

BIER
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2018

P. Thubert, Ed.
Cisco
T. Eckert
Huawei
Z. Brodard
Ecole Polytechnique
H. Jiang
Telecom Bretagne
March 3, 2018

**BIER-TE extensions for Packet Replication and Elimination Function
(PREF) and OAM
draft-thubert-bier-replication-elimination-03**

Abstract

This specification extends Bit Index Explicit Replication - Traffic Engineering (BIER-TE) forwarding to support in the data plane the DetNet Packet Replication and Elimination Functions (PREF). It also provides traceability of links/adjacencies where replication and loss happen, in a manner that is agnostic to the forwarding information (OAM).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	On BIER - Traffic Engineering	3
4.	BIER-TE-based Replication and Elimination Control	4
5.	Elimination Function (Normative)	9
6.	Summary	11
7.	Implementation Status	12
8.	Security considerations	12
9.	IANA Considerations	12
10.	Acknowledgements	12
11.	References	13
11.1.	Normative References	13
11.2.	Informative References	13
	Authors' Addresses	14

[1.](#) Introduction

Deterministic Networking (DetNet) [[I-D.ietf-detnet-problem-statement](#)] provides a capability to carry unicast or multicast data flows for real-time applications with extremely low data loss rates and known upper bound maximum latency [[I-D.ietf-detnet-architecture](#)].

DetNet applies to multiple environments where there is a desire to replace a point to point serial cable or a multidrop bus by a switched or routed infrastucture, in order to scale, lower costs, and simplify management. One classical use case is found in particular in the context of the convergence of IT with Operational Technology (OT), also referred to as the Industrial Internet. But there are many others use cases [[I-D.ietf-detnet-use-cases](#)], for instance in in professional audio and video, automotive, radio fronthauls, etc..

The DetNet data plane alternatives [[I-D.dt-detnet-dp-alt](#)] studies the applicability of existing and emerging dataplane techniques that can be leveraged to enable DetNet properties in IP networks. One critical feature in the dataplane is traceability, the capability to control the activity of intermediate nodes on a packet. For instance, if Replication and Elimination is applied to a packet, then it is desirable to determine which node performed a certain copy of

that packet that is circulating in the network. Likewise, engineered paths are required to support redundant transmission across disjoint paths in support of DetNets PREF functions.

Traceability belongs to Operations, Administration, and Maintenance (OAM) which is the toolset for fault detection and isolation, and for performance measurement. More can be found on OAM Tools in "An Overview of Operations, Administration and Maintenance (OAM) Tools" [[I-D.ietf-opsawg-oam-overview](#)].

This document proposes a new set of mechanisms based on [[RFC8279](#)] (BIER) and more specifically BIER Traffic Engineering [[I-D.ietf-bier-te-arch](#)] (BIER-TE) to control the process of Packet Replication and Elimination Functions (PREF), and provide traceability of these operations, in the DetNet dataplane. An adjacency, which is represented by a bit in the BIER header, can correspond in the dataplane to an Ethernet hop, a Label Switched Path, or it can correspond to an IPv6 loose or strict source routed path.

BIER-TE was primarily designed to carry multicast traffic, but there is nothing prohibiting for it to be used with unicast traffic, and the authors of this document think that for networks whose size requirement match the supportable bitstring length (BSL) in BIER, it can be a good choice as the forwarding plane specifically for DetNet type traffic for both multicast and unicast traffic because it would be a common solution for unicast and multicast (limiting the number of different technologies a DetNet solution requires) and likely provides the most flexible support for path engineering, replication and elimination (PREF) and the novel OAM method described in this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. On BIER - Traffic Engineering

[[RFC8279](#)] (BIER) is a network plane replication technique that was initially intended as a new method for multicast distribution. In a nutshell, a BIER header includes a bitmap that explicitly signals the listeners that are intended for a particular packet, which means that 1) the sender is aware of the individual listeners and 2) the BIER control plane is a simple extension of the unicast routing as opposed to a dedicated multicast data plane, which represents a considerable

reduction in OPEX. For this reason, the technology faces a lot of traction from Service Providers.

The simplicity of the BIER technology makes it very versatile as a network plane signaling protocol. Already, a new Traffic Engineering variation is emerging that uses bits to signal segments along a TE path.

