Fast Failure Recovery for Reliable Multicast-based Content Delivery

Ning Wang and Binbin Dong
University of Surrey
Guildford, United Kingdom

Abstract— **In this paper we introduce a new scheme to achieve fast failure recovery in IP multicast based content delivery, which is based on efficient extensions to the Not-via fast reroute (FRR) technique. The design of such an approach takes into account distinct characteristics of IP multicast routing, namely receiver-initiated and state-based, and it offers comprehensive protections against both simple and complex network failures. We also specify in the paper moderate extensions to the standard PIM-SM routing protocol in order to equip individual repairing routers with necessary knowledge for dynamically binding protected multicast trees with pre-established Not-via tunnels that are able to automatically bypass failed network components. Our simulation experiments based on both real and synthetically generated topologies indicate promising scalability performance in the proposed multicast FRR approach.**

I. INTRODUCTION

Emerging multimedia based real-time content distribution services/applications like IP Television (IPTV) and live video streaming have demanded stringent reliability requirements on the underlying network platforms. To avoid the problem of slow re-convergence upon network failures in IP routing, various fast failure recovery and fast reroute (FRR) techniques have been proposed in recent years. The main idea is to immediately divert the affected customer traffic to *pre-provisioned* backup paths once network failures have been detected by the repairing router. Common FRR mechanisms being investigated in IETF include one-hop deflection [1] and IP tunnel based approaches [2, 3]. How to guarantee *full* network protection coverage across all destination prefixes against any simple failure pattern (e.g. single link/node failures) is one of the most concerned issues in the design of FRR schemes. Simple deflections [1] and conventional IP tunnels [2] are not able to automatically guarantee 100% protection coverage. Till now the only FRR paradigm that is able to automatically produce full protection coverage is the Not-via scheme [3], in which repairing routers have the intelligence of computing backup IP tunnels without traversing the protected network component. With some recently proposed enhancements [4, 5], it has been widely believed that Not-via will become a mature FRR technique as a long-term solution in enabling future IP resilience.

It should be noted that current FRR schemes are only designed for point-to-point unicast routing, without taking into account point-to-multipoint multicast requirements. On the other hand, multicast has been regarded as a promising paradigm for supporting real-time content delivery services that demand high reliability guarantees for service assurance to end users. After nearly two decades of research and development, IP multicast [6] has finally seen some large scale deployment by tier-1 ISPs such as Level-3 and Sprint [7]. Under such circumstance, how to enable fast failure recovery in IP multicast will become an imminent research issue. First of all, same as the unicast scenario, an essential requirement is to provide *full* failure recovery for all possible receivers against any simple failure pattern. Unfortunately, the only available technique that is capable of achieving this is Not-via, which has not yet supported multicast routing.

In this paper we introduce a new FRR scheme for protecting IP multicast based services against both simple and complex network failures. This approach can be regarded as a Not-via extension to IP multicast routing, and it automatically inherits the original capability of guaranteeing full failure protection coverage. Nevertheless, due to some fundamental differences between unicast and multicast routing, some additional technical issues need to be specifically considered. A further contribution from this paper is the considerations on not only conventional single link/node failures but also multiple concurrent ones. In particular, we address the issue of how Shared Risk Link Group (SRLG) information can be used for computing comprehensive backup paths in IP multicast routing. Since multicast is mainly used in delivering real-time multimedia contents, providing survivable Quality of Service (QoS) assurance to end users is also essential. Towards this end, in our experiments with both real operational networks and synthetically generated ones, we evaluate how QoS performance such as end-to-end delay in multicast routing will be impacted by the using of Not-via based backup paths. This is in addition to the evaluation on the scalability performance in overhead maintenance required by the proposed approach.

