Comparison of Fast-Reroute Mechanisms for BIER-Based IP Multicast

Daniel Merling, Steffen Lindner, Michael Menth
Chair of Communication Networks, University of Tuebingen, Tuebingen, Germany
{daniel.merling, steffen.lindner, menth}@uni-tuebingen.de

***Abstract*—IP multicast (IPMC) delivers one-to-many traffic along distribution trees. To that end, conventional IPMC requires state in forwarding devices for each IPMC group. This limits scalability of IPMC because forwarding state in core devices may be extensive and updates are necessary when IPMC groups or the topology change. The IETF introduced Bit Index Explicit Replication (BIER) for efficient transport of IPMC traffic. BIER leverages a BIER header and IPMC-group-independent forwarding tables for forwarding of IPMC packets in a BIER domain. However, legacy devices do not support BIER. In contrary, two SDN-based implementations for OpenFlow an P4 have been published recently. In this paper, we assess BIER forwarding which may be affected by network failures. So far there is no standardized procedure to handle such situations. Two concepts have been proposed. The first approach is based on Loop-Free Alternates. It reroutes traffic to suitable neighbors in the BIER domain to steer traffic around the failure. The second approach is a tunnel-based mechanism that tunnels BIER packets to appropriate downstream nodes within the BIER distribution tree. We explain and compare both approaches, and discuss their advantages and disadvantages.**

***Index Terms*—Software-Defined Networking, Bit Index Explicit Replication, Multicast, Resilience, Scalability**

I. Introduction

IP multicast (IPMC) is used for services like IPTV, commercial stock exchange, multicast VPN, content-delivery networks, or distribution of broadcast data. Figure 1 shows the concept of IPMC.

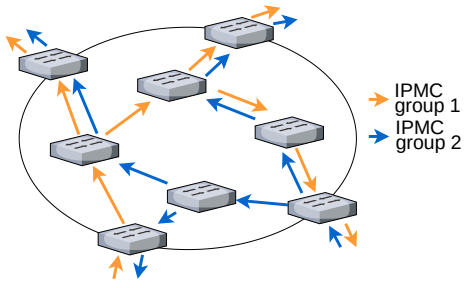

Figure 1: Two multicast distribution trees.

IPMC efficiently distributes one-to-many traffic by replicating packets and forwarding only one packet per link. Hosts join an IPMC group to receive the traffic addressed to that group. Forwarding devices maintain IPMC-group-dependent state to forward packets to the right neighbors. This decreases the scalability of IPMC for the following reasons. First, a large number of IMPC groups require a significant amount of forwarding state in core devices. Second, when subscribers of an IPMC group change, i.e., devices join or leave the group, the forwarding state needs to be updated. Third, when the topology changes or in case of a failure, the forwarding information base of possibly many devices has to be adapted.

The IETF presented BIER [1] as an efficient transport mechanism for IPMC traffic. BIER introduces a BIER domain, where only ingress routers maintain IPMC-group-dependent state. Ingress routers of the BIER domain encapsulate IPMC packets with a so-called BIER header which contains the destinations of the packet. Within the BIER domain, BIER packets are forwarded along distribution trees from the source to the destinations. Thereby only a single packet is transmitted per link. Finally, egress nodes remove the BIER header. Forwarding in the BIER domain is based on two components. First, the BIER header which contains a bit string that identifies receivers of a packet within the BIER domain. Second, the so-called Bit Index Forwarding Table (BIFT) which is the routing table of BIER devices. The entries of the BIFT are derived from information from the routing underlay, e.g., the Interior Gateway Protocol (IGP).

When a primary next-hop (NH) is unreachable due to a failure, an entire set of downstream destination nodes does not receive the traffic. When a failure is detected, IGP converges, new distribution trees are calculated, and the BIFTs are updated. This process requires a significant amount time. Therefore, BIER would benefit greatly from a fast protection mechanism that delivers traffic in the meantime. For unicast, several fast reroute (FRR) mechanisms [2] have been proposed which protect against the failure of single links or nodes until the forwarding information base is updated. FRR mechanisms use pre-computed backup entries to quickly reroute traffic when the primary NH is unreachable. No signaling between devices is necessary. Two FRR concepts for BIER have been proposed. First, LFA-based BIER-FRR [3] leverages a FRR mechanism called Loop-Free Alternates (LFAs) [4] that has been initially proposed for IP unicast. Failures are bypassed by forwarding traffic to alternative BIER NHs. Second, tunnel-based BIER-FRR tunnels traffic through the routing underlay, leveraging its FRR capabilities to steer traffic around the failure. We proposed this mechanism at the IETF [5].

However, legacy devices do not support BIER. On the contrary, the flexibility of SDN-based technologies have been leveraged recently to successfully implement BIER with OpenFlow [6] and in P4[1]. This allows the deployment of BIER

The authors acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG) under grant ME2727/1-2. The authors alone are responsible for the content of the paper.

[1] https://github.com/uni-tue-kn/p4-bier

and facilitates the implementation of additional BIER-related features, e.g. BIER-FRR.

In this paper we review LFA-based and tunnel-based BIER-FRR. First, we propose changes to tunnel-based BIER-FRR to reduce the number of forwarding entries. Then, we point out major shortcomings of the LFA-based approach and present extensions to resolve the issues. Further, we compare both mechanisms by discussing their protection capabilities, and overhead in terms of header size and forwarding state.

