DetNet
Internet-Draft
Intended status: Informational
Expires: February 18, 2017

J. Korhonen, Ed.
Broadcom
J. Farkas
G. Mirsky
Ericsson
P. Thubert
Cisco
Y. Zhuang
Huawei
L. Berger
LabN

August 17, 2016

DetNet Data Plane Protocol and Solution Alternatives
draft-dt-detnet-dp-alt-03

Abstract

This document identifies existing IP and MPLS, and other encapsulations that run over IP and/or MPLS data plane technologies that can be considered as the base line solution for deterministic networking data plane definition.

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, timely delivery and bounded packet delay variation [I-D.finn-detnet-architecture]. The deterministic networking Quality of Service (QoS) is expressed as 1) the minimum and the maximum end-to-end latency from source (talker) to destination (listener), and 2) probability of loss of a packet. Only the worst-case values for the mentioned parameters are concerned.

There are three techniques to achieve the QoS required by deterministic networks:

- Congestion protection,
- explicit routes,
- service protection.

This document identifies existing IP and Multiprotocol Label Switching (MPLS) [RFC3031], layer-2 or layer-3 encapsulations and transport protocols that could be considered as foundations for a deterministic networking data plane. The full scope of the deterministic networking data plane solution is considered including, as appropriate: quality of service (QoS); Operations, Administration and Maintenance (OAM); and time synchronization among other criteria described in Section 4.

This document does not select a deterministic networking data plane protocol. It does, however, elaborate what it would require to adapt and use a specific protocol as the deterministic networking data plane solution. This document is only concerned with data plane considerations and, specifically, with topics that potentially impact potential deterministic networking aware data plane hardware. Control plane considerations are out of scope of this document.

2. Terminology

This document uses the terminology established in the DetNet architecture [I-D.finn-detnet-architecture].

3. DetNet Data Plane Overview

A "Deterministic Network" will be composed of DetNet enabled nodes i.e., End Systems, Edge Nodes, Relay Nodes and collectively deliver DetNet services. DetNet enabled nodes are interconnected via Transit Nodes (i.e., routers) which support DetNet, but are not DetNet service aware. Transit nodes see DetNet nodes as end points. All

DetNet enabled nodes are connect to sub-networks, where a point-to-point link is also considered as a simple sub-network. These sub-networks will provide DetNet compatible service for support of DetNet traffic. Examples of sub-networks include IEEE 802.1TSN and OTN. Of course, multi-layer DetNet systems may also be possible, where one DetNet appears as a sub-network, and provides service to, a higher layer DetNet system. A simple DetNet concept network is shown in Figure 1.

```
  TSN              Edge          Transit        Relay        DetNet
  End System       Node            Node         Node         End System

  +---------+    +.........+                                 +---------+
  |  Appl.  |<---:Svc Proxy:-- End to End Service ---------->|  Appl.  |
  +---------+    +---------+                   +---------+   +---------+
  |   TSN   |    |TSN| |Svc|<-- DetNet flow ---: Service :-->| Service |
  +---------+    +---+ +---+    +---------+    +---------+   +---------+
  |Transport|    |Trp| |Trp|    |Transport|    |Trp| |Trp|   |Transport|
  +-------.-+    +-.-+ +-.-+    +--.----.-+    +-.-+ +-.-+   +---.-----+
          :  Link  :    /  ,-----.  \   :  Link  :    /  ,-----.  \
          +........+    +-[  Sub  ]-+   +........+    +-[  Sub  ]-+
                          [Network]                     [Network]
                           `-----'                       `-----'
```

Figure 1: A Simple DetNet Enabled Network

The DetNet data plane is logically divided into two layers (also see Figure 2):

DetNet Service Layer

The DetNet service layer provides adaptation of DetNet services. It is composed of a shim layer to carry deterministic flow specific attributes, which are needed during forwarding and for service protection. DetNet enabled end systems originate and terminate the DetNet Service layer and are peers at the DetNet Service layer. DetNet relay and edge nodes also implement DetNet Service layer functions. The DetNet service layer is used to deliver traffic end to end across a DetNet domain.

DetNet Transport Layer

The DetNet transport layer is required on all DetNet nodes. All DetNet nodes are end points and the transport layer. Non-DetNet service aware transit nodes deliver traffic between DetNet nodes. The DetNet transport layer operates below and supports the DetNet

Service layer and optionally provides congestion protection for DetNet flows.