While BIER-TE was like BIER primarily developed for multicast traffic, the authors think that it can equally be attractive for unicast traffic requiring the DetNet resilience of multiple transitions. If the topology of the network can well be represented by standard BIER-TE bitstring sizes of e.g.: up to 256 bits, then this would allow for a single technology for both unicast and multicast.

BIER-TE supports a Traffic Engineered forwarding plane by explicit hop-by-hop forwarding and loose hop forwarding of packets.

From the BIER-TE architecture, the key differences over BIER are:

- o BIER-TE replaces in-network autonomous path calculation by explicit paths calculated off path for example by a BIER-TE controller host.
- o In BIER-TE every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies - instead of a BFER as in BIER. processing packets as a destination (BFER) is one of the possible adjacency types.
- o BIER-TE in each BFR has no routing table but only a BIER-TE Forwarding Table (BIFT) indexed by SI:BitPosition and populated with only those adjacencies to which the BFR should replicate packets to.

The generic view of an adjacency can be over a link, a tunnel or along a path segment.

4. BIER-TE-based Replication and Elimination Control

This document only needs to introduce new functionality to support the Elimination Function and OAM. Creation of appropriate BIER-TE packets is subject to to existing work.

In the solution described below, the encapsulation/insertion of flow-identification and sequence number into packets is performed by a function on the BFIR outside the scope of this document. A companion document [draft-huang-bier-te-encapsulation](#) defines an encapsulation for BIER-TE and BIER that can support flow-id and sequence-number ID. Other encapsulations can be used as well, as long as they provide

these signaling elements and are supported by the Elimination Function described in this document (e.g.: that the EF can read these fields and therefore remove duplicates). In the remainder of this document we will call this the extended BIER encapsulation and assume that it is used when describing examples. Unless otherwise noted, we assume that the BFIR performs encapsulation of some data flow packets with an extended BIER header, indicates BIER-TE forwarding in it and fills in flow-id and sequence number. It then fills in the bitstring with two (or more) alternative paths/DAGs and sends off the packets into the BIER-TE domain, replicating it itself if so indicated by the bitstring.

In a nutshell, BIER-TE is used as follows:

- o A controller computes a complex path, sometimes called a track, which takes the general form of a ladder. The steps and the side rails between them are the adjacencies that can be activated on demand on a per-packet basis using bits in the BIER header.

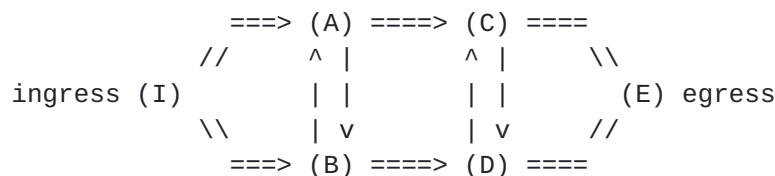


Figure 1: Ladder Shape with Replication and Elimination Points

- o The controller assigns a BIER domain, and inside that domain, assigns bits to the adjacencies. The controller assigns each bit to a replication node that sends towards the adjacency, for instance the ingress router into a segment that will insert a routing header in the packet. A single bit may be used for a step in the ladder, indicating the other end of the step in both directions.

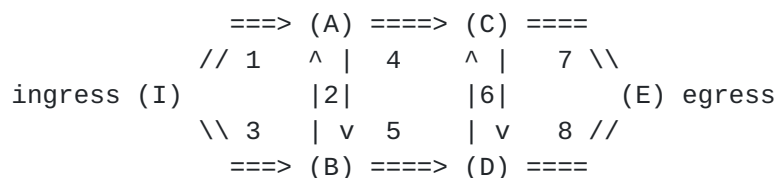


Figure 2: Assigning Bits

- o The controller activates the replication by deciding the setting of the bits associated with the adjacencies. This decision can be modified at any time, but takes the latency of a controller round trip to effectively take place. Below is an example that uses Replication and Elimination to protect the A->C adjacency. The "(EF)" in the following pictures Owner column indicates the fact that that BFR will perform the "Elimination Function" for received BIER-TE packets before further processing/copying them. In this example, only C performs EF. A (1) in the Example Bitstring indicates that the bit is set, but that the actual adjacency is not used by packets because this bit is shared with another adjacency and the overall bitstring will make the packet only use that other adjacency. This applies to bits 2 and 6.

