Performance Comparison of Resilience Mechanisms for Stateless Multicast Using BIER

Wolfgang Braun[1], Manuel Albert[1], Toerless Eckert[2], Michael Menth[1]
University of Tübingen, Department of Computer Science, Germany[1], Cisco Systems, Inc.[2]

***Abstract*—Bit Indexed Explicit Replication (BIER) is a novel multicast forwarding scheme for IP networks that avoids states in replicating routers by encoding the multicast information into a bit string in the packet header. In addition, the BIER-TE variant encodes the multicast tree in the header and allows for network programmability.**

We propose the use of maximally redundant trees (MRTs) for 1+1 protection in BIER that currently lacks this feature. We further discuss three different fast reroute (FRR) protection schemes for BIER-TE we have proposed in the Internet Engineering Task Force (IETF). They use header modification only (HM), rely on point-to-point tunnels (PPT), or leverage BIER-in-BIER encapsulation (BBE). We compare them regarding protection coverage, path lengths, traffic loads, required network capacity, state requirements and overhead in a large number of networks. The results serve the discussions in IETF where BIER and BIER-TE are currently standardized.

***Index Terms*—Bit Indexed Explicit Replication, Multicast, Resilience, Scalability, Resource Management**

I. Introduction

IP multicast allows efficient data transmission from one sender to many receivers. In general, it is desirable that traffic is optimally forwarded in such a way that the least packet duplication and load in the network occurs. The research of the past decades improved many areas for multicast, e.g., traffic-engineered multicast deployment. However, network operators today still face several operational problems when applying current multicast protocols, e.g., Protocol Independent Multicast (PIM), to common use cases such as L3VPN, IPTV, and over-the-top services [1], [2]. The protocols generally rely on explicit tree building mechanisms. The trees have to be installed in the routers and increase their overall state. For some use cases, too much state is required in the routers so that they are unable to optimally forward the multicast traffic. E.g., operators often flood packets of some multicast flows to all potential egress nodes regardless of a subscription. This wastes bandwidth to save state in routers and operators have to find the right tradeoff. Multicast state in routers also causes operational issues if the network is reconfigured or subscribers are added or removed. The control plane requires many routers to participate in the process, compute new trees, and install new rules which may result in convergence times of multiple minutes in large multicast deployments.

The authors acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG) under grant ME2727/1-1. The authors alone are responsible for the content of the paper.

The IETF currently works on Bit Indexed Explicit Replication (BIER) to address the issues mentioned above. With BIER, multicast traffic can be forwarded without per-multicast-flow state by encoding the egress nodes of a multicast flow in a new BIER header [3]. Ingress nodes add this BIER header to multicast packet, transit nodes only forward them without multicast-flow-specific information, and egress nodes remove the BIER header. Thus, BIER supports a simple multicast overlay which is easy to operate and requires minimal state in routers. In addition, Traffic Engineering for BIER (BIER-TE) [4] is proposed in the same working group. It encodes the multicast tree in the BIER header and allows for traffic-engineered multicast trees with minimal state overhead.

In this work, we propose a 1+1 protection mechanism for BIER because it currently lacks such a mechanism. It is based on existing IETF techniques and is compatible with the current BIER specification. We further suggest three different fast reroute (FRR) mechanisms for BIER-TE that we recently also proposed for standardization in IETF [5]. We explain the operation of these mechanisms and constraints for the computation of their path layout. We analyze and compare the proposed resilience mechanisms for BIER and BIER-TE on 220 network topologies with regard to protection coverage, path lengths, network loads, capacity requirements, state requirements, and header overhead.

The remainder of this paper is structured as follows. Section II discusses related work. In Section III we explain BIER and propose a novel 1+1 protection mechanism for BIER. Section IV introduces BIER-TE and propose three different FRR mechanisms for BIER-TE. Section V discusses performance results. Finally, Section VI concludes the paper.

II. Related Work

There are various approaches to construct node-redundant pairs of multicast trees [6]–[10] that can be used to implement a 1+1 protection scheme for multicast traffic. These approaches differ by the considered objective function such as cost, bandwidth, computation complexity, required network state, and update complexity. We leverage the routing topologies of Maximally Redundant Trees (MRTs) [10], [11] as node-redundant pairs of multicast trees for a novel BIER protection mechanism because they impose state and computation requirements for routing underlays that scale well for large networks. The Parallel Redundancy Protocol (PRP) [12] and High Availability Seamless Redundancy (HSR) [13]

provide node-redundant multicast trees that suffice hard real-time constraints in industrial Ethernet networks.

Reliability for multicast traffic can be implemented using acknowledgments (ACKs) and selective repeat transmissions similar to TCP. However, the number of ACKs of many receivers likely overburden a source. Therefore, the Reliable Multicast Transport Protocol (RMTP) [14] and other approaches [15], [16] use a shared ACK tree structure to overcome this problem.

MPLS is currently deployed in many ISP networks and provides multicast services with point-to-multipoint (P2MP) LSPs [17]. Such P2MP services support FRR by local repair when RSVP-TE is used. However, these solutions can be unsuitable when multicast group memberships change frequently [18].

There are various algorithms and mechanisms to compute traffic-engineered multicast trees based on multiple objective functions. An exhaustive survey of existing methods is given in [19]. The approaches are classified by objective functions, constraints, etc. The authors also propose a generalized multi-objective optimization that is based on load-balanced multiple trees. Most works focus on the computing algorithm and do not discuss the required router state in detail. Yet, these works are orthogonal to the BIER-TE approach because BIER-TE does not propose any path layout but rather supports encoding of optimized multicast trees in the BIER header.