II. BASIC NOTVIA APPROACH

The Not-via approach [3] is an intelligent scheme where routers are able to compute backup tunnels based on dedicated Not-via addresses, which do not traverse the protected network component. As such, 100% protection coverage can be achieved provided that the underlying network topology remains connected. The basic Not-via operation is illustrated in Figure 1. First of all, in addition to the normal IP address for conventional routing, another set of IP addresses known as Not-via addresses is bound to network interfaces of individual routers for traffic diversion upon failures. The semantics of a Not-via address is that a packet addressed to a Not-via address must be delivered to the node advertising that address, not via the protected component with which that address is associated. When a failure occurs, the repairing node encapsulates the affected packets to a Not-via address of the protected interface. From

the repairing node, all the nodes along the backup path are able to know to which next-hop they must deliver the packets in order to avoid traversing the failed interface. In Figure 1 each network interface of a router is assigned with a Not-via address. We assume the normal IGP path from router *r1* to *r6* is *r1-r2-r3-r6*. When *r1* detects that its next-hop node *r2* has failed, it immediately encapsulates the packets destined to *r6* with the Not-via address $r3_2$ using an IP tunnel that terminates at the *next-next-hop* (NNH) *r3* towards the original destination *r6*. This address is interpreted as tunneling the traffic from *r1* to NNH *r3* along the IGP path *not going via r2*. Since all the nodes along this path understand the Not-via address, the traffic will be delivered to *r3* without traversing *r2*. Once a packet arrives at *r3*, it is decapsulated and forwarded natively to the original destination *r6*. Currently this approach is only applicable to the unicast routing scenario.

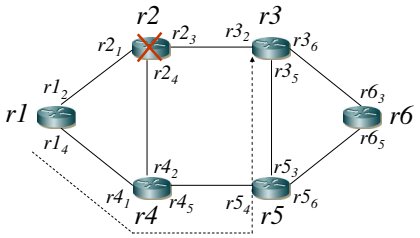

Figure 1. Basic Not-via operations in unicast routing

III. APPROACH OVERVIEW

According to the common practice, there are two fundamental requirements in the design of FRR techniques, namely *pre-computation* of backup paths and *local rerouting*. More specifically, backup paths should be computed *a priori* before any actual failure occurs, and additionally the router that is *adjacent* to the failure should be responsible for the repair. These two properties are inherited into our design of multicast FRR techniques. More specifically, all standard Not-via backup tunnels are provisioned all at once in the bootstrap phase (i.e. before any multicast tree is established). Thereafter, these pre-established tunnels will be dynamically bound to individual multicast trees while they are being constructed.

A. Fundamental challenges

Since all backup tunnels are pre-computed according to the standard Not-via approach [3] in a static way before the creation of any multicast tree, the protection of unicast routing can be assumed already available. To make use of existing backup paths for protecting multicast traffic, the following technical challenges can be conceived.

(1) Anonymous group members – in the IP multicast paradigm where the destination address contained in each data packet is a *logical* multicast group address, intermediate in-tree routers (i.e. potential repairing routers) are unaware of the physical location of downstream receivers remotely attached to the tree. This effectively means that a repairing router does not know the geographical distribution of destinations associated with the group which need to be protected within the network.

(2) Lack of knowledge on "re-grafting" points in case of *node* failures – according to the standard Not-via scheme for unicast FRR, next-next-hop (NNH) is used as the tunnel endpoint for automatically bypassing the failed next-hop node towards the protected destination. As far as multicast routing is concerned, the location of *in-tree NNHs* is not automatically known by the repairing router. As a result, the repairing router is not aware of the Not-via tunnel endpoints for local protections on each group upon a node failure.

(3) point-to-multipoint requirement – existing Not-via tunnels are all point-to-point for unicast routing protections. How these tunnels can be used for point-to-multipoint multicast routing needs to be considered. For instance, is it necessary to extend standard Not-via tunnels to be point-to-multipoint in order to support multicast FRR?