The paper is structured as follows. Section II describes related work for conventional and SDN-based multicast, and BIER. We review BIER in Section III. Section IV gives a primer on LFAs. Then, in Section V we explain tunnel-based BIER-FRR. Afterwards, we describe LFA-based BIER-FRR in Section VI, and point out its shortcomings and propose extensions in Section VII. Finally, we compare and discuss both approaches in Section VIII. We conclude the paper in Section IX.

II. Related Work

In this section we first discuss related work for conventional and SDN-based multicast. Afterwards, we review related work for BIER.

A. Multicast

In [7] the authors provide an overview of the early development of multicast. The authors of [8] discuss the limited scalability of conventional IP multicast in terms of the number of forwarding entries. They propose an extension to the multicast routing protocol MOSPF to reduce the number of required forwarding entries. Li et al. [9] propose an architecture to partition the multicast address space to increase scalability of IP multicast in data center topologies.

B. SDN-Based Multicast

The surveys [10], [11] provide a detailed overview of SDN-based multicast. We discuss only some of the mentioned papers. The authors of [12] introduce software-defined multicast (SDM), an OpenFlow-based approach that aims at providing a well-managed multicast platform for over-the-top and overlay-based live streaming services. SDM is specifically engineered for the needs of P2P-based video stream delivery. They further develop their idea of SDM in [13] by adding support for fine-granular traffic engineering capabilities. Lin et al. [14] present a multicast model to construct so-called multi-group shared trees. By deploying distribution trees that cover multiple multicast groups simultaneously, the entire network is covered with a small number of trees.

C. Protection of SDN-Based Multicast

Kotani et al. [15] propose to leverage multiple simultaneously deployed multicast trees for protection. An ID in the packet header determines along which distribution tree a packet is forwarded. When a tree is affected by a failure, the controller reconfigures the senders to forward traffic on a backup tree. The authors of [16] follow a similar approach where they leverage primary and backup trees identified by a VLAN tag. When a switch detects a failure, it reroutes the packets on a working backup tree that contains all downstream nodes. This is accomplished by switching the VLAN tag in the packet header.

D. BIER Related Work

Giorgetti et al. [6], [17] provide an implementation for both, conventional IPMC and BIER forwarding in OpenFlow. They leverage MPLS headers to encode the BIER bit string, which limits the bit string length, and thereby the number of destinations, to a maximum of 20. However, a local BIER agent is required to run on the switches to support arbitrary destinations. BIER-TE [18] extends BIER with traffic engineering capabilities. BIER-TE leverages the same header format as BIER and supports explicit coding of a distribution tree in the BIER header. However, BIER and BIER-TE are not compatible. The authors of [19] present a P4-based implementation of BIER and BIER-TE and present different demo scenarios to show the feasibility and the advantages of BIER(-TE). The authors of [20] propose 1+1 protection for BIER-TE. Traffic for each IPMC group is forwarded on two disjoint distribution trees simultaneously. The trees share as few network components as possible to still deliver traffic when one tree is interrupted by a failure. However, the approach requires two forwarding planes, and in the failure free case twice the amount of network resources are occupied.

III. Bit Index Explicit Replication (BIER)

The following section reviews BIER [1]. First, we describe its concept, the structure of the Bit Index Forwarding Table (BIFT), the BIER forwarding procedure, and a forwarding example. Afterwards we explain a compact representation of the BIFT, and characteristics of the BIER topology.

A. BIER Concept

BIER is based on a layered architecture, consisting of routing underlay, BIER layer, and IPMC layer. Figure 2 illustrates the relation between these components.

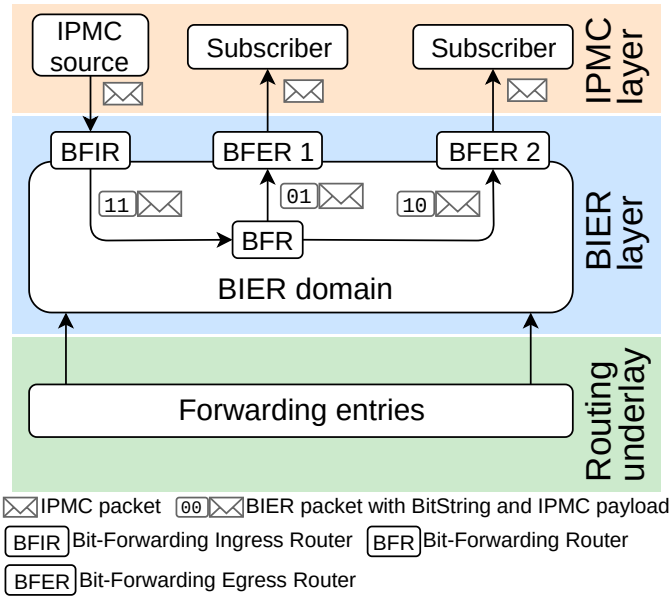

Figure 2: Layered architecture of BIER; it shows the relation between routing underlay, BIER layer, and IPMC layer.

The BIER layer serves as a point-to-multipoint tunnel for IPMC traffic through a BIER domain. The BIER domain consists of bit forwarding ingress routers (BFIRs), bit forwarding

routers (BFRs), and bit forwarding egress routers (BFERs). A BIER-capable device can be BFIR, BFR and BFER at the same time. When an IPMC packet enters the domain, the BFIR pushes a BIER header onto the IPMC packet. The BIER header identifies all receivers (BFERs) of the packet within the BIER domain. To that end, it contains a bit string which has to be at least as long as the number of BFERs in the BIER domain. In the following, 'BitString' refers to the bit string in the BIER header of the packet. Each BFER is assigned to a bit position in the BitString, starting with the least-significant bit. An activated bit means that the corresponding BFER must receive a copy of the BIER packet. BFRs forward BIER packets according to theor BitString along distribution trees to multiple BFERs.