Distinguishing the function of these two DetNet data plane layers helps to explore and evaluate various combinations of the data plane solutions available. This separation of DetNet layers, while helpful, should not be considered as formal requirement. For example, some technologies may violate these strict layers and still be able to deliver a DetNet service.

```
           .
           .
   +-----------+
   |  Service  | PW, RTP/(UDP), GRE
   +-----------+
   | Transport | (UDP)/IPv6, (UDP)/IPv4, MPLS LSPs, BIER, BIER-TE
   +-----------+
           .
           .
```

Figure 2: DetNet adaptation to data plane

The two logical layers defined here aim to help to identify which data plane technology can be used for what purposes in the DetNet context. This layering is similar to the data plane concept of MPLS, where some part of the label stack is "Service" specific (e.g., PW labels, VPN labels) and an other part is "Transport" specific (e.g, LSP label, TE label(s)).

In some networking scenarios, the end system initially provides a DetNet flow encapsulation, which contains all information needed by DetNet nodes (e.g., Real-time Transport Protocol (RTP) [RFC3550] based DetNet flow transported over a native UDP/IP network or PseudoWire). In other scenarios, the encapsulation formats might differ significantly. As an example, a CPRI "application's" I/Q data mapped directly to Ethernet frames may have to be transported over an MPLS-based packet switched network (PSN).

There are many valid options to create a data plane solution for DetNet traffic by selecting a technology approach for the DetNet Service layer and also selecting a technology approach for the DetNet Transport layer. There are a high number of valid combinations. Therefore, not the combinations but the different technologies are evaluated along the criteria collected in Section 4. Different criteria apply for the DetNet Service layer and the DetNet Transport layer, however, some of the criteria are valid for both layers.

One of the most fundamental differences between different potential data plane options is the basic addressing and headers used by DetNet end systems. For example, is the basic service a Layer 2 (e.g., Ethernet) or Layer 3 (i.e., IP) service. This decision impacts how DetNet end systems are addressed, and the basic forwarding logic for the DetNet Service layer.

3.1. Example DetNet Service Scenarios

In an attempt to illustrate a DetNet date plane, this document uses the Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254] reference model shown in Figure 3 as the foundation for different DetNet data plane deployment options and how layering could work. Other reference models are possible but not covered in this document. Note that other technologies can be also used to implement DetNet, Multi-Segment PW is only used here to illustrate functions, features and layering from the perspective of the architecture.

```
         Native  |<--------Multi-Segment Pseudowire----->|  Native
         Service |         PSN              PSN          |  Service
          (AC)   |     |<-Tunnel->|     |<-Tunnel->|     |  (AC)
           |     V     V     1    V     V     2    V     V   |
           |     +-----+          +-----+          +---- +   |
   +---+   |     |T-PE1|==========|S-PE1|==========|T-PE2|   |    +---+
   |   |---|-----|........PW1...........|...PW3..........|---|----|   |
   |CE1|   |     |     |          |     |          |     |   |    |CE2|
   |   |---------|........PW2...........|...PW4..........|--------|   |
   +---+   |     |     |==========|     |==========|     |   |    +---+
       ^         +-----+          +-----+          +-----+        ^
       |     Provider Edge 1         ^        Provider Edge 3     |
       |                             |                            |
       |                     PW switching point                   |
       |                                                          |
       |<------------------- Emulated Service ------------------->|
```

Figure 3: Pseudo Wire switching reference model

Figure 4 illustrates how DetNet can provide services for IEEE 802.1TSN end systems over a DetNet enabled network. The edge nodes insert and remove required DetNet data plane encapsulation. The 'X' in the edge and relay nodes represents a potential DetNet flow packet replication and elimination point. This conceptually parallels L2VPN services, and could leverage existing related solutions as discussed below.

```
          TSN    |<----- End to End DetNet Service ----->|  TSN
         Service |        Transit          Transit       | Service
   TSN    (AC)   |     |<-Tunnel->|     |<-Tunnel->|     |  (AC)     TSN
   End     |     V     V     1    V     V     2    V     V   |       End
   System  |     +-----+          +-----+          +---- +   |    System
   +---+   |     |T-PE1|==========|S-PE1|==========|T-PE2|   |    +---+
   |   |---|-----|.X_..DetNet Flow1..X..|...DF3........X.|---|----|   |
   |CE1|   |     |  \  |          |     |          |  /  |   |    |CE2|
   |   |         |...X_...DF2........X..|...DF4......X_..|        |   |
   +---+         |     |==========|     |==========|     |        +---+
       ^         +-----+          +-----+          +-----+        ^
       |        Edge Node        Relay Node       Edge Node       |
       |                                                          |
       |<--------------- Emulated TSN Service ------------------->|
```