B. Basic operations

In effect, the aforementioned issues can be regarded as the consequence of the original design philosophy of the IP multicast model, namely *receiver-initiated* and *state-based*. To maintain a source specific multicast (SSM [8]) tree for each group, each in-tree router needs to maintain specific multicast group state, often denoted by:

$$\{s, G, iif, oif\ list = [oif_1, oif_2, \ldots oif_n]\}$$

where *s* and *G* are source and group addresses respectively, *iif* identifies the RPF (Reverse Path Forwarding) incoming interface from the source, and *oif list* contains the IP address of individual outgoing interfaces (next-hops) towards active downstream group members. Such information is dynamically updated upon the arrival of PIM-SM (Protocol Independent Multicast – Sparse Mode [9]) join requests sent from Designated Routers (DRs) attached with newly joined hosts. In case a specific *oif* becomes unavailable due to the failure of next-hop node, the key task concerned by the multicast FRR is how all the affected downstream group members can be recovered. As previously mentioned, the upstream in-tree router that is adjacent to the failure does not have any knowledge about the distribution of affected remote downstream receivers, and nor does it know the protection tunnel endpoints from where all the affected receivers can be re-grafted onto the tree.

Let's take Figure 2(a) as an example where part of a multicast delivery tree is shown in solid lines (virtual lines denote physical network links), which connects three DRs *r8*, *r9* and *r10* attached with active group members. According to PIM-SM based IP multicast routing, the intermediate node *r1* only knows that two of its own neighbors (nodes *r3* and *r4*) are currently in the multicast tree, but nothing beyond those points. Hence in case any of these two next-hop nodes leading towards further downstream members is broken, the repairing node *r1* cannot automatically figure out which NNHs need to act as tunnel endpoints for re-grafting all the affected remote group members. For instance, it is not necessary to use a Not-via tunnel terminating at the two-hop-away node *r6* for this specific group under consideration, as *r6* is neither a member itself nor an in-tree router connecting any downstream member. Hence a fundamental issue to be concerned is to equip potential repairing routers in the multicast tree with the knowledge about the location of remote *in-tree* tunnel endpoints in order to protect all affected downstream receivers. In Figure 2(a), node *r1* needs to know all the downstream NNHs in the multicast tree in case its outgoing interface cannot detect the reachability of nodes *r3* or *r4*. Based on this knowledge,

pre-provisioned Not-via tunnels (e.g. tunnel *r1*→*r2*→*r7* whose endpoint is identified by Not-via address $r7_3$, and *r1*→*r5*→*r8* whose endpoint is identified by Not-via address $r8_4$, both in dash-dot lines) can be used as backup paths for re-connecting remote members *r9*, *r10* (against the failure of *r3*) and *r8* (against the failure of *r4*) respectively.

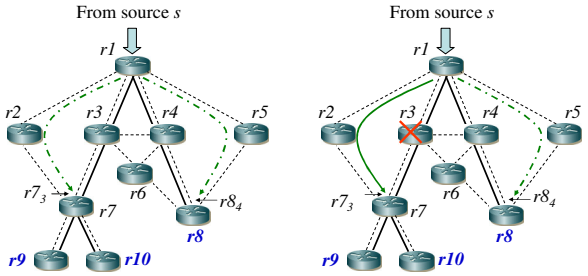

Figure 2. An example

In section III.C we will specify in detail how moderate extensions to the existing PIM-SM protocol can be realized in order to disseminate necessary NNH information while a multicast tree is being constructed. Here we first illustrate how the standard Not-via approach can be used for multicast FRR purposes.

As previously mentioned, all Not-via tunnels need to be pre-provisioned at the bootstrap phase before any multicast group join is triggered. Accompanied with the PIM-SM join requests being delivered across the network for building multicast tree for each group, the group states are also dynamically bound to those pre-provisioned Not-via tunnels at each hop travelled by the join request. More specifically, recall that conventional Not-via uses dedicated tunnels which terminate at an NNH but without traversing the protected next-hop node. Similarly, to protect a multicast delivery tree, each intermediate in-tree node should make use of its local Not-via tunnels for protecting remote downstream branches in case its next-hop node towards them becomes unavailable. Hence, upon receiving a PIM-SM join request, each router *r* needs to bind the corresponding network group state to the Not-via tunnel terminating at the NNH from the direction the join request was sent. In this case the next-hop (downstream) node, which is effectively the neighboring router that forwarded the join request to router *r*, can be protected by the tunnel. More specifically, the Not-via address corresponding to the tunnel that terminates at the NNH is added to the *protection tunnel list* for the outgoing interface towards the protected next-hop neighbor.