Bit #	Adjacency	Owner	Example Bitstring
1	I->A	I	1
2	A->B	A	1
	B->A	B	(1)
3	I->B	I	0
4	A->C	A	1
5	B->D	B	1
6	C->D	C (EF)	(1)
	D->C	D	
7	C->E	C (EF)	1
8	D->E	D	0

Replication and Elimination Protecting A->C

Table 1: Controlling Replication

- o The BIER header with the controlling BitString , flow-id and sequence number is injected in the packet by the ingress node I (BFIR). That node may act as a replication point, in which case it may issue multiple copies of the packet, but for the purpose of this example it will not do it, so that the two paths used in this example only go from A to C, and therefore require explicit path engineering. For example, bandwidth I-A and I-B may be more limited and those paths being non long-haul may not warrant the dual transmission.

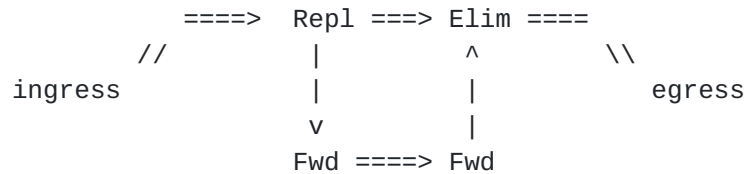


Figure 3: Enabled Adjacencies

- o For each of its bits that is set in the BIER header, the owner replication point resets the bit used for a copy and transmits towards the associated adjacency; to achieve this, the replication point copies the packet and inserts the relevant data plane information, such as next-hop label, MAC-address or source route header (for a BIER-TE routed adjacency), towards the adjacency that corresponds to the bit

Adjacency		BIER BitString	
I->A		01011110	
A->B		00011110	
B->D		00010110	
D->C		00010010	
A->C		01001110	

BitString in BIER Header as Packet Progresses

Table 2: BIER-TE in Action

- o Adversely, an elimination node on the path performs the Elimination Function which will remove duplicate packets (same flow-id, same sequence number) and performs a bitwise AND on the BitStrings from the various copies of the packet that it has received, before it forwards the packet with the resulting BitString. Details of the Elimination Function are described below.

+-----+-----+		
Operation BIER BitString		
+-----+-----+		
D->C	00010010	
A->C	01001110	

AND in C	00000010	
C->E	00000000	
+-----+-----+		

BitString Processing at Elimination Point C

Table 3: BIER-TE in Action (cont.)

- o In this example, all the transmissions succeeded and the BitString at arrival has all the bits reset - note that the egress may be an Elimination Point in which case this is evaluated after this node has performed its AND operation on the received BitStrings).

+-----+-----+		
Failing Adjacency Egress BIER BitString		
+-----+-----+		
I->A	Frame Lost	
I->B	Not Tried	
A->C	00010000	
A->B	01001100	
B->D	01001100	
D->C	01001100	
C->E	Frame Lost	
D->E	Not Tried	
+-----+-----+		

BitString indicating failures

Table 4: BIER-TE in Action (cont.)

- o But if a transmission failed along the way, one (or more) bit is never cleared. Table 4 provides the possible outcomes of a transmission. If the frame is lost, then it is probably due to a failure in either I->A or C->E, and the controller should enable I->B and D->E to find out. A BitString of 00010000 indicates unequivocally a transmission error on the A->C adjacency, and a BitString of 01001100 indicates a loss in either A->B, B->D or D->C; enabling D->E on the next packets may provide more information to sort things out.

In more details:

The BIER header is of variable size, and a DetNet network of a limited size can use a model with 64 bits if 64 adjacencies are enough, whereas a larger deployment may be able to signal up to 256 adjacencies for use in very complex paths. The format of this header is common to BIER and BIER-TE.

For the DetNet data plane, a replication point is an ingress point for more than one adjacency, and an elimination point is an egress point for more than one adjacency.

A pre-populated state in a replication node indicates which bits are served by this node and to which adjacency each of these bits corresponds. With DetNet, the state is typically installed by a controller entity such as a PCE. The way the adjacency is signaled in the packet is fully abstracted in the bit representation and must be provisioned to the replication nodes and maintained as a local state, together with the timing or shaping information for the associated flow.