Figure 4: IEEE 802.1TSN over DetNet

Figure 5 illustrates how end to end native DetNet service can be provided. In this case, the end systems are able to send and receive native DetNet flows. For example, as PseudoWire (PW) encapsulated IP. Like earlier the 'X' in the end systems, edge and relay nodes represents potential DetNet flow packet replication and elimination points. Here the relay nodes may change the underlying transport, for example replacing IP with MPLS or tunneling IP over MPLS (e.g., via L3VPNs), or simply interconnect network domains.

```
          DetNet                                                DetNet
          Service         Transit          Transit        Service
   DetNet   |          |<-Tunnel->|     |<-Tunnel->|         |    DetNet
   End      |          V     1    V     V     2    V         |    End
   System   |    +-----+          +-----+          +-----+   |    System
   +---+    |    |S-PE1|==========|S-PE2|==========|S-PE3|   |    +---+
   |  X....DFa.....X_.......DF1.......X_....DF3........X.....DFa...X  |
   |CE1|=========|  \  |          |  /  |          |  /  |========|CE2|
   |   |    |    |   \......DF2.....X_......DF4....../   |   |    |   |
   +---+         |     |==========|     |==========|     |        +---+
       ^         +-----+          +-----+          +-----+        ^
       |        Relay Node       Relay Node       Relay Node      |
       |                                                          |
       |<------------- End to End DetNet Service ---------------->|
```

Figure 5: Native DetNet

Figure 6 illustrates how a IEEE 802.1TSN end system could communicate with a native DetNet end system through an edge node which provides a TSN to DetNet inter-working capability. The edge node would add and remove required DetNet data plane encapsulation as well as provide any needed address mapping. As in previous figures, the 'X' in the

end systems, edge and relay nodes represents potential DetNet flow packet duplication and elimination points.

```
        TSN    |<----- End to End DetNet Service -------------->|
       Service |        Transit          Transit                |
TSN     (AC)   |     |<-Tunnel->|     |<-Tunnel->|       DetNet | DetNet
End      |     V     V     1    V     V     2    V      Service | End
System   |     +-----+          +-----+          +-----+   |    V System
 +---+   |     |T-PE1|==========|S-PE1|==========|S-PE2|   |    +---+
 |   |---|-----|.X_.......DF1......X..|...DF3........X.|...DFa...X  |
 |CE1|   |     |  \  |          |     |          |  /  |========|CE2|
 |   |         |   \.....DF2.......X..|...DF4....../   |   |    |   |
 +---+         |     |==========|     |==========|     |        +---+
     ^         +-----+          +-----+          +-----+        ^
     |        Edge Node        Relay Node      Relay Node       |
     |                                                          |
     |<----------------- End to End Service ------------------->|
```

Figure 6: IEEE 802.1TSN to native DetNet

4. Criteria for data plane solution alternatives

This section provides criteria to help to evaluate potential options. Each deterministic networking data plane solution alternative is described and evaluated using the criteria described in this section. The used criteria enumerated in this section are selected so that they highlight the existence or lack of features that are expected or seen important to a solution alternative for the data plane solution.

The criteria for the DetNet Service layer:

#1 Encapsulation and overhead
#2 Flow identification (Service ID part of the DetNet flows)
#3 Packet sequencing and duplicate elimination
#5 Flow duplication and merging
#6 Operations, Administration and Maintenance (capabilities)
#8 Class and quality of service capabilities (DetNet Service specific)
#10 Technical maturity

The criteria for the DetNet Transport layer:

#1 Encapsulation and overhead
#2 Flow identification
#4 Explicit routes (network path)
#5 Flow duplication and merging (sometimes, flow duplication and merging is also doable at the transport layer, not just at the service layer)

#6 Operations, Administration and Maintenance (capabilities, performance management, packet traceability)
#8 Class and quality of service capabilities (DetNet Transport specific)
#9 Packet traceability (can be part of OAM)
#10 Technical maturity

[Editor's Note: numbering is off because #7 is removed.]

[Editor's Note: #9 should(?) be integrated into #6.]

Most of the criteria is relevant for both the DetNet Service and DetNet Transport layers. However, different aspects of the same criteria may relevant for different layers, for example, as it is the case with criteria #5 Packet replication and elimination.

4.1. #1 Encapsulation and overhead

Encapsulation and overhead is related to how the DetNet data plane carries DetNet flow. In several cases a DetNet flow has to be encapsulated inside other protocols, for example, when transporting a layer-2 Ethernet frame over an IP transport network. In some cases a tunneling like encapsulation can be avoided by underlying transport protocol translation, for example, translating layer-2 Ethernet frame including addressing and flow identification into native IP traffic. Last it is possible that sources and destinations handle deterministic flows natively in layer-3. This criteria concerns what is the encapsulation method the solution alternative support: tunneling like encapsulation, protocol translation or native layer-3 transport. In addition to the encapsulation mechanism this criteria is also concerned of the processing and specifically the encapsulate header overhead.