There are several software-defined networking (SDN) approaches for IP multicast. In [20], the authors implement IP multicast using VXLAN in datacenters and remove the need for periodic control plane interaction. A highly scalable IP multicast datacenter method is proposed in [21] which leverages flow aggregation to support large numbers of multicast joins simultaneously. "Dynamic Software-Defined Multicast" [22] reduces control plane complexity in multicast deployments of ISP networks and adds traffic engineering aspects using multiple trees similar to [19]. BIER and most SDN approaches simplify the control plane while supporting frequent multicast changes. BIER introduces header overhead but only requires a minimal amount of state in transit routers. In contrast, most SDN methods leverage existing header fields but require significantly more state in forwarding devices.

III. Bit Indexed Explicit Replication (BIER)

We first introduce the BIER architecture as defined in [3] and then propose a 1+1 protection scheme for BIER that is compatible with the current specification.

A. Architecture and Forwarding Procedure

BIER provides a multicast overlay without any states for multicast tree on core routers. A BIER domain consists of so-called bit forwarding routers (BFRs). Ingress BFRs add a BIER header to incoming multicast packets. The BIER header indicates all BFRs that should receive a copy of the packet, so-called egress BFRs. To that end, each BFR in the network is represented by one bit position in the BIER header which is set if the BFR is an egress BFR for the packet. BFRs forward BIER packets solely based on their BIER header and a Bit Indexed Forwarding Table (BIFT) whose entries depend on the routing underlay, which may be an Interior Gateway Protocol (IGP) such as OSPF or ISIS.

The forwarding procedure works as follows. A BFR essentially forwards BIER packets towards all egress BFRs indicated in the BIER header over the routing underlay. However, it sends at most one copy towards each next-hop (NH) and clears all egress BFRs in the BIER header that are not reachable by itself over the respective NH. The BIFT contains information that supports a BFR to perform these operations efficiently for fast packet processing. When a packet reaches an egress BFR, the BIER header is decapsulated and the packet is forwarded as usual. A salient feature of BIER is that only ingress BFRs need to know multicast groups including egress BFRs in order to add appropriate BIER headers to packets. All other BFRs in the BIER domain do not need to know these groups for multicast forwarding. This makes BIER scalable and requires only reconfiguration of ingress BFRs if multicast group membership of egress BFRs changes. Essentialy, BIER removes the multicast state of conventional multicast routing protocols from core routers by encoding the multicast egress nodes into the packet headers.

BFRs may support bit strings (BIER headers) between 64 and 4096 bits, the support for a bit string length 256 is mandatory. Amongst other protocol information, it mainly contains the bits for potential egress BFRs. If the bitstring is too small to accommodate all BFRs, the set of potential egress BFRs can be broken down into several sets [3]. They basically receive traffic over egress-BFR-disjoint but possibly overlapping multicast trees which increases the traffic load in the network. Thereby, very large topologies can be supported. BIER supports different subdomains for which different routing underlays may be used. Thereby, they can be leveraged to forward multicast packets differently, e.g., for traffic engineering purposes. If multiple subdomains are in use, the ingress BFR chooses the appropriate one for an incoming packet and indicates it in the BIER header.

B. Multicast only Fast Reroute (MoFRR for BIER)

BIER does not provide a protection mechanism but rather relies on the restoration process of the routing underlay which may be slow and cause packet loss. We propose to combine Multicast only Fast Reroute (MoFRR) [23] with BIER to provide fast protection. MoFRR is based on 1+1 protection: the traffic is duplicated at the source and sent over redundant paths to the destination. The destination node continuously measures the quality of the streams from both paths, selects the one with highest quality for forwarding, and discards the other. In case of multicast, two redundant trees are required. The ingress node duplicates the traffic, sends it over both trees, and egress nodes forward packets from only one of them. If one tree fails, the egress node is likely to still receive the traffic from the other tree.

The BIER architecture may be upgraded as follows to implement MoFRR. At least two subdomains with different routing underlays are needed to allow for redundant multicast

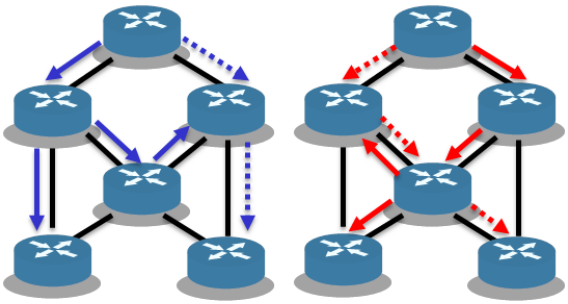

Fig. 1: The blue and red MRT routing topologies forward traffic over node-redundant paths.

trees. The ingress BFR copies incoming BIER packets to two subdomains that provide redundant multicast trees. The egress BFRs read from the two streams and forward only a single copy.

A challenge is the provision of two routing underlays such that they yield redundant trees. We propose to leverage Maximally Redundant Trees (MRTs) for that purpose which have been standardized by the IETF [24]. This idea has already been proposed for other multicast mechanisms in [25], but the draft has been abandoned. MRTs calculate dual routing topologies in a distributed way. In the absence of failures, a packet can be delivered of both of them to all destinations. The resulting paths in the two routing topologies do not necessarily form a pair of trees but the paths are node-redundant in the sense that the packet reaches any destination over at least one routing topology in case of any single link failure. Figure 1 illustrates how a packet is forwarded over two redundant topologies (blue and red) from one source to all other nodes depending on its destination address. The straight lines represent a tree along which a packet may be delivered to all intermediate nodes and leaves. The dashed lines show how the packet reaches a single remaining node over an additional path whose intermediate nodes receive packets over the tree. This example helps to understand some evaluation results in Section V. However, traffic is mostly carried over real trees and it was in fact difficult to find this small counterexample.

To protect unicast forwarding, MRTs are applied as follows. In the failure-free case, traffic is forwarded over shortest paths instead of one of the two routing topologies. In case of a failure, it is locally rerouted over a working routing topology. This may lead to very long backup paths [26].