The DetNet data plane uses BIER-TE to control which adjacencies are used for a given packet. This is signaled from the path ingress, which sets the appropriate bits in the BIER BitString to indicate which replication must happen.

The replication point clears the bit associated to the adjacency where the replica is placed, and the elimination points perform a logical AND of the BitStrings of the copies that it gets before forwarding.

As is apparent in the examples above, clearing the bits enables to trace a packet to the replication points that made any particular copy. BIER-TE also enables to detect the failing adjacencies or sequences of adjacencies along a path and to activate additional replications to counter balance the failures.

Finally, using the same BIER-TE bit for both directions of the steps of the ladder enables to avoid replication in both directions along the crossing adjacencies. At the time of sending along the step of the ladder, the bit may have been already reset by performing the AND operation with the copy from the other side, in which case the transmission is not needed and does not occur (since the control bit is now off).

5. Elimination Function (Normative)

This section defines the normative behavior of the Elimination Function with optional OAM sub-function.

The Elimination Function is performed logically on reception of BIER-TE packets. It is therefore not part of the adjacencies or otherwise assigned to a specific bit. "Logically" means that this specification does not constrain implementations, especially on multi-linecard/multi-chassis systems to perform EF on a physical egress module. It just implies that it has to happen before replication to the bits in the bitstring.

TBD: In addition to being an ingress, EF could as well be modelled as a new adjacency assigned to bits. The full adjacency of a bit could then be a sequence of EF followed by one (or more) of existing adjacencies. This is currently not considered by this document due to the lack of identified need to support this option - e.g.: problems that can not be equally/better be solved with EF logically on ingress.

The Elimination Function is more formally written as EF(OAM, BIFT, {flows}/*), and is configured like BIFTs from the BIER-TE controller host and/or other future mechanisms.

OAM is boolean and indicates whether OAM function of bitwise AND of received packet copies is performed. This OAM function requires additional memory/processing over EF without OAM. Note that the OAM function does not change the effect of the Elimination Function for BFR/receivers - they will continue to just receive the first copy of a packet. Instead, it will continue to track further copies solely for the purpose of providing OAM information. This also requires some timeout or sequence number advancement to decide when to terminate waiting for further copies of packets before considering the OAM analysis of this packet to be complete. BFR supporting this document SHOULD support the OAM sub-function.

BIFT indicates the <SD,SI,BSL> for which to perform EF. Devices SHOULD support enabling per EF. {flows}/* indicates the set of flows for which EF operates (using the specified BIFT). Duplicate elimination has to create per-flow state to remember which sequence number packets for this flow were already received. In the case of OAM also what bits were set in that received prior copy of the packet.

When a device supports "*", then it will automatically allocate such a flow-state for every new recognized flow and expire such flow state after an operator determined timeout of activity - for example with a default of 10 seconds. Dynamic allocation of flow-state may cause some initial duplicates before this state is working and it makes the BFR more vulnerable to state DOS attacks, but it will allow BIER applications to send flows with the benefit of EF without the help of the controller having to know and program every flow.

In the {flows} option, control procedures (e.g.: BIER-TE controller host) indicate to the BFR explicitly the set of flows for which it should install/operate the EF function. Note that the flow-id in the extended BIER encapsulation is the combination of BFIR-ID and entropy field of the BIER header.

BFR supporting this document MUST support the {flows} option and MAY support the "*" option.

The following picture explains the results of EF being performed on ingres in a typical example:

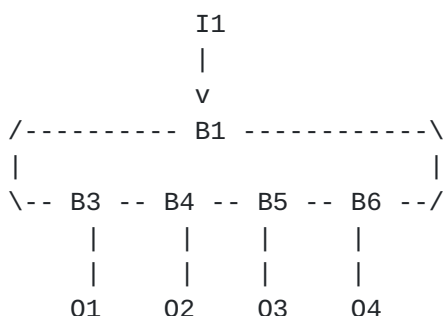


Figure 4: EF with Rings

Consider a simple ring where BFIR I1 generates BIER-TE packets. The bitstring indicates that the packet is sent hop-by-hop counterclockwise B1->B3->B4->B6 and counterclockwise B1->B6->B5->B4->B3. Bits for BFER 01, 02, 03 and 04 are also set. B3,B4,B5,B6,B7 perform EF. The result of this setup is that B2 creates two copies of the packets received from I1, one going to B6, the other to B3. Assume B4 first received the counter-clockwise copy from B3 and B5 the clockwise copy from B6. They will both forward these packets to each other because those were the first copies they saw, but they would block these second copies. Therefore only the link B4<->B5 will have carried the packet copy twice (once in each direction). All the other ring links will only carry one copy of the packet.