4.2. #2 Flow identification

The solution alternative has to provide means to identify specific deterministic flows. The flow identification can, for example, be explicit field in the data plane encapsulation header or implicitly encoded into the addressing scheme of the used data plane protocol or their combination. This criteria concerns the availability and details of deterministic flow identification the data plane protocol alternative has.

4.3. #3 Packet sequencing and duplicate elimination

The solution alternative has to provide means for end systems to number packets sequentially and transport that sequencing information along with the sent packets. In addition to possible reordering

packets other important uses for sequencing are detecting duplicates and lost packets.

In a case of intentional packet duplication a combination of flow identification and packet sequencing allows for detecting and eliminating duplicates at the destination (see Section 4.5 for more details).

4.4. #4 Explicit routes

The solution alternative has to provide a mechanism(s) for establishing explicit routes that all packets belonging to a deterministic flow will follow. The explicit route can be seen as a form of source routing or a pre-reserved path e.g., using some network management procedure. It should be noted that the explicit route does not need to be detailed to a level where every possible intermediate node along the path is part of the named explicit route. RSVP-TE [RFC3209] supports explicit routes, and typically provides pinned data paths for established LSPs. At Layer-2, the IEEE 802.1Qca [IEEE802.1Qca] specification defines how to do explicit path control in a bridged network and its IETF counter part is defined in [RFC7813]. This criteria concerns the available mechanisms for explicit routes for the data plane protocol alternative.

4.5. #5 Flow duplication and merging

Flow duplication and flow merging are methods being considered to provide DetNet service protection. The objective for supporting flow duplication and flow merging is to enable hitless (or lossless) 1+1 protection. Other methods, if so identified, are also permissible.

The solution alternative has to provide means for end systems, relay and edge nodes to be able to duplicate packets into duplicate flows, and later merge the flows into one for duplicate elimination. The duplication and merging may take place at multiple points in the network in order to ensure that one (or more) equipment failure event(s) still leave at least one path intact for a deterministic networking flow. The goal is again to enable hitless 1+1 protection in a way that no packet gets lost or there is no ramp up time when either one of the paths fails for one reason or another.

Another concern regarding packet duplication is how to enforce duplicated packets to take different route or path while the final destination still remains the same. With strict source routing, all the intermediate hops are listed and paths can be guaranteed to be non-overlapping. Loose source routing only signals some of the intermediate hops and it takes additional knowledge to ensure that there is no single point of failure.

The IEEE 802.1CB (seamless redundancy) [IEEE8021CB] is an example of Ethernet-based solution that defines packet sequence numbering, flow duplication, flow merging, duplicate packet identification and elimination. The deterministic networking data plane solution alternative at layer-3 has to provide equivalent functionality. This criteria concerns the available mechanisms for packet replication and duplicate deletion the data plane protocol alternative has.

4.6. #6 Operations, Administration and Maintenance

The solution alternative should demonstrate an availability of appropriate standardized OAM tools that can be extended for deterministic networking purposes with a reasonable effort, when required. The OAM tools do not necessarily need to be specific to the data plane protocol as it could be the case, for example, with MPLS-based data planes. But any OAM-related implications or requirements on data plane hardware must be considered.

The OAM includes but is not limited to tools listed in the requirements for overlay networks [I-D.ooamdt-rtgwg-ooam-requirement]. Specifically, the performance management requirements are of interest at both service and transport layers.

4.7. #8 Class and quality of service capabilities

Class and quality of service, i.e., CoS and QoS, are terms that are often used interchangeably and confused. In the context of DetNet, CoS is used to refer to mechanisms that provide traffic forwarding treatment based on aggregate group basis and QoS is used to refer to mechanisms that provide traffic forwarding treatment based on a specific DetNet flow basis. Examples of CoS mechanisms include DiffServ which is enabled by IP header differentiated services code point (DSCP) field [RFC2474] and MPLS label traffic class field [RFC5462], and at Layer-2, by IEEE 802.1p priority code point (PCP).

Quality of Service (QoS) mechanisms for flow specific traffic treatment typically includes a guarantee/agreement for the service, and allocation of resources to support the service. Example QoS mechanisms include discrete resource allocation, admission control, flow identification and isolation, and sometimes path control, traffic protection, shaping, policing and remarking. Example protocols that support QoS control include Resource ReSerVation Protocol (RSVP) [RFC2205] (RSVP) and RSVP-TE [RFC3209] and [RFC3473].

A critical DetNet service enabled by QoS (and perhaps CoS) is delivering zero congestion loss. There are different mechanisms that maybe used separately or in combination to deliver a zero congestion

loss service. The key aspect of this objective is that DetNet packets are not discarded due to congestion at any point in a DetNet aware network.