We propose to use the MRT routing topologies for multicast forwarding although they do not form a tree. Nevertheless, we denote them as multicast trees when used in that context. For this purpose, multicast traffic is distributed along the two routing topologies to all required destinations. In case of a failure on one topology, the traffic is dropped instead of being switched to the other. In the context of BIER, the two MRT routing topologies must be maintained by routing underlays in the two different subdomains and ingress BFRs copy incoming multicast packets to both of them. Packets are delivered only once to egress BFRs over a routing topology because BFRs modify the set of egress BFRs on forwarding in the packet header. This mechanism prevents that egress BFRs accidentally obtain separate copies over a solid and the dashed path within a routing topology and ensures that BIER packets are forwarded only when needed.

Standardized algorithms for calculation of the routing topologies are available in [11]. They can be computed using a few spanning tree operations in a very fast manner which is often desired in ISP networks. However, the path layout is not always optimal. Therefore, this MRT-based MoFRR solution is appealing for the deployment of BIER in large networks. The improvement of a node-redundant path layout for routing underlays with little state is an open research question.

IV. BIER Traffic Engineering (BIER-TE)

In this section, we present the BIER-TE architecture and its forwarding operations. We suggest a general operation of FRR for BIER-TE an then propose three different implementation options for BIER-TE FRR.

A. The BIER-TE Architecture

The BIER-TE architecture [4] is based on a segment routing [27] approach that is similar to SDN in the sense that the path layout for each flow can be explicitly configured by a controller. BIER-TE leverages the BIER header defined in the BIER architecture [3]. There are two main differences between BIER and BIER-TE. First, BIER-TE encodes both the links and the egress nodes of a multicast tree in the BIER header while BIER only encodes the latter. Second, unlike BIER, BIER-TE does not necessarily require an IGP control network or a routing underlay.

The BIER-TE architecture consists of a BIER controller and BFRs. The controller computes traffic-engineered multicast trees and instructs ingress BFRs to apply appropriate BIER headers to incoming multicast traffic. These BIER headers contains forwarding information and reflects the multicast structure. Furthermore, the controller installs forwarding rules in the forwarding tables of the BFRs which are called Bit Indexed Forwarding Table (BIFT). Their contents is independent of existing multicast flows. A link between two BFRs in the BIER overlay is called an adjacency. An adjacency may be a physical link directly connected to a BFR neighbor or a tunnel provided by the routing underlay (remote adjacency). It is possible to have more than one adjacency between two BFRs when the destination BFR is reachable through different interfaces.

B. BIER-TE Forwarding

We illustrate the main forwarding procedure in Figure 2. Packets are sent from A to C and D. The initial header contains the links A→C, A→B, B→D and the egress nodes C and D. The egress node bits are called `local_decap` bits. The adjacencies and the `local_decap` bit of a BFR are called Bits of Interest (BOI) and are highlighted in blue in the headers. There are two BOI for A, A→C and A→B. A clears its BOI from the header and sends the packets to C and B. The packet arriving at C only has C set. The BIER header

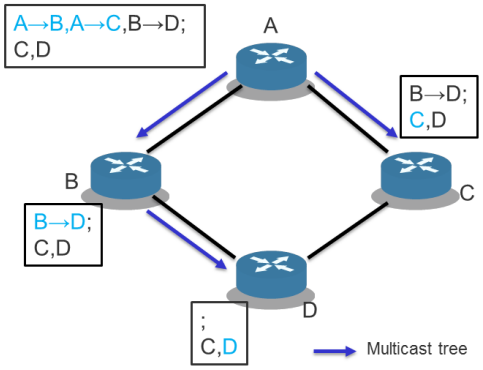

Fig. 2: The BIER header contains adjacencies (first line) and `local_decap` bits (second line). The BOI of the nodes are highlighted in blue. The BIER header is modified before the packet is forwarded to the NH.

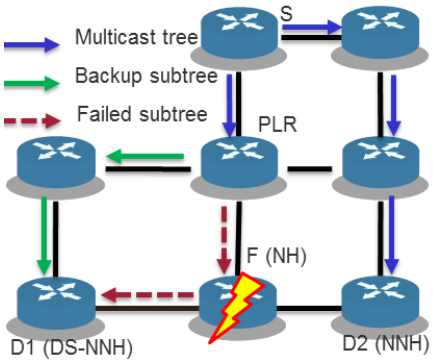

Fig. 3: With node protection, the PLR reroutes traffic to all downstream next-next-hops (DS-NNH).

is decapsulated and the packet leaves the BIER overlay. Note that the BIER header is not empty at C and still contains bits with regard to the other subtree of the multicast tree. This is important for one of the FRR schemes presented in Section IV-C. The packets at subtree at B are processed in the same way as at A.

C. Fast Reroute for BIER-TE

The reachability of neighboring BFRs over a specific adjacency can be controlled by a Bidirectional Forwarding Detection (BFD) component so that a BFR can detect that a neighbor is no longer reachable and locally reroute affected traffic, i.e., the BFR acts as point of local repair (PLR).

If a BFR detects that a BIER packet needs to be forwarded over a failed adjacency, the BFR uses information in the BIER header to consult the BIER-TE Adjancency Fast reroute Table (BTAFT). This yields a backup path including its encoding that the BFR applies to the packet header. Then, the BFR forwards the packet over another adjacency according to the modified header. The state information in the BTAFT of a BFR depends only on the number of its neighbors but not on traversing multicast flows.

To protect a link failure, the PLR forwards an affected packet to its NH over a backup path that bypasses the failed link. To protect a node failure, the PLR forwards an affected packets to all downstream next-next-hops (DS-NNH) over possibly several backup paths that bypass the failed node. A DS-NNH is a next-next-hop (NNH) of a PLR that receives the packet over the failed subtree. This concept is illustrated in Figure 3. A BFR can efficiently determine DS-NNHs using the BIER header and the information in the BTAFT. When link and node protection is combined, the PLR forwards an affected packet to all DS-NNHs and to the NH only if the NH is a destination of the packet. In general, the BFR cannot differentiate between link and node failures. Therefore, the controller configures the BTAFT of all BFRs such that link protection, node protection, or both are supported.