This is notably different from schemes where EF is not performed before replication, but afterwards. In those schemes, both copies of the packets would flow counterclockwise around (most of) the ring, occupying more bandwidth.

6. Summary

With the addition of the functions of this document, BIER-TE becomes a potential option for the DetNet dataplane specifically beneficial when PREF (replication and elimination) is required for resilience

(to reduce packet loss). For DetNet multicast but also DetNet unicast. The unique capabilities of this approach are:

- o Explicit per-packet path selection for packet. Multicast and Unicast.
- o Control which replication take place on a per packet basis, so that replication points can be configured but not actually utilized
- o Trace the replication activity and determine which node replicated a particular packet
- o Measure the quality of transmission of the actual data packet along the replication segments and use that in a control loop to adapt the setting of the bits and maintain the reliability.

7. Implementation Status

A research-stage implementation of the forwarding plane for a 6TiSCH IOT use case was developed at Cisco's Paris Innovation Lab (PIRL) by Zacharie Brodard. It was implemented on OpenWSN Open-source firmware and tested on the OpenMote-CC2538 hardware. It implements the header types 15, 16, 17, 18 and 19 (bit-by-bit encoding without group ID) in order to allow a BIER-TE protocol over IEEE802.15.4e.

This work was complemented with a Controller-based control loop by Hao Jiang. The controller builds the complex paths (called Tracks in 6TiSCH) and decides the setting of the BitStrings in real time in order to optimize the delivery ratio within a minimal energy budget.

Links:

github: <https://github.com/zach-b/openwsn-fw/tree/BIER>
OpenWSN firmware: <https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187>
OpenMote hardware: <http://www.openmote.com/>

8. Security considerations

TBD.

9. IANA Considerations

This document has no IANA considerations.

10. Acknowledgements

The method presented in this document was discussed and worked out together with the DetNet Data Plane Design Team:

Jouni Korhonen
Janos Farkas
Norman Finn
Olivier Marce
Gregory Mirsky
Pascal Thubert
Zhuangyan Zhuang

The authors also like to thank the DetNet chairs Lou Berger and Pat Thaler, as well as Thomas Watteyne, 6TiSCH co-chair, for their contributions and support to this work.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

11.2. Informative References

[I-D.dt-detnet-dp-alt]
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", [draft-dt-detnet-dp-alt-04](#) (work in progress), September 2016.

[I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication (BIER-TE)", [draft-ietf-bier-te-arch-00](#) (work in progress), January 2018.

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", [draft-ietf-detnet-architecture-04](#) (work in progress), October 2017.

[I-D.ietf-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", [draft-ietf-detnet-problem-statement-02](#) (work in progress), September 2017.

[I-D.ietf-detnet-use-cases]

Grossman, E., "Deterministic Networking Use Cases", [draft-ietf-detnet-use-cases-14](#) (work in progress), February 2018.

[I-D.ietf-opsawg-oam-overview]

Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", [draft-ietf-opsawg-oam-overview-16](#) (work in progress), March 2014.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [draft-ietf-spring-segment-routing-15](#) (work in progress), January 2018.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", [RFC 8279](#), DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

Authors' Addresses

Pascal Thubert (editor)
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Toerless Eckert
Huawei USA - Futurewei Technologies Inc.
2330 Central Expy
Santa Clara 95050
USA

Email: tte+ietf@cs.fau.de

Zacharie Brodard
Ecole Polytechnique
Route de Saclay
Palaiseau 91128
FRANCE

Phone: +33 6 73 73 35 09
Email: zacharie.brodard@polytechnique.edu

Hao Jiang
Telecom Bretagne
2, rue de la Chataigneraie
Cesson-Sevigne 35510
FRANCE

Phone: +33 7 53 70 97 34
Email: hao.jiang@telecom-bretagne.eu