In the context of the data plane solution there should be means for flow identification, which then can be used to map a flow against specific resources and treatment in a node enforcing the QoS. Hereto, certain aspects of CoS and QoS may be provided by the underlying sub-net technology, e.g., actual queuing or IEEE 802.3x priority flow control (PFC).

4.8. #9 Packet traceability

For the network management and specifically for tracing implementation or network configuration errors any means to find out whether a packet is a replica, which node performed replication, and which path was intended for the replica, can be very useful. This criteria concerns the availability of solutions for tracing packets in the context of data plane protocol alternative. Packet traceability can also be part of OAM.

4.9. #10 Technical maturity

The technical maturity of the data plane solution alternative is crucial, since it basically defines the effort, time line and risks involved for the use of the solution in deployments. For example, the maturity level can be categorized as available immediately, available with small extensions, available with re-purposing/redefining portions of the protocol or its header fields. Yet another important measure for maturity is the deployment experience. This criteria concerns the maturity of the data plane protocol alternative as the solution alternative. This criteria is particularly important given, as previously noted, that the DetNet data plane solution is expected to impact, i.e., be supported in, hardware.

5. Data plane solution alternatives

The following sections describe and rate deterministic data plane solution alternatives. In "Analysis and Discussion" section each alternative is evaluated against the criteria given in Section 4 and rated using the following: (M)eets the criteria, (W)ork needed, and (N)ot suitable or too much work envisioned.

5.1. DetNet Transport layer technologies

5.1.1. Native IPv6 transport

5.1.1.1. Solution description

This section investigates the application of native IPv6 [RFC2460] as the data plane for deterministic networking along the criteria collected in Section 4.

The application of higher OSI layer headers, i.e., headers deeper in the packet, can be considered. Two aspects have to be taken into account for such solutions. (i) Those header fields can be encrypted. (ii) Those header fields are deeper in the packet, therefore, routers have to apply deep packet inspection. See further details in Section 5.2.5.

5.1.1.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

IPv6 can encapsulate DetNet Service layer headers (and associated DetNet flow payload) like any other upper-layer header indicated by the Next Header. The fixed header of an IPv6 packet is 40 bytes [RFC2460]. This overhead is bigger if any Extension Header is used, and a generic behaviour for host and forwarding nodes is specified in [RFC7045]. However, the exact overhead (Section 4.1) depends on what solution is actually used to provide DetNet features, e.g., explicit routing or DetNet service protection if any of these is applied.

IPv6 has two types of Extension Headers that are processed by intermediate routers between the source and the final destination and may be of interest for the data plane signaling, the Routing Header that is used to direct the traffic via intermediate routers in a strict or loose source routing way, and the Hop-by-Hop Options Header that carries optional information that must be examined by every node along a packet's delivery path. The Hop-by-Hop Options Header, when present, must immediately follow the IPv6 Header and it is not possible to limit its processing to the end points of Source Routed segments.

IPv6 also provides a Destination Options Header that is used to carry optional information to be examined only by a packet's destination node(s). The encoding of the options used in the Hop-by-Hop and in the Destination Options Header indicates the expected behavior when a processing IPv6 node does not recognize the Option Type, e.g. skip or drop; it should be noted that due to

performance restrictions nodes may ignore the Hop-by-Hop Option Header, drop packets containing a Hop-by-Hop Option Header, or assign packets containing a Hop-by-Hop Option Header to a slow processing path [I-D.ietf-6man-rfc2460bis] (e.g. punt packets from hardware to software forwarding which is highly detrimental to the performance).

The creation of new Extension Headers that would need to be processed by intermediate nodes is strongly discouraged. In particular, new Extension Header(s) having hop-by-hop behavior must not be created or specified. New options for the existing Hop-by-Hop Header should not be created or specified unless no alternative solution is feasible [RFC6564].

#2 Flow identification (W)

The 20-bit flow label field of the fixed IPv6 header is suitable to distinguish different deterministic flows. But guidance on the use of the flow label provided by [RFC6437] places restrictions on how the flow label can be used. In particular, labels should be chosen from an approximation to a discrete uniform distribution. Additionally, existing implementations generally do not open APIs to control the flow label from the upper layers.

Alternatively, the Flow identification could be transported in a new option in the Hop-by-Hop Options Header.

#4 Explicit routes (W)

One possibility is for a Software-Defined Networking (SDN) [RFC7426] based approach to be applied to compute, establish and manage the explicit routes, leveraging Traffic Engineering (TE) extensions to routing protocols [RFC5305] [RFC7752] and evolving to the Path Computation Element (PCE) Architecture [RFC5440], though a number of issues remain to be solved [RFC7399].