D. Three Implementation Options for BIER-TE FRR

We propose three different implementation options for BIER-TE FRR. For more technical details, in particular for the HM method, we refer to our specification in [5].

1) Point-to-Point Tunneling (PPT): With PPT, the PLR reroutes BIER packets by tunneling them to appropriate NHs and DS-NNHs over unicast tunnels. They are provided by a routing underlay and bypass the failed links and nodes, respectively. E.g., MPLS [28] may be used as a routing underlay. The provision of the tunnels possibly complicates the operation of the routing underlay and increases its state information. Moreover, each tunnel represents an additional adjacency and requires a separate bit in the BIER header. If a PLR reroutes a BIER packet over several unicast tunnels, some of them may share common links, which unnecessarily increases the traffic load on these links compared to the use of point-to-multipoint tunnels.

2) BIER-in-BIER Encapsulation (BBE): With BBE, the PLR identifies the set of NH and DS-NNHs to which a BIER packet needs to be forwarded. The BTAFT helps to create a BIER-TE header towards these nodes avoiding the failed link or node, respectively. The BIER packet is encapsulated with that additional BIER-TE header and sent over the point-to-multipoint structure, which avoids unnecessary traffic increase on some links. The BIER packet is decapsulated at the egress nodes of this multipath.

3) Header Modification (HM): With HM, the backup path is encoded in the existing BIER header through application of an AddBitmask and a ResetBitmask. Due to the forwarding mechanism of BIER-TE, this may cause duplicate packets for some multicast leaves. Therefore, some bits have to be cleared to avoid such duplicates by applying a ResetBitmask. We explain the occurrence of duplicates by the example shown in Figure 4. There are two multicast trees: (1) A sends to C and D, (2) B sends to C and D. If the link B→C fails, packets have to be rerouted by B over A and D towards C. Thus, B→A, A→D and D→C are added to the header. If we do so without clearing additional bit, the BIER header for multicast tree (1) still contains at node B the `local_decap`(D). As a consequence, the packet will be delivered to C and D. However, the packet is also directly delivered from the source A to D. Thus, `local_decap`(D) should be cleared in the header of the rerouted packet. This is different for multicast

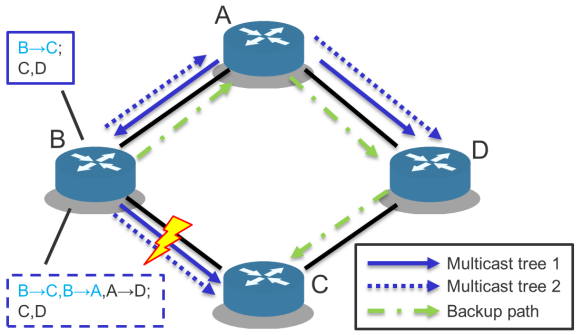

Fig. 4: If A→D fails, clearing the `local_decap`(D) bit at B prevents a duplicate packet at D for multicast tree (1) but causes packet loss at D for multicast tree (2).

tree (2) because BIER-TE sends only a single packet over each interface and must, therefore, encode the backup path in the BIER header. If the `local_decap`(D) is set before transmission at B towards A, the packet will be delivered to D, otherwise it will not be delivered so that D loses the packet although its reachability is not affected by the failure.

Thus, after the backup path is added to the packet header through the AddBitmask, a ResetBitmask must be applied before sending the packet to avoid duplicates which, however, may cause packet loss for other destination. The ResetBitmask contains both the `local_decap` bits of the nodes on the backup path and their outgoing adjacencies. The latter are needed to ensure that duplicates are not propagated into other multicast subtree if the backup path traverses them.

Although the HM method may lose some packets in case of failures, it is of interest because it avoids the overhead of an encapsulation header, it does not extend the BIER header by adding further adjacencies, and does not require support from a routing underlay.

4) Notation: Link and node protection {L,N} can be implemented with any of the presented protection methods for BIER-TE. We denote them by $\{\text{HM, PPT, BBE}\}_{\{\text{L,N}\}}$.

V. Results

In this section we evaluate key performance metrics of the discussed FRR mechanisms for BIER and BIER-TE. We first present the networks under study and explain our evaluation methodology. We quantify the effectiveness of the HM protection variants for BIER-TE. For the other protection methods we compare path lengths, consider resource requirements, and discuss state and header overheads.

A. Networks under Study

For our study, we leverage networks from the Topology Zoo [29] which contains research and commercial wide area and Internet service provider networks from mainly North America and Europe. We simplify the networks by consecutively removing all vertices that are attached to the network with only a single edge. We consider a network to have a ring structure if at least 60% of its nodes have node degree two, otherwise it has a mesh structure. This definition categorizes our 220 considered networks into 118 ring topologies T_R and 102 mesh topologies T_M.

The networks vary in size and their average numbers of nodes are 17.7 and 21.2 for T_R and T_M, respectively, the average numbers of unidirectional links are 22.9 and 33.4. Unlike BIER, BIER-TE encodes not only nodes but also links in the header. To that end, BIER-TE requires in ring topologies on average 40.7 bits in the header and at most 154 bits. In mesh topologies, BIER-TE requires on average 54.7 bits and at most also 143 bits.

B. Methodology

We analyze the forwarding behavior for all BIER variants in all 220 considered topologies. A topology is represented as a graph $G = (V, E)$. We investigate three different sets of failure scenario. First, the failure-free scenario (FF). Second, the set of single link failures (SLF). It contains all bidirectional single link failures and, thus, consists of $|E|$ scenarios. And third, the set of single node failures (SNF).

The Topology Zoo does not provide traffic models or matrices. Therefore, we define a traffic model that is suitable for a systematic evaluation of multicast protection mechanisms. Every node in the network is sender of a multicast tree which has all other nodes as receivers, and any node sends the same unit rate. There is no other traffic in the network.

