DetNet P. Thubert, Ed.
Internet-Draft Cisco
Intended status: Standards Track Z. Brodard
Expires: March 18, 2017 Ecole Polytechnique
H. Jiang
Telecom Bretagne
September 14, 2016

BIER-TE-based OAM, Replication and Elimination
draft-thubert-bier-replication-elimination-00

Abstract

This specification leverages Bit Index Explicit Replication - Traffic Engineering to control in the data plane the DetNet Replication and Elimination activities, and to provide traceability on links where replication and loss happen, in a manner that is abstract to the forwarding information.

Status of This Memo

Copyright Notice

Table of Contents

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.finn-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.

Traceability belongs to Operations, Administration, and Maintenance (OAM), which is the toolset for fault detection and isolation, and

for performance measurement. An Overview of OAM Tools is available at [I-D.ietf-detnet-use-cases].

This document proposes a new set to OAM tools based on Bit Indexed Explicit Replication [I-D.ietf-bier-architecture] (BIER) and more specifically BIER Traffic Engineering [I-D.eckert-bier-te-arch] (BIER-TE) to control the process or Replication and Elimination, 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.

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

BIER [I-D.ietf-bier-architecture] 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 is mainly a multicast technology that typically leverages a unicast distributed control plane through IGP extensions, BIER-TE [I-D.eckert-bier-te-arch] is mainly a unicast technology that leverages a central computation to setup path, compute segments and install the mapping in the intermediate nodes.

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 by the 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.

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.

With Segment Routing [I-D.ietf-spring-segment-routing] a segment can be signaled as an MPLS label, or an IPv6 routing header . A segment may be loosely of strictly source routed, depending on the need for full non-congruence and the confidence that loose routing may still achieve that need.

4. BIER-TE-based Replication and Elimination Control

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.

```
                   ===> (A) ====> (C) ====
                 //     ^ |       ^ |     \\
        ingress (I)     | |       | |       (E) egress
```

```
                 \\     | v       | v     //
                  ===> (B) ====> (D) ====
```

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.


```
                  ===> (A) ====> (C) ====
                 // 1   ^ |  4    ^ |   7 \\
       ingress (I)        |2|       |6|       (E) egress
                 \\ 3   | v  5    | v   8 //
                  ===> (B) ====> (D) ====
```

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.

Bit #	Adjacency	Owner	Example Bit Setting
1	I->A	I	1
2	A->B	A	1
	B->A	B	
3	I->C	I	0

4	A->C	A	1
5	B->D	B	1
6	C->D	C	1
	D->C	D	
7	C->E	C	1
8	D->E	D	0

Replication and Elimination Protecting A->C

Table 1: Controlling Replication

- o The BIER header with the controlling BitString is injected in the packet by the ingress node of the deterministic path. That node may act as a replication point, in which case it may issue multiple copies of the packet

```
                ====>  Repl ===> Elim ====
              //         |         ^        \\
      ingress            |         |          egress
                         v         |
                        Fwd ====> Fwd
```

Figure 3: Enabled Adjacencies

- o For each of its bits that is set in the BIER header, the owner replication point resets the bit and transmits towards the associated adjacency; to achieve this, the replication point copies the packet and inserts the relevant data plane information, such as a source route header, 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

- Adversely, an elimination node on the way strips the data plane information 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.

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.)

- 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.)

- 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. Summary

BIER-TE occupies a particular position in the DetNet dataplane. In the one hand it is optional, and only useful if replication and elimination is taking place. In the other hand, it has unique capabilities to:

- 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.

6. Implementation Status

A research-stage implementation of the forwarding plane fir 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 IEE802.15.4e.

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 oif 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/

7. Security considerations

TBD.

8. IANA Considerations

This document has no IANA considerations.

9. 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.

10. References

10.1. Normative References

[I-D.finn-detnet-architecture]
Finn, N. and P. Thubert, "Deterministic Networking Architecture", draft-finn-detnet-architecture-08 (work in progress), August 2016.

[I-D.ietf-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-04 (work in progress), July 2016.

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

10.2. 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-03 (work in progress), August 2016.

[I-D.eckert-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", draft-eckert-bier-te-arch-04 (work in progress), July 2016.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-10 (work in progress), June 2016.

[I-D.ietf-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-ietf-detnet-problem-statement-00 (work in progress), April 2016.

[I-D.ietf-detnet-use-cases]
Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y., Varga, B., Farkas, J., Goetz, F., and J. Schmitt,

"Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-10 (work in progress), July 2016.

[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., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-09 (work in progress), July 2016.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

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

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