Segment Routing (SR) [I-D.ietf-spring-segment-routing] is a new initiative to equip IPv6 with explicit routing capabilities. The idea for the DetNet data plane would be to apply SR to IPv6 with the addition of a new type of routing extension header [I-D.ietf-6man-segment-routing-header] to explicitly signal the path in the data plane between the source and the destination, and/or between replication points and elimination points if this functionality is used.

#5 Flow duplication and merging (W)

The functionality of replicating a packet exists in IPv6 but is limited to multicast flows. In order to enforce replicated packets to take different routes and eventually again merge flow (bring them to a specific merging point), IP-in-IP encapsulation and Segment Routing could be leveraged to signal a segment in a packet. A replication point would insert a different routing header in each copy it makes, the routing header providing explicitly the hops to the merging point for that particular replica of the packet, in a strict or in a loose source routing fashion. A flow merging point would pop the routing headers from the various copies it gets and do the rest of the required processing for merging the two flows into one flow.

#6 Operations, Administration and Maintenance (M/W)

IPv6 enjoys the existing toolbox for generic IP network management. However, IPv6 specific management features are still not at the level comparable to that of IPv4. Particular areas of concerns are those that are IPv6 specific, for example, related to neighbor discovery protocol (ND), stateless address autoconfiguration (SLAAC), subscriber identification, and security. While the standards are already mostly in place the implementations in deployed equipment can be lacking or inadequate for commercial deployments. This is larger issue with older existing equipment.

#8 Class and quality of service capabilities (W)

IPv6 provides support for CoS and QoS. CoS is provided by DiffServ which is enabled by IP header differentiated services code point (DSCP) and QoS is defined as part of RSVP [RFC2205]. DiffServ support is widely available, while RSVP for IP packets is generally not supported.

#9 Packet traceability (W)

The traceability of replicated packets involves the capability to resolve which replication point issued a particular copy of a packet, which segment was intended for that replica, and which particular packet of which particular flow this is. Sequence also depends on the sequencing mechanism. As an example, the replication point may be indicated as the source of the packet if IP-in-IP encapsulation is used to forward along segments. Another alternate to IP-in-IP tunneling along segments would be to protect the original source address in a destination option similar to the Home Address option [RFC6275] and then use the address of the replication point as source in the IP header.

The traceability also involves the capability to determine if a particular segment is operational. While IPv6 as such has no support for reversing a path, it appears source route extensions such as the one defined for segment routing could be used for tracing purposes. Though it is not a usual practice, IPv6 [RFC2460] expects that a Source Route path may be reversed, and the standard insists that a node must not include the reverse of a Routing Header in the response unless the received Routing Header was authenticated.

#10 Technical maturity (M/W)

IPv6 has been around about 20 years. However, large scale global and commercial IPv6 deployments are rather new dating only few years back to around 2012. While IPv6 has proven itself for best effort traffic, DiffServ usage is less common and QoS capabilities are not currently present. Additional, there are number of small issues to work on as they show up once operations experience grows.

The Cisco 6Lab site [1] provides information on IPv6 deployment per country, indicating figures for prefixes, transit AS, content and users. Per this site, many countries, including Canada, Brazil, the USA, Germany, France, Japan, Portugal, Sweden, Finland, Norway, Greece, and Ecuador, achieve a deployment ratio above 30 percent, and the overall adoption reported by Google Statistics [2] is now above 10 percent.

5.1.1.3. Summary

IPv6 supports a significant portion of the identified DetNet data plane criteria today. There are aspects of the DetNet data plane that are not fully supported, notably QoS, but these can be incrementally added or supplemented by the underlying sub-network layer. IPv6 may be a choice as the DetNet Transport layer in networks where other technologies such as MPLS are not deployed.

5.1.2. Native IPv4 transport

5.1.2.1. Solution description

IPv4 [RFC0791] is in principle the same as IPv6, except that it has a smaller address space. However, IPv6 was designed around the fact that extension headers are an integral part of the protocol and operation from the beginning, although the practice may some times prove differently [RFC7872]. IPv4 does support header options, but these have historically not been supported on in hardware-based

forwarding so are generally blocked or handled at a much slower rate. In either case, the use of IP header options is generally avoided. In the context of deterministic networking data plane solutions the major difference between IPv4 and IPv6 seems to be the practical support for header extensibility. Anything below and above the IP header independent of the version is practically the same.

5.1.2.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

The fixed header of an IPv4 packet is 20 bytes [RFC0791]. IP options add overhead, but are not generally used and are not considered as part of this document.