We apply shortest path routing to construct multicast trees for BIER-TE, shortest path around links for link protection, and shortest path around nodes for node protection methods. The path layout for MoFRR for BIER leverages MRT routing topologies that are computed according to the lowpoint algorithm in [11].

C. Efficiency of BIER-TE Protection with Header Modification

As outlined in the previous section, BIER-TE FRR with HM may lose traffic in order to avoid duplicate packets. In the following, we quantify the average fraction of traffic that may get lost for HM in all SLF and SNF, respectively. Also the perfect FRR schemes PPT and BBE lose traffic under some conditions:

- The network is not two-connected. If a critical link or node fails, the network is disected so that senders in one part of the network cannot reach the senders in the other part of the network.
- In case of node failures, traffic from or to a failed (ingress/egress) node is lost.
- If link protection is applied, traffic cannot be protected in case of a node failure.

Therefore, we consider the traffic loss occurred for PPT_N as lower bound on avoidable traffic loss.

Figure 5 shows the cumulative distribution function (CDF) of the percentage of lost traffic over all networks. Every data point on a curve corresponds to one network. Note that the curves consist of many more data points than markers which only improve their readability.

In case of SLF, all traffic can be protected by perfect FRR methods in more than 90% of the networks. The remaining

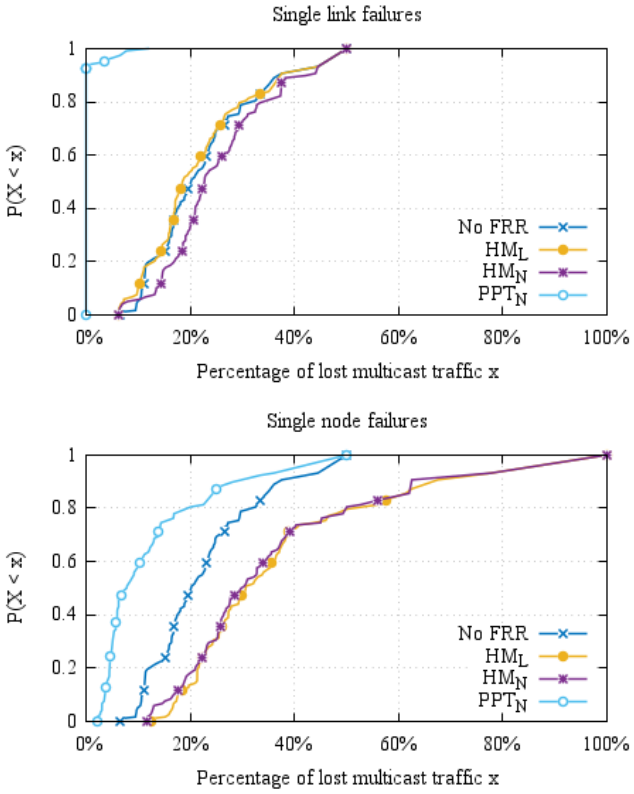

Fig. 5: CDF of the percentage of lost multicast traffic in all networks for SLF and SNF, respectively.

networks are not two-connected with regard to links so that the failure of a specific link disects the network. Another curve illustrates that between 7% and 50% of the traffic is lost without protection (No FRR). Surprisingly, HM methods cause about the same amount of lost traffic due to subtree pruning. When HM is configured to protect against node failures, it loses a bit more traffic than without protection while when it is configured to protect against link failures, it yields slightly less traffic loss than without protection.

In case of SNF, also the perfect FRR methods lose between 2% and 50% of the traffic in case of node failures, but at most 10% in 60% of the networks. This happens for the reasons given above and is unavoidable. Without protection, significantly more traffic is lost. The HM methods lead to even clearly more traffic loss than without protection. The reason for that node failures activate more backup paths than link failures so that more traffic is pruned from subtrees.

These results clearly demonstrate that HM is not efficient to protect against failures, yet it can be counterproductive in particular in case of node failures. Therefore, we exclude the HM methods from further discussions.

Nevertheless, HM can fully protect 20% – 40% of all multicast flows against SLF in 80% of the networks. This potential of HM could be further elaborated in future studies, which can be of interest to protect small multicast trees in some networks with only little technological complexity.

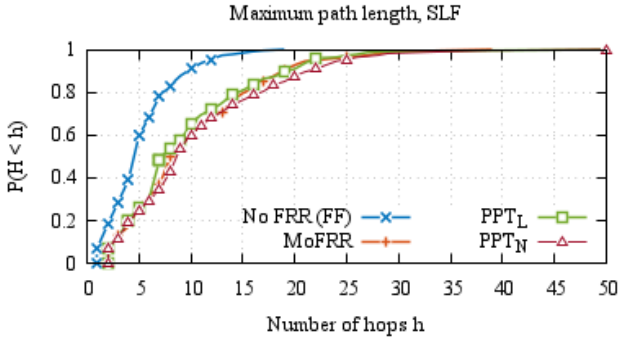

Fig. 6: CDF of maximum path lengths H for shortest path routing in the failure-free case and for BIER variants in SLF.

D. Path Lengths

The path layout depends on the applied routing mechanism and impacts path length in failure-free and failure cases. It is the same for PPT and BBE, but depends on link or node protection. From previous work [26] we know that MRTs may lead to excessive path length. Therefore, a comparison of the new resilience mechanisms with regard to path length is important.

We observe that most average path lengths are betwee 2 and 6 hops in the failure-free case. When we consider only average path lengths of affected multicast flows for all SLF, we mostly observe an average path stretch between 0.5 up to 1 hop for BIER-TE FRR mechanisms that bypass the traffic on shortest paths around the failure location. The values are rather small as the paths to some destinations are not extended. Most interesting is the finding that the path lengths for BIER MoFRR using MRTs are hardly longer than those for BIER-TE FRR although MRTs cause significant path stretch when used for the protection of unicast flows. This can be explained as follows. MRT routing topologies are used differently for multicast compared to unicast. Unicast flows are carried over shortest paths until a failure occurs and are then switched to a working MRT routing topology. In contrast, multicast flows are transmitted in parallel over both MRT routing topologies without any switching. During failure-free operation, we consider the length of the shortest of both paths, in case of a failure, we consider the length of the working path, which may be shorter than the failed path.