Let's use Figure 2(b) as an example. Once router *r1* has received a PIM-SM join request from its downstream neighbor *r3* for a specific group *G*, it binds the group *G* state to the pre-provisioned Not-via tunnel *r1*→r2→*r7* (whose endpoint is identified by the Not-via address $r7_3$) for protecting the failure of *r3*. Not-via address $r7_3$ represents a new backup tunnel associated with *r1*'s outgoing interface towards *r3* in multicast group *G*. Upon the failure at *r3*, the local repair router *r1* is able to immediately divert the multicast traffic of group *G* onto this tunnel to reach its in-tree NNH which is *r7*, but without traversing *r3*, such that both affected downstream members *r9* and *r10* can be recovered. More specifically, on detecting the failure of *r3*, *r1* immediately encapsulates the affected multicast packets (whose destination address is *G*) using the bound Not-via address $r7_3$, in which case the traffic can be rerouted via the activated backup tunnel *r1*→*r2*→*r7* shown in the solid line in Figure 2(b). Once the tunnel endpoint *r7* has received the encapsulated packets, it decapsulates them into conventional multicast packets containing the original IP multicast group address *G*. From this tunnel endpoint, the affected multicast traffic can be natively delivered across the original branch(s) to reach their downstream group members. Naturally the encapsulated multicast packets should not undergo the conventional RPF check at the tunnel endpoint, as anyway they are not supposed to come from the interface leading towards the source *s*.

From the description above, we can see that the principle of using Not-via for multicast tree protection is still on per-hop basis instead of being end-to-end. It is also worth mentioning that multiple multicast trees can share the same Not-via tunnel if they have some overlapping branches. For instance, in Figure 2(b) in case *r1* receives another PIM-SM join from *r3* regarding group *G*', it performs another independent binding of the group state *G*' to the same Not-via tunnel *r1*→*r2*→*r7* (endpoint identified by the Not-via address $r7_3$) even if the downstream group member distribution is different from that of group *G* (e.g., only *r9* is attached with receivers for *G*'). In this case, it can be easily inferred that this approach scales very well as the total number of Not-via tunnels needed is independent of the number of multicast trees within the network.

C. *Extensions of PIM-SM protocol*

As we mentioned, extensions to the PIM-SM routing protocol is necessary for supporting such multicast FRR operations. Specifically, there are two distinct features associated with the extension to the protocol: (1) additional information to be carried by PIM-SM join requests, and (2) changed packet forwarding rule on these join requests.

- *Adding a new NNH address entry in PIM-SM join request packets*

First of all, in order to provide each potential repairing router the necessary information about in-tree NNH for protecting its immediate next-hop, the IP address of the NNH should be carried in a new entry contained in PIM-SM join requests in order to allow the potential repairing router to bind the multicast group address with the pre-provisioned Not-via tunnels. For instance, in Figure 2(b), upon receiving a PIM-SM join request for group *G* from its downstream neighbor *r9*, router *r7* needs to insert the IP address of *r9* into the newly added *NNH address entry* in the join request before forwarding it to *r3* in the direction towards the source *s*. Upon receiving the join request, *r3* immediately binds group *G* with the Not-via tunnel identified by $r9_7$ (not shown in the Figure) based on the corresponding NNH address *r9*. That is, *r3* adds $r9_7$ to the Not-via protection tunnel list associated with the *oif* pointing towards *r7*, and this can be done by a local mapping from the received NNH address *r9* to the Not-via address $r9_7$. Similarly, *r3* needs to insert the IP address of *r7* (to replace that of *r9*) into the NNH address entry in the PIM-SM join request before forwarding it to *r1*. As a result, *r1* is able to bind group *G* with its local tunnel identified by $r7_3$ based on the received NNH address *r7*.