#2 Flow identification (W)

The IPv4 header has a 16-bit identification field that was originally intended for assisting fragmentation and reassembly of IPv4 packets as described in [RFC0791]. The identification field has also been proposed to be used for actually identifying flows between two IP addresses and a given protocol for detecting and removing duplicate packets [RFC1122]. However, recent update [RFC6864] to both [RFC0791] and [RFC1122] restricts the use of IPv4 identification field only to fragmentation purposes.

The IPv4 also has a stream identifier option [RFC0791], which contains a 16-bit SATNET stream identifier. However, the option has been deprecated [RFC6814]. The conclusion is that stream identification does not work nicely with IPv4 header alone and a traditional 5-tuple identification might not also be enough in a case of a flow duplication or encrypted flows. For a working solution, upper layer protocol headers such as RTP or PWs may be required for unambiguous flow identification. There is also emerging work within the IETF that may provide new flow identification alternatives.

#4 Explicit routes (W)

IPv4 has two source routing option specified: the loose source and record route option (LSRR), and the strict source and record route option (SSRR) [RFC0791]. The support of these options in the Internet is questionable but within a closed network the support may be assumed. But as both these options use IP header options, which are generally not supported in hardware, use of these options are questionable. Of course, the same options of SDN and SR approaches discussed above for IPv6 may be equally applicable to IPv4.

#5 Flow duplication and merging (W/N)

The functionality of replicating a packet exists in IPv4 but is limited to multicast flows. In general the issue regarding the IPv6 packet replication also applies to IPv4. Duplicate packet detection for IPv4 is studied in [RFC6621] to a great detail in the context of simplified multicast forwarding. In general there is no good way to detect duplicated packets for IPv4 without additional upper layer protocol support.

#6 Operations, Administration and Maintenance (M)

IPv4 enjoys the extensive and "complete" existing toolbox for generic IP network management.

#8 Class and quality of service capabilities (M/W)

IPv4 provides support for CoS and QoS. CoS is provided by DiffServ which is enabled by IP header differentiated services code point (DSCP) and QoS is defined as part of RSVP [RFC2205]. DiffServ support is widely available, while RSVP for IP packets is generally not supported.

#9 Packet traceability (W)

The IPv4 has similar needs and requirements for traceability as IPv6 (see Section 5.1.1.2). The IPv4 has a traceroute option [RFC6814] that could be used to record the route the packet took. However, the option has been deprecated [RFC6814].

#10 Technical maturity (M/W)

IPv4 can be considered mature technology with over 30 years of implementation, deployment and operations experience. As with IPv6, today's commercial implementations and deployments of IPv4 generally lack any support for QoS.

5.1.2.3. Summary

The IPv4 has specifications to support most of the identified DetNet data plane criteria today. However, several of those have already been deprecated or their wide support is not guaranteed. The DetNet data plane criteria that are not fully supported could be incrementally added or supplemented by the underlying sub-network layer. Unfortunately, the IPv4 has had limited success getting its extensions deployed at large. However, introducing new extensions might have a better success in closed networks (like DetNet) than in

Internet. Due to the popularity of the IPv4, it should be considered as a potential choice for the DetNet Transport layer.

5.1.3. Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching Architecture (MPLS) [RFC3031] and its variants, MPLS with Traffic Engineering (MPLS-TE) [RFC3209] and [RFC3473], and MPLS Transport Profile (MPLS-TP) [RFC5921] is a widely deployed technology that switches traffic based on MPLS label stacks [RFC3032] and [RFC5960]. MPLS is the foundation for Pseudowire-based services Section 5.2.3 and emerging technologies such as Bit-Indexed Explicit Replication (BIER) Section 5.1.4 and Source Packet Routing [3].

MPLS supports the equivalent of both the DetNet Service and DetNet Transport layers, and provides a very rich set of mechanisms that can be reused directly, and perhaps augmented in certain cases, to deliver DetNet services. At the DetNet Transport layer, MPLS provides forwarding, protection and OAM services. At the DetNet Service Layer it provides client service adaption, directly, via Pseudowires Section 5.2.3 and via other label-like mechanisms such as EPVN Section 5.2.4. A representation of these options are shown in Figure 7.

```
   PW-Based               EVPN Labeled                 IP
   Services                  Services               Transport
 |------------|  |-----------------------------|  |------------|

   Emulated       EVPN over LSP   EVPN w/ ESI ID        IP
   Service
                                  +------------+
                                  |  Payload   |
 +------------+   +------------+  +------------+             (Service)
 | PW Payload |   |  Payload   |  |ESI Lbl(S=1)|
 +------------+   +------------+  +------------+  +------------+
 |PW Lbl(S=1) |   |VPN Lbl(S=1)|  |VPN Lbl(S=0)|  |     IP     |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 |LSP Lbl(S=0)|   |LSP Lbl(S=0)|  |LSP Lbl(S=0)|  |LSP Lbl(S=1)|
 +------------+   +------------+  +------------+  +------------+
       .                .               .               .
       .                .               .               .    (Transport)
       .                .               .               .