Paths in the BIER domain are derived from the routing underlay, e.g., the IGP. As a consequence, BIER traffic follows the same paths as the corresponding unicast traffic from source to destination. At the domain boundary, BFERs remove the BIER header and pass the IPMC packet to the IPMC layer.

B. BIFT Structure

Table 1 shows the BIFT of BFR 1 from Figure 3. For each BFER, the BIFT contains one forwarding entry that consists of the primary NH and the so-called Forwarding Bit Mask (F-BM). The F-BM is a NH-specific bit string similar to the bit string in the BIER header. It indicates the BFERs with the same NH. In one particular F-BM, only bits of BFERs that are reached over the same NH are activated. During forwarding, BFRs use the F-BM to clear bits from the BitString.

C. BIER Forwarding

When a BFR receives a BIER packet, it stores its BitString to account to which BFERs the packet needs to be sent. We refer to that stored bit string by the term 'remaining bits'. The following procedure is repeated until the remaining bits do not contain any activated bits anymore.

The BFR determines the least-significant activated bit in the remaining bits. This bit indicates the BFER to be processed. Then, the BFR performs a looks up in the BIFT to get the NH and F-BM for that BFER. After a successful match, the BFR creates a copy of the received BIER packet. The BFR clears the BFERs from the BitString of the packet copy that have a different NH. To that end, the BFR performs a bitwise AND operation of the F-BM and the BitString of the packet copy. Then the BFR writes the result into the BitString of the packet copy. This procedure is called applying the F-BM. Thus, only bits that correspond to BFERs which share the same primary NH remain active in the BitString of the packet copy. Clearing other bits avoids duplicates at the receivers. Afterwards, the packet copy is forwarded to the NH. Finally, the BFERs, to which a packet has just been sent, are removed from the remaining bits. To that end, a bitwise AND operation of the bitwise complement of the F-BM and the remaining bits is performed.

D. BIER Forwarding Example

Figure 3 shows an example topology with four BFRs. Each BFR is in addition a BFIR and a BFER. Table 1 shows the BIFT of BFR 1.

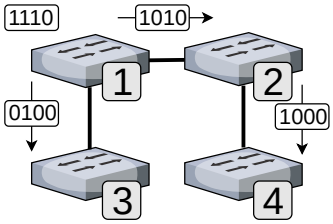

Figure 3: BIER topology and BitStrings of forwarded BIER packets.

BFER	NH	F-BM
1	-	-
2	2	1010
3	3	0100
4	2	1010

Table 1: BIFT of BFR 1.

BFR 1 receives a BIER packet with the BitString `1110`. The least-significant activated bit in the remainings bits identifies BFR 2. Therefore, BFR 1 creates a copy of the packet, applies the corresponding F-BM `1010`, and forwards the packet copy with the BitString `1010` to BFR 2. This sends a packet to BFER 2 and BFER 4. Afterwards, the bits of the F-BM are cleared from the remaining bits `0100`. The least-significant activated bit in the remaining bits corresponds to BFER 3. The F-BM is applied and a packet clone with the BitString `0100` is forwarded to the NH which is BFR 3. After clearing the F-BM from the remaining bits, processing stops because no active bits remain.

E. Compact BIFT

The number of entries of the BIFT scales with the number of BFERs. For improved scalability in terms of forwarding entries, the authors of [21] propose a compact representation of the BIFT that requires only one forwarding entry per neighbor. To that end, all entries with the same NH and F-BM are aggregated. As a result, all BFERs indicated in the F-BM share a single forwarding entry. During lookup, an entry is considered a match when at least one of the associated BFERs is a destination of the BIER packet. Table 2 shows the compact BIFT of BFR 1 from Figure 3.

BFERs	NH	F-BM
2, 4	2	1010
3	3	0100

Table 2: Compact BIFT of BFR 1.

F. Characteristics of the BIER Topology

In this paragraph we first discuss the impact of differences between the Layer 3 topology and BIER topology. Afterwards, we review how BIER devices detect whether BIER neighbors are still reachable.

1) Differences Between Layer 3 Topology and BIER Topology: In a Layer 3 topology some Layer 3 devices may not be BIER capable. Thus, the BIER topology may be different from the Layer 3 topology. Neighbors in the BIER topology are either connected directly to each other, or through at least one intermediate Layer 3 device that is no BIER device. BIER nodes receive information about their connection to their neighbors from the routing underlay. If two BIER neighbors are directly adjacent, they forward packets over Layer 2 to each other. If they are not directly adjacent, the BIER neighbors

leverage a Layer 3 tunnel to exchange packets. In both cases forwarding still follows the paths from the routing underlay.

2) Detection of Unreachable NHs: To quickly detect unreachable BIER neighbors, the authors of [22] propose bidirectional forwarding detection (BFD) [23] for BIER. When a BFD is established between two BIER nodes, they periodically exchange notifications to observe the reachability.

IV. Loop-Free Alternates

In this section we explain the concept of Loop-Free Alternates (LFAs) [4]. Afterwards, we review extensions for improved protection capabilities and loop detection.

A. Foundations of LFAs

LFAs implement a FRR mechanism for IP unicast traffic that prevents rerouting loops. Figure 4 shows the concept of LFAs.