- *Changing the forwarding rule on PIM-SM join requests*

According to the standard PIM-SM routing, each join request terminates at the first node along the join path that has already been included in the tree, without necessarily being delivered all the way towards the tree root (e.g. the source in SSM). This forwarding rule is not sufficient to achieve full protection chains on the entire tree with multiple branches. Let's take Figure 2(a) as an example again. Assume *r9* has already joined group *G* – when a new member *r10* sends a join request to *r7* in the direction towards the source *s*, *r7* will not further forward this request to *r3* as it has already been in the multicast tree. As a result, *r3* is not aware of its new NNH address *r10* and hence cannot bind group *G* with the Not-via tunnel terminating at *r10* (not shown in the Figure) in order to protect it against the failure of *r7*. Based on this observation, we propose that each PIM-SM join request should be forwarded further *by one single hop* when it hits the first in-tree router. It is easy to figure out that to forward join requests by one single hop is sufficient, as further up merged tree branches can be commonly protected by a single tunnel.

According to the above description, we show in Figure 3 a general scenario of multicast tree protection using Not-via tunnels (only in-tree links are shown), based on the proposed extensions to PIM-SM. At each in-tree router, the enhanced group state with Not-via tunnel protection information can be viewed as:

$$\{s, G, iif, [oif_1 (no.\ of\ tunnels, tunnel1, tunne2\ldots)], [oif_2,(\ldots)] \ldots [oif_n (\ldots)]\}$$

From this entry structure, we can notice that each outgoing interface *oif* is effectively protected by a set of Not-via tunnels terminating at different in-tree NNHs, depending on the distribution of its own downstream group members. Each tunnel list (represented by the associated Not-via addresses) on per *oif* basis is dynamically updated upon the receipt of new PIM-SM join requests.

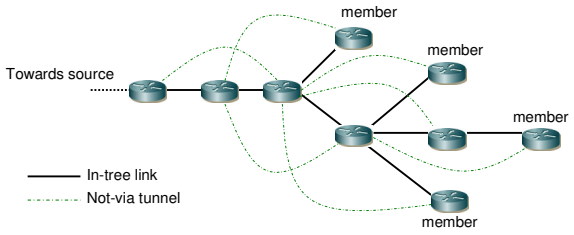

Figure 3. A general Not-via based protection scenario

IV. DEALING WITH COMPLEX FAILURE PATTERNS

In this section we discuss how to protect multicast trees against complex network failures. In particular, we focus on the scenario of multiple concurrent *link* failures with shared risk link group (SRLG) information which has been investigated widely in the literature. According to [3], the computation of backup Not-via tunnels in unicast routing is to bypass all links belonging to the same SRLG. In particular, the backup path should not contain any link that belongs to the same SRLG as the protected one. One special scenario is that multiple links of the same SRLG may constitute the *primary* path from a repairing router to the destination.

As shown in Figure 4, if links *r1*-*r2* and *r3*-*r4* along the primary path from *r1* to *r5* belong to the same SRLG *A*, there are two distinct strategies for failure protection: either to establish a *global* backup path from *r1* to *r4* to bypass both protected links (as shown in the dash-dot line), or to perform separated/decoupled protections on individual links (as shown in the dash lines).

The advantage of the first option is straightforward – such an approach is more efficient in terms of the number of tunnels required as well as the overhead associated with packet encapsulations and decapsulations. Nevertheless, as far as multicast routing is concerned, additional adaptations on both the packet format and the forwarding rule on PIM-SM join requests are required. First of all, the original *NNH address* carried by join requests needs to be replaced by the SRLG group ID, which allows the potential repair router (like *r1* in Figure 4) to bind the multicast group state to the pre-computed (global) backup tunnel identified by the Not-via address associated with that SRLG. In addition, each join requests should be delivered all the way to the source (root of the tree), regardless whether a router that receives the request has already been in the tree or not. In Figure 4, assume that *r7* has already joined a multicast group *G* whose source is to the left side of *r1* (not shown), hence routers *r1*, *r2*, *r3*, *r6* and *r7* are currently included in the tree. In case a new join request is sent from *r5*, this request should be forwarded all the way to the source in order to allow the repairing router that is closest to the source (in the worst case the DR of the source itself) to bind the group state to the *global* backup tunnel that automatically bypasses all the protected links belonging to SRLG A along the join path.