The longest path prolongation in failure cases can be significant. Figure 6 shows the CDF of the maximum path length over all networks in the failure-free scenario and for SLF scenarios. The longest paths are mostly twice as long as in the failure-free scenario. Again, BIER with MoFRR and BIER-TE with PPT lead to about the same maximum path lengths. In case of node protection, path lengths are slightly longer than in case of link protection because traffic for NNHs is explicitly bypassed around the NH which may extend the backup path length.

Thus, MRT-based MoFRR for BIER does not cause excessive path length compared to shortest path routing, but backup paths of any FRR mechanism can lead to significant path

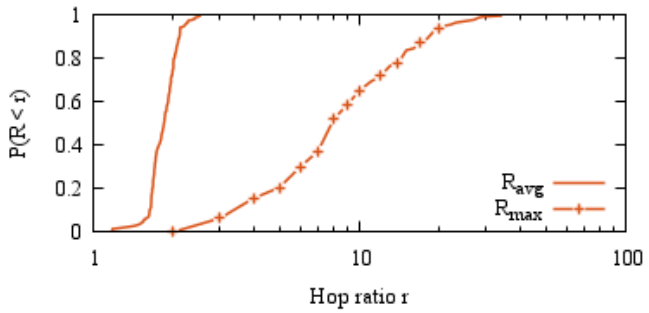

Fig. 7: CDF of the average and maximum ratios of lengths of redundant paths for MoFRR in all networks.

stretch – not on average, but in the worst case.

E. Difference in Path Lengths for MoFRR

With MoFRR, egress routers measure the quality of the traffic stream received over the two independent paths and choose the traffic from the one with highest quality. In case of a failure, they detect the failure by the fact that the signal from one path is lost and choose the signal from the remaining path. The detection is faster and the switch-over smoother if packets from both paths are received simultaneously by the egress node or with only little delay difference. Delay difference may result from different path lengths. We quantify different path lengths by the ratio R of the longer and shorter length of the two redundant paths over the two MRT routing topologies.

Figure 7 shows the CDF of the average ratio $R_{avg}$and the maximum ratio R_{max}over all considered network topologies. The average ratio is about 2 for most topologies, i.e., mostly one of the path is twice as long as the other path. Maximum ratios are significantly larger. For 60% of the topologies, the maximum ratio is between 2 and 10, and for the other networks we observe maximum ratios between 10 and 33. As a result, the delay difference of both redundant paths for MoFRR can be significant for some leaves of some multicast trees. This makes failure detection more difficult than for simultaneously received signals, requires more buffer, and may cause more jitter or packet loss in case of a switch-over.

F. Load and Capacity Analysis

MoFRR for BIER duplicates all traffic at the source. PPT_N for BIER-TE sends traffic over multiple unicast tunnels to DS-NNHs in case of node protection while BBE_N for BIER-TE uses multicast for that purpose. This observation calls for an analysis of traffic loads and required transmission capacities.

We first consider the average load in the network, i.e., we summarize the rates of all links in the network in particular failure scenarios and average over all of them. To report load values from differently large networks in one figure, we normalize the average network load by the average network load in the failure-free case for shortest path routing which represents a lower bound. This yields an average network load relative to shortest path routing without failures that we call relative average network load. Figure 8 shows CDFs of the relative average network loads for SLF over all networks for

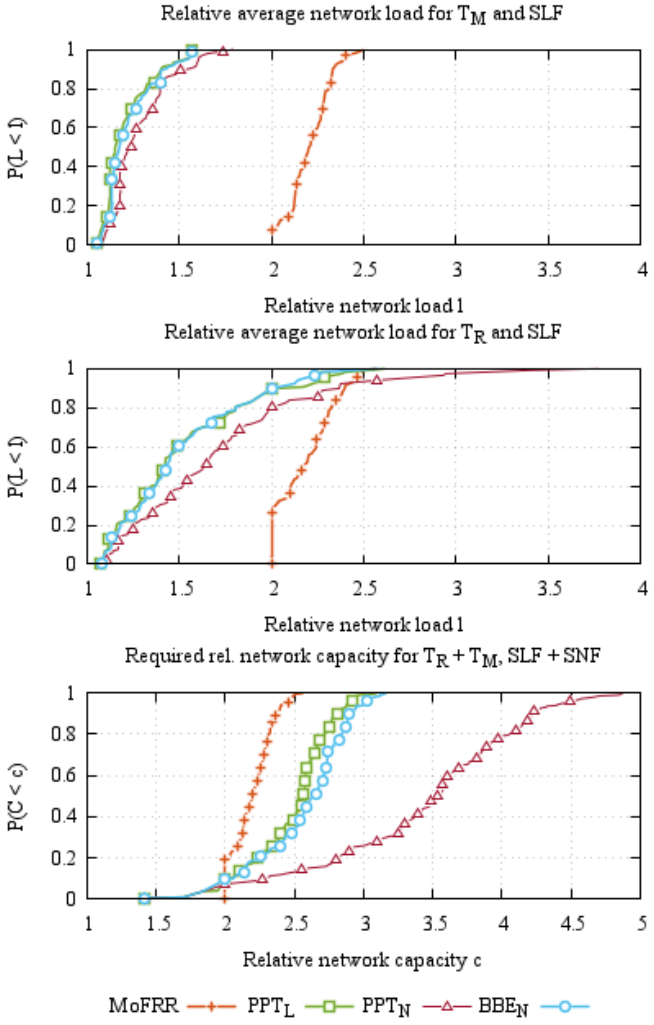

Fig. 8: CDFs of relative average network loads (for different network topologies) and relative network capacities.