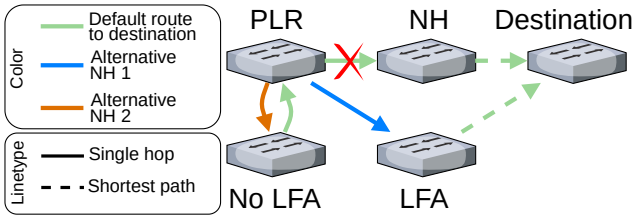

Figure 4: Concept of LFAs.

When a node cannot reach a primary NH, it acts as point of local repair (PLR), i.e., it leverages a pre-computed backup entry to reroute the packet via an alternative NH on a backup path towards the destination. Such neighbors are called LFAs and they have to be chosen in a way that rerouting loops are avoided. Some neighbors must not be chosen as LFAs because rerouting the packet would result in a forwarding loop.

LFAs have different properties for protection and loop avoidance. Some protect against link failures, others against node failures. Link-protecting LFAs (LP-LFAs) have a shortest path towards the destination that does not include the link between PLR and primary NH. Thus, LP-LFAs protect against the failure of the link between PLR and primary NH. The authors of [24] and [25] analyze the protection capabilities of LP-LFAs with a comprehensive set of topologies. They find that LP-LFAs protect only 70% of destinations against single link failures. Furthermore, LP-LFAs may cause loops when at least one node or multiple links fail instead of a single link only. To protect against the failure of the primary NH, node-protecting LFAs (NP-LFAs) have a shortest path to the destination that does not include the primary NH. In [24] the authors evaluate NP-LFAs in different scenarios on a large set of topologies. They show that NP-LFAs prevent loops for single link and single node failures, but they protect only 40% of destinations against single link failures.

B. Extensions for LFAs

In this paragraph we explain remote LFAs (rLFAs), topology independent LFAs (TI-LFAs), and explicit LFAs (eLFAs) to complement LFAs for increased protection capabilities. All three LFA variants support link and node protection. We indicate the protection mode with the prefix 'LP-' for link protection, and 'NP-' for node protection. Furthermore, we review a loop detection mechanism for LFAs. Figure 5 shows the concept of rLFAs, TI-LFAs, and eLFAs, which we explain in detail in the following.

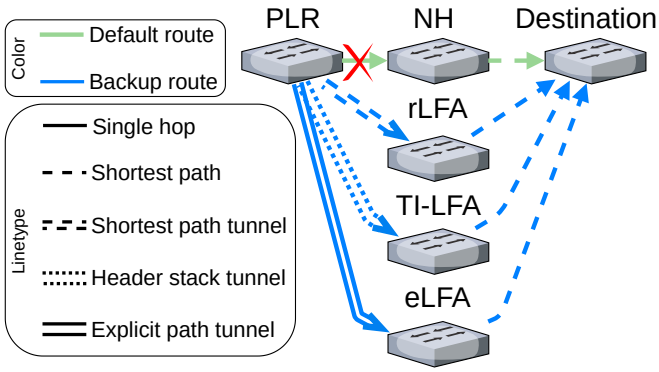

Figure 5: Concept of rLFAs, TI-LFAs, and eLFAs.

1) Remote LFAs (rLFAs): rLFAs [26] are remote nodes in the network. When the PLR cannot reach a primary NH, the packet is rerouted through a shortest path tunnel to the rLFA. From there, the packet is forwarded on a shortest path towards the destination. In [26] the authors prove that there is always a LP-rLFA to protect against a single link failure in unit-link-cost topologies. However, the authors of [25] find that this property does not hold for topologies with arbitrary link costs. NP-rLFAs cannot protect against all single link or single node failures.

2) Topology-Independent LFAs (TI-LFAs): TI-LFAs [27] are remote nodes in the network. When the primary NH is unreachable, the PLR leverages a header stack of IP headers to deviate traffic to the TI-LFA. The TI-LFA then sends the original packet on a shortest path towards the destination. As long as there is still a working shortest path to the destination, LP-TI-LFAs can protect against any single link failure, and NP-TI-LFAs against any single node failure.

3) Explicit LFAs (eLFAs): eLFAs [25] follow a similar concept as TI-LFAs. An eLFA is a remote node that serves as tunnel-end point when the PLR cannot reach the primary NH. The PLR reroutes the packet through an tunnel on an explicit path to the eLFA. The eLFA then forwards the packet on a shortest path to the destination. In contrast to TI-LFAs, eLFAs leverage additional forwarding entries for explicit paths to prevent an IP header stack. The authors of [25] evaluate eLFAs on a comprehensive set of different topologies. As long as the destinaton is still reachable, LP-eLFAs protect against any single link failure and NP-eLFAs protect against any single node failure.

4) Loop Detection: LFAs and all of its variants share the shortcoming that their deployment may cause forwarding loops [24], [25] in case of unprotected failures. In [24] the authors present a loop detection mechanism for LFAs. It is based on a bit string in the packet header where each forwarding device in the network is assigned a bit position. When a node needs to reroute a packet, it checks whether its own bit is activated. If this is not the case, the node activates the bit and reroutes the packet. However, if the bit is already activated, the packet

has been rerouted by the node before, and thus, the packet is dropped to prevent a loop. In [25] the authors describe loop detection for all LFA variants.

V. Tunnel-Based BIER-FRR

In this section we review tunnel-based BIER-FRR. We introduced this mechanism at the IETF [5]. First, we describe the concept, explain two modes of operation and an example. Then, we present changes to tunnel-based BIER-FRR for deployment with the compact BIFT. Finally, we discuss forwarding state.