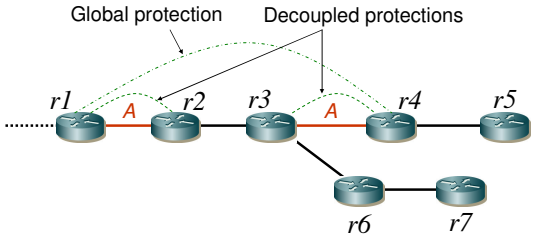

Figure 4. Not-via protections in the SRLG scenario

An alternative approach is to treat these failures as separated ones and deal with them in a decoupled manner. For instance, when both links *r1*-*r2* and *r3*-*r4* fail along a specific multicast tree branch, repairing router *r1* only needs to use the local Not-via tunnel (in dash-line) to divert the affected multicast traffic to its next-hop *r2*, but not via the failed link that directly connects the two nodes, and from *r2* the multicast packets are natively delivered along the normal branch until they reach *r3*. Once again, *r3* needs to encapsulate the packets and send them via the Not-via tunnel to reach *r4*. Although this approach needs multiple encapsulation/decapsulation operations, the necessary extensions on the PIM-SM routing protocol can be much simplified. Indeed, according to [3], it is also suggested that decoupled treatment of local failures is more desired, mainly due to the potential combinatorial explosion issue of Not-via addresses in global protections.

V. PERFORMANCE EVALUATION

To evaluate the performance of our proposed scheme, we use both a real operational network topology of GÉANT [10] and a synthetically generated topology in our simulation experiments. In 2004, the GÉANT topology contained 23 nodes and 74 links, and the corresponding IGP link weight configuration was also published which has been used in our experiments. To evaluate the performance at a larger scale, the synthetic network topology generated by the BRITE topology generator [11] contains 100 nodes. In both cases we evaluate the relevant performances by using 10 randomly generated multicast groups.

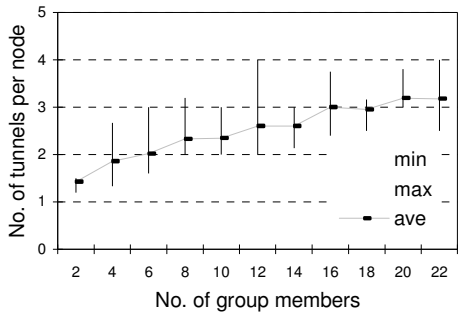

Figure 5. Scalability in overhead maintenance (GÉANT)

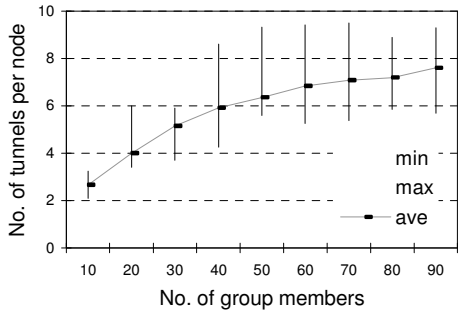

Figure 6. Scalability in overhead maintenance (Synthetic topology)

First of all we study the scalability performance of the proposed multicast FRR scheme. Recall that one or multiple Not-via addresses, each representing the endpoint of a dedicated backup tunnel, need to be bound with the multicast group state on per outgoing interface basis. Hence it would be interesting to study the corresponding memory overhead of maintaining such extra information at individual repairing routers. We look at the total number of Not-via tunnels that are needed to be associated to a specific multicast group at each in-tree router. Figure 5 shows the corresponding results (each data point associated with three values: min/max/average) based on the GÉANT network topology, with the average number of members per group varying from 2 to 22. As can be inferred, with the growth of each multicast tree, the increasing (1) out-degree of each in-tree router and (2) the number of in-tree NNHs certainly indicate that more backup tunnels are needed for protection. Nevertheless, the good news is that such growth in the maintenance overhead is not sharp with the increase of the group member population. Even with densely populated multicast groups with 22 receivers (note the network size is 23), the average number of tunnels to be maintained is only 3.3, with the worst case being 4 tunnels per node. Such a moderate increment incurred against the population of multicast group members indicates promising scalability performance in terms overhead maintenance. Similar observations have been obtained based on the random topology with larger network size (100 nodes).