BIER with MoFRR, PPT for BIER-TE with link and node protection, respectively, and for BBE for BIER-TE with node protection. One figure provides results for mesh topologies and another for ring topologies. BIER-TE protection methods cause in most mesh topologies an average load increase of up to 50%, but in most ring topologies a load increase of even up to 100% because more traffic is affected by failures and backup paths are longer in ring topologies than in mesh topologies. PPT with node protection causes more network load than with link protection because the PLR replicates the traffic and sends it over separate point-to-point tunnels to the DS-NNHs. This is unlike for BBE with node protection which leads to hardly more network load than PPT or BBE with link protection. With MoFRR and BIER we observe at least twice the relative average network load as in failure-free scenarios, and in some networks slightly larger values. The latter may be surprising with the notion of two redundant MRT-based multicast trees for MoFRR in mind. But this believe is wrong as MRT routing topologies do not necessarily form trees in some networks (cf. Section III-B).

Second, we investigate the required network capacity to cover all SLF and SNF in a network. We compute the maximum traffic load per link over all considered failure scenarios and summarize these maximum rates of all links in a network. For easier comparison, we normalize the required network capacity by the network load in the failure-free scenario with shortest path routing and obtain a relative required network capacity. The third chart of Figure 8 illustrates the CDF of the relative required network capacity in all networks. Surprisingly, BIER with MoFRR requires by far the least capacity in most networks. BIER-TE variants require more capacity because traffic may be routed on very different paths depending on the specific failure so that high traffic rates can occur on many links. This is different with BIER with MoFRR: in case of a failure, traffic is not rerouted but dropped so that the required capacity for a network equals its traffic load in the failure-free scenario. BIER-TE with PPT and node protection requires very large amounts of transmission resources, mostly 70% more than other BIER-TE variants and 2 – 3 times as much as BIER with MoFRR. The reason is that with PPT and node protection a PLR deviates traffic towards multiple DS-NNHs by unicast instead of multicast. This causes very large rates on backup paths for which capacity must be provided.

G. State and Header Overhead Considerations

BIER and BIER-TE require additional routing tables whose contents depends only on the topology and the routing in the underlay but not on supported multicast traffic which is good for scalability.

BIER without FRR requires only a single routing plane, e.g., shortest path routing. In contrast, BIER with MoFRR based on MRT routing topologies requires two additional routings planes whose state information scales with the number of nodes in the network. MRTs are adopted in the IETF and provide acceptable scaling in ISP networks [11]. They increase the routing state of normal IP routing by approximately 200%. Also the state information for BIER routing tables increases linearly with the number of nodes. The BIER header requires one bit for each node in the network. The default size of the BIER header is 256 bits (32 bytes) and was sufficient for all considered networks. Larger headers of up to 4096 bits are possible.

BIER-TE without FRR requires one bit per node in the network and one bit per link, which causes larger header overhead than BIER. Moreover, a controller is needed that constructs the multicast trees and encodes them in the packet headers. The FRR methods BBE and PPT have different scaling properties. BBE requires two BIER headers which results into an additional 32 bytes header overhead in our study because all networks were small enough to encode their links and nodes in a 256 bits header. PPT leverages point-to-point tunnels provided by a routing underlay. These tunnels need to be encoded as forward-adjacencies in the BIER header so that even more bits are needed. Moreover, the tunnels need to be provided by the routing underlay, increasing its complexity. To protect against link failures, every unidirectional link in the network has to be protected so that the number of additional bits equals the number of unidirectional links. The networks under study had 56 links on average and at most 176 links. To protect against node failures, every node needs to be protected by additional tunnels. In the networks under study, 156 tunnels were needed on average and at most 716 tunnels. Thus, for the largest network, the BIER header requires for PPT 892 additional bits (112 bytes) for unicast tunnels. This causes more overhead than a small 32 bytes encapsulation header for BBE. Moreover, tunneling over the routing underlay may also add some header overhead. Therefore, BBE may be the most efficient FRR melthod for BIER-TE in terms of complexity and header overhead.

As BIER represents only nodes in the headers while BIER-TE represents both links and nodes, BIER-TE leads to larger header overhead than BIER which may be relevant for very large networks so that the resulting header size may be technically still feasible but not efficient. When fast protection is required, the complexity of the routing underlays may be problematic for BIER MoFRR because it requires three different routing planes whereas BIER-TE with BBE does not even depend on a routing underlay.

VI. Conclusion

In this work we proposed fast reroute (FRR) mechanisms for BIER and BIER-TE and compared their performance on 220 network topologies. We suggested to implement MoFRR for BIER leveraging the calculation of MRT topologies to obtain redundant multicast forwarding structures. For BIER-TE FRR we suggested three different methods: header modification (HM), unicast tunneling (PPT), and BIER-TE-in-BIER-TE tunneling (BBE).

BIER MoFRR implements 1+1 protection and, therefore, causes at least double traffic load during operation compared to without protection. However, it requires mostly less additional capacity than BIER-TE-FRR because it does not reroute traffic in failure cases. Although the orignal MRT method is known for possibly long backup paths, path lengths resulting from BIER with MoFRR do not exhibit significantly more path stretch than with BIER-TE-FRR. An implementation challenge may be a possibly large difference in length of the two redundant paths. Below the line, MoFRR leveraging MRT calculation seems an effective FRR option for BIER.

HM is the simplest method for BIER-TE FRR in terms of technology. However, it is not effective because it loses about the same amount of traffic as without protection just to avoid duplicate packets. In contrast, PPT and BBE can protect against all failures if topologically possible. PPT configured for node protection may lead to excessive capacity requirements, large header overhead, and complicates the routing underlay. As BBE avoids these drawbacks, requires only an additional BIER header of moderate size, and does not need support from the routing underlay, we recommend BBE as preferred protection method for BIER-TE.

