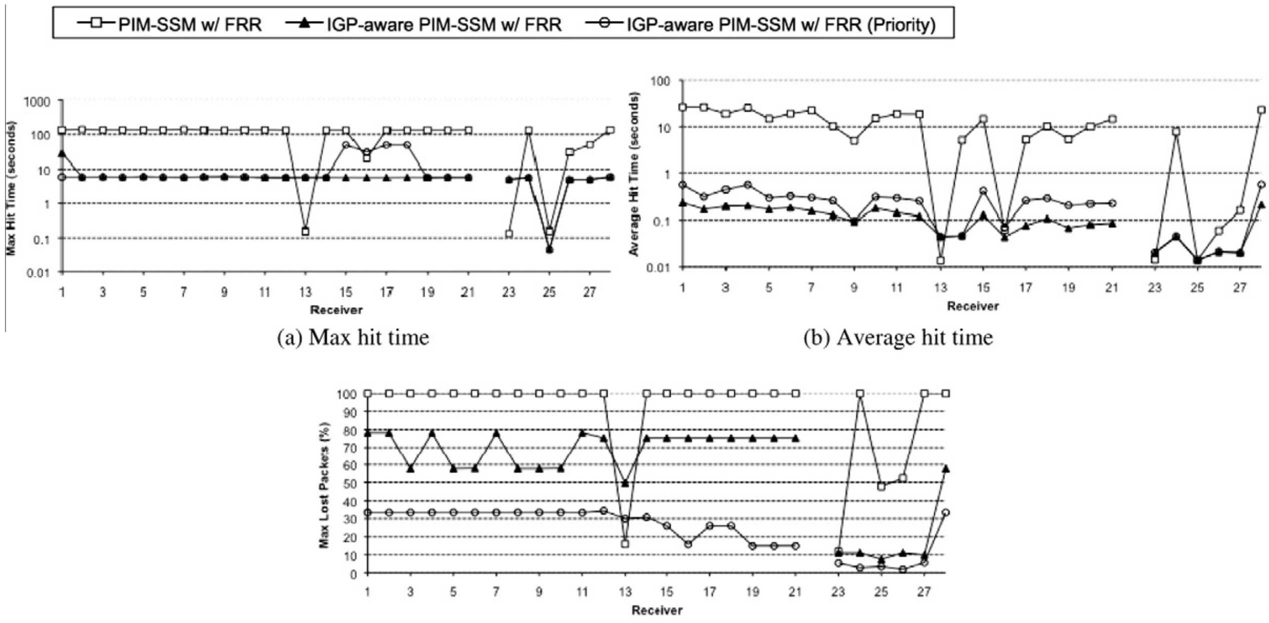
(c) Max percentage of packets lost

Fig. 24. Topology C with intelligent link weights: results for restoration from double failures.

In terms of packet loss, as shown in Figs. 20(c), 21(c), 22(c), 23(c), 24(c), and 25(c), PIM-SSM with FRR can lose up to 100% of the packets while the IGP-aware protocols may lose up to 80% of the packets. However, in terms of the average number of packets lost, our experiments showed that PIM-SSM with FRR loses about 10% of the packets while the IGP-aware protocols lose less than 0.1% of the packets during the restoration from a double failure. Again, the method of choosing link weights does not change this relative performance noticeably. For Topology-C that has fewer backup links, prioritization of FRR traffic helps significantly (Figs. 23(c) and 25(c)) due to the fact that packet loss caused by overlaps are more likely.

7. Summary

The tight QoS constraints of distributing real-time multimedia require rapid restoration. Production networks rarely shorten their timers enough to achieve sub-second IGP convergence because of the potential of false alarms. The standard solution is to use layer-2 fast reroute (FRR) based restoration. However, layer-2 FRR hides the backup



(a) Max hit time

(b) Average hit time

(c) Max percentage of packets lost

Fig. 25. Topology C with equal link weights: results for restoration from double failures.

path from the IGP protocol. If the FRR backup path is left to stay up during the period of the link failure, this can have a negative impact if a subsequent failure occurs. It also requires very carefully chosen backup paths with intelligent weight setting to avoid link congestion due to potential path overlap.

We proposed protocol modifications to cure these shortcomings. Our methodology makes the IGP layer aware of the failure, notifies the multicast agent of the routing changes, re-converges the network quickly to the new multicast tree, and takes down the FRR backup path, all in a "hitless" manner. When the failure is subsequently repaired, it restores the normal routing, again in a "hitless" manner. Our method quickly re-converges the network and has the added advantage of working well even without intelligent weight setting for the backup path. We demonstrate the efficacy of our schemes by running ns-2 simulations over two different network topologies. As an example of possible improvements, the average recovery time on double failures (which can be almost 20% of all link failures) which is more than 10 s for most of the receivers for PIM-SSM reduces to 100 ms with our enhancements.

Acknowledgement

Dr. Yuksel acknowledges partial support from NSF under awards 0721600 and 0627039.