Since IP multicast is usually used for delivering real-time multimedia based contents, QoS assurance requirements should also be considered when protecting multicast trees. One specific issue we address is the extra delay introduced by using alternate backup tunnels as compared to the primary tree paths. We define the metric of end-to-end *extra delay proportion* (*EDP*) for each multicast group *G*:

$$EDP^G = \frac{1}{|G|}\sum_{k=1}^{|G|}\frac{D^t(s^G, r_k^G)}{D(s^G, r_k^G)}$$

where $D(s^G, r_k^G)$ and $D^t(s^G, r_k^G)$ represent respectively the end-to-end delay from the source of the group s^G to k^{th} receiver r_k^G along the primary path and the end-to-end delay from s^G to r_k^G when using the Not-via tunnel upon the failure of node *t*. Figure 7 shows the average and the worst case of *EDP* against each node failure scenario in the GÉANT network topology across 10 groups. According to [12], the actual link weight setting in the GÉANT network is based on the (propagation) link delay, and hence *EDP* values can be calculated according to the actual link weight setting. From the figure we can see that in most of the node failure scenarios, the average *EDP* is below 2.0, meaning that the end-to-end delay by using backup Not-via tunnels upon each node failure is usually no more than twice as that of the normal paths under failure-free conditions. Nevertheless, in some cases such as the failure of node 10 the worst case *EDP* value can be as high as 5.9. This is because many affected group members need to be reconnected on to the multicast tree via much longer backup tunnels upon the failure of the node, particularly due to the wide range of delay across all GÉANT network links.

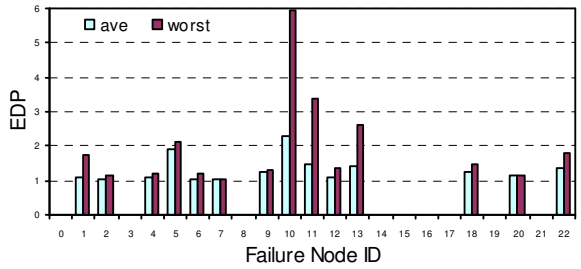

Figure 7 *EDP* performance (GÉANT)

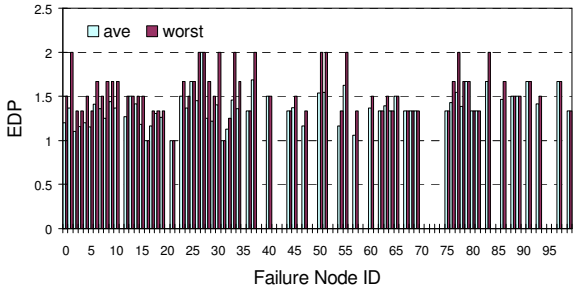

Figure 8 *EDP* performance (Synthetic topology)

On the other hand, the reason why the failures of some nodes do not have *EDP* values in the Figure is that these nodes do not provide "transit" services to any other node pairs according to the GÉANT link weight setting, and hence their failures do not impact the communication between any other nodes. As for the synthetically generated network topology where: (1) the link weights are simply set according to hop-counts and (2) the delay of each link is assumed to be the same, the corresponding *EDP* values are lower as indicated in Figure 8. In particular, the worst case *EDP* values can be bounded up to 2.0. This is phenomenon is expected as for most of the cases, the length of Not-via tunnels is not significantly larger than that of the primary paths in such a topology where neither the link weight setting nor the delay significantly varies across all links. From the above observations, we can infer that the setting of IGP link weights may significantly impact the relevant *EDP* performance in multicast FRR.