A. Concept

When a BFR cannot forward a packet to a NH, the neighbor may still be reachable on a backup path. Tunnel-based BIER-FRR tunnels traffic through the routing underlay around the failure to BIER nodes downstream in the BIER distribution tree. A tunnel may be affected by the same failure but the routing underlay quickly restores connectivity with FRR mechanisms. With link protection, tunnel-based BIER-FRR tunnels the BIER packet to the NH. With node protection, BIER packets with adjusted BitStrings are tunneled to the next-next-hops (NNHs). Additionally, one BIER packet is tunneled to the NH to deliver a packet if only the link between PLR and NH failed.

Protection capabilities of tunnel-based BIER-FRR depend on the properties of the routing underlay. Tunnel-based BIER-FRR protects against any single component failure which can be handled by FRR mechanisms in the routing underlay. We describe the operation of tunnel-based BIER-FRR for link and node protection based on the normal BIFT.

1) Link Protection: Tunnel-based BIER-FRR with link protection does not require changes to the BIFT. When a primary NH is unreachable, the packet copy is tunneled to the NH instead of being forwarded on Layer 2. The routing underlay leverages IP-FRR to deliver the packet to the NH.

2) Node Protection: Tunnel-based BIER-FRR with node protection tunnels BIER packets to the NNHs. However, usually the NH adapts the BitString before the packet is forwarded to the NNH. Thus, before the packet is tunneled, the PLR performs modifications on the BitString that are usually done by the NH, i.e., applying the F-BM. To that end, backup entries in the BIFT are required which consist of a backup NH, and a backup F-BM. There are two categories of backup entries. First, for BFERs that are also NHs. In such backup entries, the NH is the backup NH and in the backup F-BM only the bit of the BFER is activated. This tunnels a packet to the NH in case only the link between PLR and NH failed. The second category of backup entries is for BFERs that are not NHs. For their entries, the backup NH is the NNH towards the BFER. The backup F-BM is the primary F-BM of the NH for the NNH.

When a primary NH is unreachable, the BFR performs three operations. First, the BFR applies the primary and the backup F-BM to the packet clone. The primary F-BM clears BFERs from the BitString that have a different NH. The backup F-BM clears BFERs from the BitString that have a different NNH. This leaves only bits of BFERs active that are activated in both, the primary and backup F-BM, i.e., all BFERs that have the same NH and the same NNH. Second, the packet copy is tunneled to the backup NH. Third, only bits that are active in both, the primary and backup F-BM are cleared from the remaining bits.

B. Forwarding Example

Figure 6 shows a BIER topology with a node failure where each BFR is also a BFIR and BFER. Table 3 displays the BIFT of BFR 1 with backup entries for node protection.

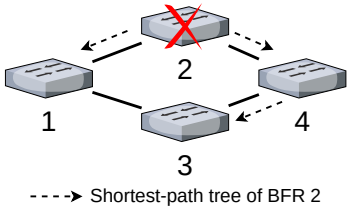

Figure 6: BIER topology with a node failure. The shortest-path tree of BFR 2 is shown to derive the backup F-BM of BFR 1 for BFER 4.

BFER	NH	F-BM
2	2	1010
	2	0010
3	3	0100
	3	0100
4	2	1010
	4	1100

Table 3: BIFT of BFR 1 with backup entries for node protection.

BFR 1 processes a packet with the BitString `1000`. The least-significant activated bit identifies BFER 4. However, the primary NH BFR 2 is unreachable. Thus, both, the primary F-BM `1010` and the backup F-BM `1100` are applied to the BitString of the packet copy. This leaves the BitString `1000` and the packet is tunneled to BFR 4 through the routing underlay. Bits that are activated in both, the primary and backup F-BM are cleared from the remaining bits which leaves `0000` and processing stops. The packet is eventually delivered by the routing underlay to BFR 4.

C. Compact BIFT

When the compact BIFT is used, tunnel-based BIER-FRR with link protection can be deployed as described in Section V-A1. Tunnel-based BIER-FRR with node protection requires two modifications. First, multiple backup entries are required for each primary forwarding entry. In the compact BIFT, each primary forwarding entry corresponds to one specific NH. For each NNH of the NH, one backup entry is required. The backup entries are calculated as described in Section V-A2. Second, when a BFR detects that a specific NH is unreachable, it matches incoming packets on the backup entries of the affected primary entry instead.

D. State Discussion

Tunnel-based BIER-FRR requires one backup entry for each primary entry. Therefore, in a topology with n BFERs the normal BIFT with backup entries contains $n + n$ forwarding entries. Deployment with the compact BIFT requires significantly fewer forwarding entries because the average

number of neighbors is significantly smaller than the number of destinations in a network. In a topology with an average node-degree of k, each node has k neighbors, and each NH has $k-1$ NHs on average. As a result the average number of forwarding entries per node is the sum of primary forwarding entries and backup entries $k + k \cdot (k-1)$.

VI. LFA-Based BIER-FRR

In this section we review LFA-based BIER-FRR [3]. We explain the concept, derivation of backup entries, and a forwarding example.

A. Concept

LFA-based BIER-FRR leverages backup entries in the BIFT to deviate traffic on backup paths when the primary path is interrupted. A backup entry consists of a backup NH, and a backup F-BM. When a primary NH is unreachable, further processing depends on the availability of a backup entry. If there is no backup entry, the bit of the BFER is cleared from the remaining bits and no packet is delivered to this particular BFER. Processing resumes with the next BFER. If there is a backup entry, further packet processing differs in three ways from regular BIER forwarding. First, the PLR applies the backup F-BM instead of the primary F-BM to the BitString of the packet clone. Second, the BIER packet is forwarded to the backup NH instead of the primary NH. Third, the bits of the backup F-BM instead of the primary F-BM are cleared from the remaining bits. Afterwards, the next BFER is processed.