References

[1] <http://www.sbc.com/gen/press-room?pid=5838>.
[2] <http://www22.verizon.com/FiosForHome/Channels/fios/FiosTV_comingsoon.aspx>.
[3] <http://www.iptvnews.net>.
[4] <http://www.sbc.com/Common/files/pdf/IPvideonetwork.pdf>.
[5] A. Adams, J. Nicholas, W. Siadak, "Protocol Independent Multicast – Dense Mode (PIM-DM): Protocol specification (revised)," IETF RFC 3973, 2005.
[6] D. Estrin (Ed.), "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol specification, IETF RFC 2362, 1998.
[7] S. Bhattacharyya (Ed.), "An overview of Source-Specific Multicast (SSM)," IETF RFC 3569, 2003.
[8] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching architecture," IETF RFC 3031, 2001.
[9] P. Pan, G. Swallow, A. Atlas (Eds.), "Fast reroute extensions to RSVP-TE for LSP tunnels," IETF RFC 4090, 2005.
[10] R. Doverspike, G. Li, K. Oikonomou, K.K. Ramakrishnan, D. Wang, "IP Backbone Design for Multimedia Distribution: Architecture and Performance," in: Proceedings of IEEE INFOCOM, April 2007.
[11] S. Kini, S. Ramasubramanian, A. Kvalbein, A.F. Hanse, "Fast Recovery from Dual Link Failures in IP Networks," in: Proceedings of IEEE INFOCOM, 2009.
[12] M. Goyal, K.K. Ramakrishnan, W.-C. Feng, "Achieving faster failure detection in OSPF networks", in: Proceedings of IEEE ICC, May 2003.
[13] A. Mahimkar et.al., "Towards Automated Performance Diagnosis in a Large IPTV Network", in: Proceedings of ACM SIGCOMM, 2009.
[14] M. Cha, W.A. Chaovalitwongse, Z. Ge, J. Yates, S. Moon, "Path protection routing with SRLG constraints to support IPTV in WDM mesh networks," in: Proceedings of IEEE Global Internet Symposium, 2006.
[15] K. Nahrstedt, R. Steinmetz, Multimedia fundamentals, Media Coding and Content Processing, second ed., vol. 1, Prentice Hall, 2002.
[16] B. Quinn, K. Almeroth, "IP multicast applications: Challenges and solutions," IETF RFC 3170, 2001.
[17] D. Velten, R. Hinden, J. Sax, "Reliable Data Protocol (RDP)," IETF RFC 908, 1984.
[18] M. Cha, S. Moon, C.-D. Park, A. Shaikh, "Placing relay nodes for intra-domain path diversity," in: Proceedings of IEEE INFOCOM, 2006.
[19] G. Li, D. Wang, R. Doverspike, Efficient distributed MPLS P2MP fast reroute, in: Proceedings of IEEE INFOCOM, 2006.
[20] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin (Eds.), "Resource ReSerVation Protocol (RSVP) – version 1 functional specification," IETF RFC 2205, 1997.
[21] C.-C. Wen, C.-S. Wu, K.J. Chen, Centralized Control and Management Architecture Design for PIM-SM Based IP/MPLS Multicast Networks, in: Proceedings of IEEE GLOBECOM, 2007.
[22] T. Speakman et al., "PGM Reliable Transport Protocol Specification," IETF RFC 3208, 2001.
[23] P. Francois, C. Filsfils, J. Evans, O. Bonaventure, Achieving sub-second IGP convergence in large IP networks, ACM SIGCOMM Computer Communication Review 35 (3) (2005) 35–44.
[24] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, C.-N. Chuah., "Proactive vs. reactive approaches to failure resilient routing," in: Proceedings of IEEE INFOCOM, March 2004.
[25] J. Moy, OSPF Anatomy of an Internet Routing Protocol, Addison Wesley, 2000.

[26] K. Oikonomou, R. Sinha, R. Doverspike, Multi-layer network performance and reliability analysis, International Journal of Interdisciplinary Telecommunications and Networking 1 (3) (2009) 1–30.

[27] W. Tan, A. Zakhor, Video multicast using layered FEC and scalable compression, IEEE Transactions on Circuits and Systems for Video Technology 11 (3) (2001) 373–386.

[28] The Network Simulator, ns-2. Available from: <http://www.isi.edu/nsnam/ns>.

[29] G. Ahn, W. Chun, "Design and implementation of MPLS network simulator supporting LDP and CR-LDP," in: Proceedings of IEEE ICON, 2000.

[30] A. Todimala, K.K. Ramakrishnan, R.K. Sinha, "Cross-layer Reconfiguration for Surviving Multiple-link Failures in Backbone Networks", in: Proceedings of OFC 2009, San Diego, March 2009.

[31] M. Yuksel, K.K. Ramakrishnan, R.D. Doverspike, R. Sinha, G. Li, K. Oikonomou, D. Wang, "Cross-Layer Techniques for Failure Restoration of IP Multicast with Applications to IPTV", in: Proceedings of IEEE International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 2010.