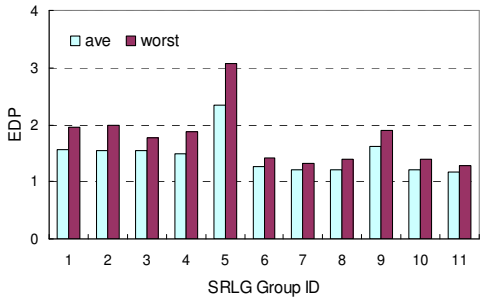

Figure 9. SRLG-based *EDP* performance (GÉANT)

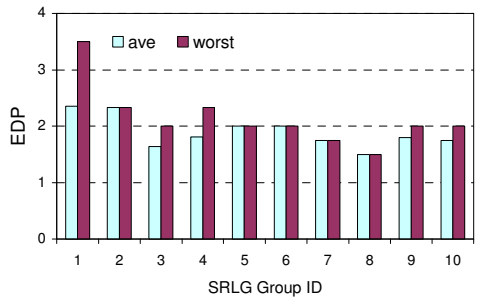

Figure 10. SRLG-based *EDP* performance (Synthetic topology)

We now study the *EDP* performance under SRLG scenarios in terms of *dual link failures*. The approach we evaluate is the separate/decoupled protection as described in section IV, since it is simple to implement especially in terms of PIM-SM extensions. In Figure 9 we show the average and the worst-case performance in dual-link failure scenarios in the GÉANT network. More specifically, ten independent SRLG groups are examined, each associated with two simultaneously failed network links (either adjacent or non-adjacent to each other). From the figure we can see that for most of the SRLG groups the worst-case *EDP* value is below 2.0, meaning that for most of the affected group members the backup path has up to doubled delay as compared to that of the primary tree path. This situation is similar to the single *node* failure scenario – note that the failure of one node means all its directly attached links will become unavailable. Figure 10 shows the corresponding performance with the synthetically generated topology. It should be noted that such extra delay (in both simple and complex failure scenarios) is not the result from our proposed scheme, but due to the automatic calculation of alternate routes by the standard Not-via scheme. On the other hand, we can clearly see the relevant performance impact on backup paths from the IGP link weight configurations. As such, how to tune the underlying IGP link weight in order to improve the performance (e.g. *EDP*) would be an interesting issue to be investigated in the future.

VI. Conclusion

Compared to various FRR mechanisms applied in unicast routing, how to protect IP multicast trees against failures has not been widely investigated till now. In this paper we proposed an efficient Not-via based FRR scheme for multicast routing. Through dynamic associating pre-established backup tunnels with the multicast trees to be protected, fast failure recovery can be achieved against both simple and complex network failures. To realize relevant multicast FRR functions, moderate extensions to the underlying PIM-SM protocol is necessary, and has been discussed in this paper. Our simulation results show promising scalability performance in terms of overhead maintenance for backup tunnels. In addition, we have also evaluated QoS performance in terms of extra end-to-end delay incurred by using these backup tunnels.

VII. Acknowledgement

This work was partially funded by the EU FP7 4WARD Project (216041) and COMET Project (248784).

References

[1] A. Atlas, A. Zini, "Basic Specification for IP Fast Re-route: Loop-free-Alternates," IETF RFC 5286, September 2008
[2] S. Bryant, M. Shand, "IP Fast Reroute Using Tunnels," draft-bryant-ipfrr-tunnels-03, November 2007
[3] M. Shand et al, "IP Fast Reroute Using Not-Via Addresses," draft-ietf-rtgwg-ipfrr-notvia-addresses-05, March 2010, work in progress
[4] A. Li et al, "On improving Improving the Efficiency and Manageability of NotVia", Proc. ACM CoNext 2007
[5] G. Enyedi et al, "IP Fast Reroute: Lightweight NovVia", Proc. IFIP Networking 2009
[6] S. Deering, "Host Extensions for IP Multicasting", IETF RFC 1112, August 1989
[7] Multicast ISP List: http://www.multicast-isp-list.com/
[8] H. Holbrook et al, "Source Specific Multicast for IP", IETF RFC 4607, August 2006
[9] D. Estrin et al, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF RFC 2362, June 1998
[10] The GEANT network topology: http://www.geant.net
[11] The BRITE topology generator: http://www.cs.bu.edu/brite/
[12] S. Uhlig et al, "Providing Public Information Traffic Matrices to the Research Community", ACM CCR Vol. 36, No. 1, 2006, pp. 83-86