B. Derivation of Backup Entries

We describe how we derive a backup entry consisting of a backup NH and a backup F-BM for a specific primary entry. First, we identify BIER neighbors that are LFAs as described in Section IV. LFA computation has to be performed on the BIER topology because Layer 3 LFAs may not be available on BIER layer due to topology differences. If no LFA is available, the primary forwarding entry remains without a backup entry. If there is an LFA L, it is selected as the backup NH. The activated bits in the backup F-BM are determined as follows. The bit that corresponds to an arbitrary BFER B is activated in the backup F-BM only if one of the two following conditions is fulfilled. First, L is an LFA to protect B. Second, L is the primary NH on the path to B. This aggregates all BFERs that are reached on a primary or backup path where L is the NH.

C. Forwarding Example

Figure 7 shows a BIER topology with a failed link between BFR 1 and 2. Each BFR is both a BFIR and a BFER. Table 4 contains the BIFT of BFR 1 with backup entries for link protection.

BFR 1 processes a BIER packet with the BitString `1110`. The least-significant activated bit identifies BFER 2. However, the primary NH BFR 2 is unreachable and there is no backup entry. Thus, the bit for BFER 2 is cleared from the remaining bits `1100` and no packet is sent. The next destination is BFER 3. Since the primary NH BFR 3 is reachable, the primary F-BM is applied and a packet clone with the BitString `0100` is forwarded to BFR 3. Clearing the F-BM from the remaining bits leaves only one bit activated `1000` which corresponds to BFER 4. However, the primary NH BFR 2 is unreachable. Thus, the backup F-BM is applied and a packet copy with the BitString `1000` is forwarded to the backup NH BFR 3. After the backup F-BM has been cleared from the remaining bits, no activated bits remain and processing stops. BFR 3 then forwards the packet to its destination BFR 4.

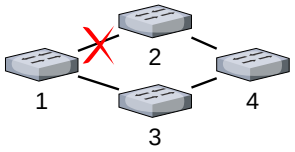

Figure 7: BIER topology with a link failure.

BFER	NH	F-BM
2	2	1010
	-	-
3	3	0100
	-	-
4	2	1010
	3	1100

Table 4: BIFT of BFR 1 with backup entries for link protection.

VII. Extensions for LFA-Based BIER-FRR

In this section, we expose major shortcomings of LFA-based BIER-FRR in terms of matching order, coverage, and forwarding state, and propose solutions. In the end we discuss scalability in terms of forwarding entries.

A. Matching Order

In the previous example two packets are forwarded to BFR 3. This is caused by the order in which receivers of a packet are processed. The following scenario describes when more than one packet is forwarded to one specific NH P. First, a packet is forwarded to the primary NH P towards a set of BFERs. Second, another BFER that should receive the packet is processed but its primary NH is unreachable. However, P is the backup NH. Thus, a second packet is forwarded to P on a backup path. To avoid sending multiple packets over one link, it is necessary to first process forwarding entries whose primary NH is unreachable. Then, no additional packet is sent because the backup F-BM aggregates primary and backup paths that have the same NH.

B. Coverage

Depending on the topology, LFAs cannot protect against arbitrary single component failures. rLFAs protect against any single link failure on unit-link-cost topologies. TI-LFAs and eLFAs guarantee protection against any single component failure on arbitrary topologies. However, the deployment of each of the three LFA extensions requires some sort of IP or segment routing tunnel. Nevertheless, full protection is an important property and we suggest to augment LFA-based BIER-FRR with rLFAs, TI-LFAs, or eLFAs to increase the coverage. rLFAs, TI-LFAs, and eLFAs need to be BFRs. Therefore, computations have to be performed on the BIER topology because not all Layer 3 devices may be BIER devices.

C. Compact BIFT

We explain scalability issues of LFA-based BIER-FRR and propose a solution that requires changes to how backup entries are derived.

1) Problem Statement and Solution: LFA-based BIER-FRR has been described for the BIFT that contains one primary forwarding entry per BFER. In its proposed form LFA-based BIER-FRR is incompatible with the compact representation of the BIFT, which requires only one primary entry per neighbor. In the following we describe the necessary changes to use LFA-based BIER-FRR with the compact BIFT.

We propose to use a default BIFT that does not contain any backup entries and is used for forwarding in the failure-free case. In addition, we use failure-specific backup BIFTs. When a BFR detects that a specific neighbor is unreachable, it matches incoming packets on the backup BIFT that is associated with the unreachable NH. When the failure has been repaired or forwarding entries are updated, the BFR continues matching on the default BIFT.

2) Derivation of Backup BIFTs: We explain how the backup BIFT for a specific neighbor N is derived in two steps. First we fill the BIFT with entries and afterwards activate bits in specific F-BMs. We start with an empty backup BIFT. In the first step, for each neighbor that is not N, the corresponding primary entry from the default BIFT is added to the backup BIFT. In the second step, for each BFER B whose primary NH is N, LFAs are identified on the BIER topology. If an LFA is available, the bit that corresponds to B is activated in the F-BM of the BFR that is the LFA. If no LFA is available, B cannot be protected.