REFERENCES

[1] G. Shepherd, A. Dolganow, and arkadiy.gulko@thomsonreuters.com, "Bit Indexed Explicit Replication (BIER) Problem Statement," Internet Engineering Task Force, Internet-Draft draft-ietf-bier-problem-statement-00, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-bier-problem-statement-00

[2] C. Bestler, N. Kumar, R. Asati, M. Chen, X. Xu, A. Dolganow, T. Przygienda, arkadiy.gulko@thomsonreuters.com, D. Robinson, and V. Arya, "BIER Use Cases," Internet Engineering Task Force, Internet-Draft draft-ietf-bier-use-cases-04, Jan. 2017, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-bier-use-cases-04

[3] I. Wijnands, E. C. Rosen, S. Aldrin, T. Przygienda, and A. Dolganow, "Multicast using Bit Index Explicit Replication," Internet Engineering Task Force, Internet-Draft draft-ietf-bier-architecture-05, Oct. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-bier-architecture-05

[4] T. Eckert, G. Cauchie, W. Braun, and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE," Internet Engineering Task Force, Internet-Draft draft-eckert-bier-te-arch-04, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-eckert-bier-te-arch-04

[5] ——, "Fast ReRoute (FRR) Extensions for BIER-TE," Internet Engineering Task Force, Internet-Draft draft-eckert-bier-te-frr-00, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-eckert-bier-te-frr-00

[6] W. Zhang, G. Xue, J. Tang, and K. Thulasiraman, "Faster Algorithms for Construction of Recovery Trees Enhancing QoP and QoS," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 642–655, Jun. 2008.

[7] N. Taft-Plotkin, B. Bellur, and R. Ogier, "Quality-of-Service Routing using Maximally Disjoint Paths," in *International Workshop on Quality of Service*, 1999, pp. 119–128.

[8] Y. Guo, F. A. Kuipers, and P. V. Mieghem, "Link-disjoint Paths for Reliable QoS Routing," *Int. J. Communication Systems*, vol. 16, no. 9, pp. 779–798, 2003.

[9] S. Cho, T. Elhourani, and S. Ramasubramanian, "Independent Directed Acyclic Graphs for Resilient Multipath Routing," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 153 –162, Feb. 2012.

[10] G. Enyedi, G. Retvari, P. Szilagyi, and A. Csaszar, "IP Fast ReRoute: Lightweight Not-Via," in *IFIP-TC6 Networking Conference (Networking)*, Aachen, Germany, May 2009.

[11] G. S. Envedi, A. Csaszar, A. Atlas, C. Bowers, and A. Gopalan, "An Algorithm for Computing IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)," RFC 7811, Jun. 2016.

[12] H. Kirrmann, M. Hansson, and P. Muri, "IEC 62439 PRP: Bumpless Recovery for Highly Available, Hard Real-Time Industrial Networks," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, Sept 2007, pp. 1396–1399.

[13] H. Heine and O. Kleineberg, "The High-Availability Seamless Redundancy Protocol (HSR): Robust Fault-Tolerant Networking and Loop Prevention Through Duplicate Discard," in *IEEE International Workshop Factory Communication Systems (WFCS)*, May 2012, pp. 213–222.

[14] S. Paul, K. K. Sabnani, J. C. H. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, Apr 1997.

[15] B. N. Levine, D. B. Lavo, and J. J. Garcia-Luna-Aceves, "The Case for Reliable Concurrent Multicasting Using Shared ACK Trees," in *ACM International Conference on Multimedia*, 1996, pp. 365–376.

[16] Y. Ofek and B. Yener, "Reliable Concurrent Multicast From Bursty Sources," in *IEEE Infocom*, vol. 3, Mar 1996, pp. 1433–1441.

[17] D. Frost, S. Bryant, M. Bocci, and L. Berger, "A Framework for Point-to-Multipoint MPLS in Transport Networks," RFC 7167, Oct. 2015.

[18] S. Y. (Ed.), "RFC4655: Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)," Apr. 2006.

[19] R. Fabregat, Y. Donoso, B. Baran, F. Solano, and J. L. Marzo, "Multi-objective Optimization Scheme for Multicast Flows: A Survey, a Model and a MOEA Solution," in *IFIP/ACM Latin American Conference on Networking*, 2005, pp. 73–86.

[20] Y. Nakagawa, K. Hyoudou, and T. Shimizu, "A management method of ip multicast in overlay networks using openflow," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 91–96.

[21] X. Li and M. J. Freedman, "Scaling IP Multicast on Datacenter Topologies," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 61–72.

[22] J. Ruckert, J. Blendin, R. Hark, and D. Hausheer, "DYNSDM: Dynamic and Flexible Software-Defined Multicast for ISP Environments," in *International Conference on Network and Services Management (CNSM)*, Nov 2015, pp. 117–125.

[23] A. Karan, C. Filsfils, I. Wijnands, and B. Decraene, "Multicast only Fast Re-Route," https://tools.ietf.org/html/rfc7431, Aug. 2015.

[24] G. S. Envedi, A. Atlas, and C. Bowers, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)," RFC 7812, Jun. 2016.

[25] A. Atlas, R. Kebler, I. Wijnands, A. Csaszar, and G. Enyedi, "An Architecture for Multicast Protection Using Maximally Redundant Trees," http://tools.ietf.org/id/draft-atlas-rtgwg-mrt-mc-arch, Mar. 2012.

[26] M. Menth and W. Braun, "Performance Comparison of Not-Via Addresses and Maximally Redundant Trees (MRTs)," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ghent, Belgium, Apr. 2013.

[27] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing Architecture," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-10, Nov. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-10

[28] E. C. Rosen, A. Viswanathan, and R. Callon, "RFC3031: Multiprotocol Label Switching Architecture," Jan. 2001.

[29] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765 –1775, Oct. 2011.