[32] A. Markopoulou, Y. Ganjali, G. Iannaccone, S. Bhattacharyya, C.-N. Chua, C. Diot, Characterization of failures in an operational IP backbone network, IEEE/ACM Transactions on Networking 16 (4) (2008).

[33] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with Rocketfuel, in: Proceedings of ACM SIGCOMM, 2002.

Murat Yuksel is currently an Assistant Professor at the CSE Department of The University of Nevada - Reno (UNR), Reno, NV. He was with the ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY as a Postdoctoral Research Associate and a member of Adjunct Faculty until 2006. He received a B.S. degree from Computer Engineering Department of Ege University, Izmir, Turkey in 1996. He received M.S. and Ph.D. degrees from Computer Science Department of RPI in 1999 and 2002 respectively. His research interests are in the area of computer communication networks with a focus on protocol design, network economics, wireless routing, free-space-optical mobile ad hoc networks (FSO-MANETs), and peer-to-peer. He is a member of IEEE, ACM, Sigma Xi, and ASEE.

Kadangode K. Ramakrishnan is a Distinguished Member of Technical Staff at AT&T Labs-Research in Florham Park, New Jersey. He joined AT&T Bell Labs in 1994 and has been with AT&T Labs. Research since its inception in 1996. Prior to 1994, he was a Consulting Engineer and Technical Director in Networking at Digital Equipment Corporation. Between 2000 and 2002, he was at TeraOptic Networks, Inc., as Founder and Vice President. His current research interests are in networking and communications, including congestion control, multimedia distribution, content dissemination and problems associated with large scale distributed systems. He has published over 100 papers and has over 90 patents issued. He is an IEEE Fellow and an AT&T Fellow, recognized for his work on congestion control and VPN services. His contributions on congestion control, channel access protocols (for Ethernet and Cable), network interfaces, operating system support for network I/O, VPN services and IP Telephony have been adopted and implemented in the industry.

K.K. has an MS degree from the Indian Institute of Science (1978), an MS (1981) and Ph.D. (1983) in Computer Science from University of Maryland, College Park. K.K. has been on the editorial board of the IEEE/ACM Transactions on Networking and IEEE Network Magazine and has been a member of the National Research Council Panel on Information Technology for NIST. He has participated in numerous standards bodies working on communication networks.

Robert D. Doverspike (IEEE Fellow) received his undergraduate degree from the University of Colorado and Masters and Ph.D. degrees from Rensselaer Polytechnic Institute (RPI). He began with Bell Labs in 1979 and, upon divestiture of the Bell System, went to Bellcore (now Telcordia). In 1997 he returned to AT&T Labs (Research) where he is now Executive Director of Network Evolution Research. Dr. Doverspike has made extensive contributions to the field of optimization of multi-layered transmission and switching networks and pioneered the concept of packet transport in metro and long distance networks. He also pioneered work in spearheading the deployment of new architectures for transport and IP networks, network restoration, and integrated network management of IP-over-optical-layer networks. He has over 1000 citations to his books and articles (Google scholar, as of March 2010) over diverse areas such as Telecommunications, Optical Networking, Mathematical Programming, IEEE Communications Society, Operations Research, Applied Probability, and Network Management. Dr. Doverspike holds many professional leadership positions and awards, such as INFORMS Fellow, IEEE Fellow, member of Optical Society of America (OSA), co-founder of the INFORMS Technical Section on Telecommunications, member of OFC subcommittee, editor of Elsevier eREF Optical Networks series, steering committee of Design of Reliable Communications Networks (DRCN), and associate editor of the Journal of Heuristics.

Rakesh K. Sinha is a Lead Member of Technical Staff in the Network Evolution Research Departmet of AT&T Labs – Research. Prior to joining AT&T, he worked at the routing and signaling group of Ciena CoreDirector switches and before that at the Networking Research Department of Lucent Bell Laboratories. He received his B.Tech. (Computer Science) from Indian Institute of Technology, Kanpur (India) and his Ph.D. (Computer Science) from University of Washington, Seattle. His research interests are in the areas of Networking and Algorithms.

Guangzhi Li received the M.S. and Ph.D. degrees in computer science from the College of William and Mary, Williamsburg, VA. He is a principle member of technical staff researcher at AT&T Labs-Research. His research interests include IP-based control plane for optical networks, optical layer restoration/protection schemes and algorithms, network simulation and performance evaluation as well as network related applications. He has filed about 20 patents and has published more than 50 research papers in journals and conferences.

Kostas N. Oikonomou is a Principal Member of Technical Staff in the Network Evolution Research Department of AT&T Labs Research. He received his PhD in Electrical Engineering from the University of Minnesota. His research interests are in the areas of probabilistic and combinatorial modeling and analysis, with emphasis on networks.

Dongmei Wang received the Ph.D. degree from the College of William and Mary, Williamsburg, VA, in 2000. Since then, she joined AT&T Research Lab and has been working on network related research topics, from optical layer to application, architectures to protocols, algorithms to simulation, provisioning to restoration. Most recently, she has been focusing on IPTV related problems. She is the author of more than 30 research papers and has filed 15 patents.