D. State Discussion

In a topology with n BFERs the normal BIFT contains n primary forwarding entries. LFA-based BIER-FRR requires n additional backup entries, which totals in $n + n$ forwarding entries. In contrast, the compact BIFT contains only one forwarding entry for each neighbor. Therefore, when the average node degree is k, the compact BIFT requires on average only k primary forwarding entries. On average each node has k backup BIFTs with on average $k - 1$ entries, which results in $k + k \cdot (k - 1)$ forwarding entries. Since the average node degree is significantly smaller than the number of destinations in a network, scalability of the compact BIFT is considerably better.

VIII. Comparison of LFA- and Tunnel-Based Protection for BIER

In this section we compare LFA-based and tunnel-based BIER-FRR. We point out similarities, and analyze protection capabilities and overhead with regard to header size and forwarding state. Afterwards, we discuss the impact of differences between Layer 3 topology and BIER topology.

A. Similarities

Both approaches implement FRR for BIER for resilient transport of IP multicast. Forwarding devices need to detect unreachable NHs, e.g. through a BFD. Both FRR mechanisms are based on pre-computed backup entries in addition to the primary forwarding entries. It is not necessary to change the structure of the BIFT. When the PLR cannot reach a primary NH, affected packets are rerouted according to the backup entries. Two modes of operation for link and node protection with different protection properties are available. For both, LFA- and tunnel-based BIER-FRR it is necessary to augment the forwarding procedure of BIER.

B. Protection Capabilities

We compare coverage properties and occurrence of loops.

1) Coverage: Tunnel-based BIER-FRR is able to protect traffic against arbitrary single component failures by design when the routing underlay provides full FRR coverage. As long as the destination is still reachable, an IP or segment routing tunnel is deployed to deliver the traffic to the unreachable NH or NNHs.

Protection of LFA-based BIER-FRR depends on the topology. The authors of [24] evaluate LP- and NP-LFAs on a comprehensive set of topologies. They find that LP-LFAs protect only 70% of destinations against single link failures and cause loops when nodes fail. NP-LFAs avoid loops when a node fails, but protect only 40% of destinations against single link failures. LP-rLFAs protect against any single link failure on unit link cost topologies. For any further guarantees TI-LFAs, or eLFAs have to be deployed. Both LFA extensions guarantee full protection for any single component failure in the network. However, augmenting LFA-based BIER-FRR with rLFAs, TI-LFAs, or eLFAs requires an additional header. TI-LFAs require an IP header stack, eLFAs require additional forwarding entries to implement backup paths.

2) Loops: Tunnel-based BIER-FRR cannot cause loops on the BIER layer because the packet is tunneled to the backup NH. When the packet is successfully delivered at the backup NH, BIER forwarding continues. If the tunnel is interrupted, the routing underlay is responsible for avoiding loops.

LFA-based BIER-FRR cannot guarantee to avoid loops because depending on the failure scenario and the mode of operation, all LFA variants can cause loops [24], [25]. With link protection, traffic may loop if at least one node or multiple links fail. With node protection, loops are prevented as long as not multiple components fails. In Section IV-B4 we review a loop detection mechanism for LFAs and all variants to prevent loops in any failure scenario. However, this mechanism significantly increases operational complexity and modifications to the packet header are necessary.

C. Overhead

We compare both protection approaches according to packet header size and required forwarding state.

1) Header Size: Tunnel-based BIER-FRR requires tunneling to protect traffic against failures. This adds an additional header to the packet. When the tunnel is interrupted and the routing underlay leverages a tunnel-based FRR protection mechanism for unicast, e.g. TI-. or eLFAs, an additional header is added to the packet. The basic form of the LFA-based BIER-FRR approach does not require tunneling. However, rLFAs, TI-LFAs, or eLFAs increase the protection capabilities of LFAs to an appropriate level but require at least one additional IP

header. More header reduce the throughput and the Maximum Transmission Unit (MTU) has to be decreased at domain boundaries. The LFA-based approach requires a loop detection mechanism to prevent loops. Such a mechanism is available, however it increases packet header size even further.

2) Forwarding State: Both BIER-FRR approaches require the same amount of forwarding state. In a topology with n BFERs and an average node degree of k, the regular BIFT contains $n + n$ forwarding entries while the compact BIFT requires on average only $k + k \cdot (k - 1)$ entries. Since k is significantly smaller than n, deployment with the compact BIFT provides better scalability.

D. Influence of the BIER Topology

When some network nodes in a Layer 3 network do not support BIER, Layer 3 LFAs may disappear on the BIER layer. Thus, coverage of LFA-based BIER-FRR depends on the BIER topology. When regular LFAs have low coverage, LFA-based BIER-FRR needs to be complemented with rLFAs, TI-LFAs, or eLFAs. Backup paths may become longer in a sparse BIER topology because LFAs may be reachable only through a long Layer 3 tunnel. Tunnel-based BIER-FRR leverages tunnels through the routing underlay to the BIER NH or BIER NNHs for protection. Thus, tunnel-based BIER-FRR is not affected in a negative way by a BIER topology that is different from the Layer 3 topology.

IX. Conclusion

In this paper we compared LFA-based and tunnel-based BIER-FRR for resilient and scalable transport of IP multicast. Our discussion showed shortcomings of the LFA-based approach. Sometimes multiple packets are sent over one link, not all single link or single node failures can be protected, and in some scenarios backup traffic may loop. We propose extensions to overcome those shortcomings so that the capabilities of LFA-based and tunnel-based BIER-FRR mechanisms are equal. Differences remain in backup path length when the BIER topology is different from the Layer 3 topology, and in the need for additional headers.

References

[1] I. Wijnands, E. Rosen, A. Dolganow, T. Przygienda, and S. Aldrin, *RFC 8279: Multicast Using Bit Index Explicit Replication (BIER)*, https://datatracker.ietf.org/doc/rfc8279/, Nov. 2017.

[2] J. Papán, P. Segeč, M. Moravčík, M. Kontšek, L. Mikuš, and J. Uramová, "Overview of IP Fast Reroute solutions," in *ICETA*, 2019.

[3] I. Wijnands, G. J. Shepherd, C. J. Martin, and R. Asati, *Per-Prefix LFA FRR with Bit Indexed Explicit Replication*, https://patents.google.com/patent/US20180278470A1/en, Sep. 2018.

[4] G. Rétvári, J. Tapolcai, G. Enyedi, and A. Császár, "IP fast ReRoute: Loop Free Alternates revisited," in *IEEE INFOCOM*, 2011.

[5] D. Merling and M. Menth, *BIER Fast Reroute*, https://tools.ietf.org/html/draft-merling-bier-frr-00, Mar. 2019.

[6] A. Giorgetti, A. Sgambelluri, F. Paolucci, P. Castoldi, and F. Cugini, "First Demonstration of SDN-based Bit Index Explicit Replication (BIER) Multicasting," in *IEEE EuCNC*, 2017.

[7] K. C. Almeroth, "The Evolution of Multicast: From the MBone to Interdomain Multicast to Internet2 Deployment," *IEEE Network*, vol. 14, 2000.

[8] B. Zhang and H. T. Mouftah, "Forwarding State Scalability for Multicast Provisioning in IP Networks," *IEEE ComMag*, vol. 41, 2003.

[9] X. Li and M. J. Freedman, "Scaling IP Multicast on Datacenter Topologies," in *ACM CoNEXT*, 2013.

[10] S. Islam, N. Muslim, and J. W. Atwood, "A Survey on Multicasting in Software-Defined Networking," *IEEE Comst*, vol. 20, 2018.

[11] Z. AlSaeed, I. Ahmad, and I. Hussain, "Multicasting in Software Defined Networks: A Comprehensive Survey," *JNCA*, vol. 104, 2018.

[12] J. Rückert *et al.*, "Software-Defined Multicast for Over-the-Top and Overlay-based Live Streaming in ISP Networks," *JNSM*, vol. 23, 2015.

[13] J. Rückert, J. Blendin, and D. Hausheer, "Flexible, Efficient, and Scalable Software-Defined Over-the-Top Multicast for ISP Environments With DynSdm," *IEEE TNSM*, vol. 13, 2016.

[14] Y.-D. Lin, Y.-C. Lai, H.-Y. Teng, C.-C. Liao, and Y.-C. Kao, "Scalable Multicasting with Multiple Shared Trees in Software Defined Networking," *JNCA*, vol. 78, 2017.

[15] D. Kotani, K. Suzuki, and H. Shimonishi, "A Multicast Tree Management Method Supporting Fast Failure Recovery and Dynamic Group Membership Changes in OpenFlow Networks," *JIP*, vol. 24, 2016.

[16] T. Pfeiffenberger, J. L. Du, P. B. Arruda, and A. Anzaloni, "Reliable and Flexible Communications for Power Systems: Fault-tolerant Multicast with SDN/OpenFlow," in *IFIP NTMS*, 2015.

[17] A. Giorgetti, A. Sgambelluri, F. Paolucci, N. Sambo, P. Castoldi, and F. Cugini, "Bit Index Explicit Replication (BIER) Multicasting in Transport Networks," in *ONDM*, 2017.

[18] T. Eckert, G. Cauchie, W. Braun, and M. Menth, *Traffic Engineering for Bit Index Explicit Replication BIER-TE*, http://tools.ietf.org/html/draft-eckert-bier-te-arch, Nov. 2017.

[19] W. Braun, J. Hartmann, and M. Menth, "Demo: Scalable and Reliable Software-Defined Multicast with BIER and P4," in *IFIP/IEEE IM*, 2017.

[20] W. Braun, M. Albert, T. Eckert, and M. Menth, "Performance Comparison of Resilience Mechanisms for Stateless Multicast using BIER," in *IFIP/IEEE IM*, 2017.

[21] Z. Zhang and A. Baban, *Bit Index Explicit Replication (BIER) Forwarding for Network Device Components*, https://patents.google.com/patent/US20160191372, Dec. 2014.

[22] Q. Xiong, G. Mirsky, F. Hu, and C. Liu, *BIER BFD*, https://datatracker.ietf.org/doc/draft-hu-bier-bfd/, Oct. 2017.

[23] D. Katz and D. Ward, *Bidirectional Forwarding Detection (BFD)*, https://datatracker.ietf.org/doc/rfc5880/, Jul. 2004.

[24] W. Braun and M. Menth, "Loop-Free Alternates with Loop Detection for Fast Reroute in Software-Defined Carrier and Data Center Networks," *JNSM*, vol. 24, 2016.

[25] D. Merling, W. Braun, and M. Menth, "Efficient Data Plane Protection for SDN," in *IEEE NetSoft*, 2018.

[26] L. Csikor and G. Rétvári, "IP fast reroute with remote Loop-Free Alternates: The unit link cost case," in *ICUMT*, 2012.

[27] P. Francois, C. Filsfils, A. Bashandy, B. Decraene, and S. Litkowski, *Topology Independent Fast Reroute using Segment Routing*, https://tools.ietf.org/html/draft-francois-rtgwg-segment-routing-ti-lfa-00, Aug. 2015.