~~~~~~~~~~~ denotes DetNet Service <-> DetNet Transport layer boundary
```

Figure 7: MPLS-based Services

MPLS can be controlled in a number of ways including via a control plane, via the management plane, or via centralized controller (SDN) based approaches. MPLS also provides standard control plane reference points. Additional information on MPLS architecture and control can be found in [RFC5921]. A summary of MPLS control plane related functions can be found in [RFC6373]. The remainder of this section will focus [RFC6373]. The remainder of this section will focus on the MPLS transport data plane, additional information on the MPLS service data plane can be found below in Section 5.2.2.

5.1.3.1. Solution description

The following draws heavily from [RFC5960].

Encapsulation and forwarding of packets traversing MPLS LSPs follows standard MPLS packet encapsulation and forwarding as defined in [RFC3031], [RFC3032], [RFC5331], and [RFC5332].

Data plane Quality of Service capabilities are included in the MPLS in the form of Traffic Engineered (TE) LSPs [RFC3209] and the MPLS Differentiated Services (DiffServ) architecture [RFC3270]. Both E-LSP and L-LSP MPLS DiffServ modes are defined. The Traffic Class field (formerly the EXP field) of an MPLS label follows the definition of [RFC5462] and [RFC3270].

Except for transient packet reordering that may occur, for example, during fault conditions, packets are delivered in order on L-LSPs, and on E-LSPs within a specific ordered aggregate.

The Uniform, Pipe, and Short Pipe DiffServ tunneling and TTL processing models are described in [RFC3270] and [RFC3443] and may be used for MPLS LSPs.

Equal-Cost Multi-Path (ECMP) load-balancing is possible with MPLS LSPs and can be avoided using a number of techniques. The same holds for Penultimate Hop Popping (PHP).

MPLS includes the following LSP types:

o Point-to-point unidirectional
o Point-to-point associated bidirectional
o Point-to-point co-routed bidirectional
o Point-to-multipoint unidirectional

Point-to-point unidirectional LSPs are supported by the basic MPLS architecture [RFC3031].

A point-to-point associated bidirectional LSP between LSRs A and B consists of two unidirectional point-to-point LSPs, one from A to B and the other from B to A, which are regarded as a pair providing a single logical bidirectional transport path.

A point-to-point co-routed bidirectional LSP is a point-to-point associated bidirectional LSP with the additional constraint that its two unidirectional component LSPs in each direction follow the same path (in terms of both nodes and links). An important property of co-routed bidirectional LSPs is that their unidirectional component LSPs share fate.

A point-to-multipoint unidirectional LSP functions in the same manner in the data plane, with respect to basic label processing and packet-switching operations, as a point-to-point unidirectional LSP, with one difference: an LSR may have more than one (egress interface, outgoing label) pair associated with the LSP, and any packet it transmits on the LSP is transmitted out all associated egress interfaces. Point-to-multipoint LSPs are described in [RFC4875] and [RFC5332]. TTL processing and exception handling for point-to-multipoint LSPs is the same as for point-to-point LSPs.

Additional data plane capabilities include Linear Protection, [RFC6378] and [RFC7271]. And the in progress work on MPLS support for time synchronization [I-D.ietf-mpls-residence-time].

5.1.3.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

There are two perspectives to consider when looking at encapsulation. The first is encapsulation to support services. These considerations are part of the DetNet service layer and are covered below, see Sections 5.2.3 and 5.2.4.

The second perspective relates to encapsulation, if any, is needed to transport packets across network. In this case, the MPLS label stack, [RFC3032] is used to identify flows across a network. MPLS labels are compact and highly flexible. They can be stacked to support client adaptation, protection, network layering, source routing, etc.

The number of DetNet Transport layer specific labels is flexible and support a wide range of applicable functions and MPLS domain characteristics (e.g., TE-tunnels, Hierarchical-LSPs, etc.).

#2 Flow identification (M)

MPLS label stacks provide highly flexible ways to identify flows. Basically, they enable the complete separation of traffic classification from traffic treatment and thereby enable arbitrary combinations of both.

For the DetNet flow identification the MPLS label stack can be used to support n-layers of DetNet flow identification. For example, using dedicated LSP per DetNet flow would simplify flow identification for intermediate transport nodes, and additional hierarchical LSPs could be used to facilitate scaling.

#4 Explicit routes (M)

MPLS supports explicit routes based on how LSPs are established, e.g., via TE explicit routes [RFC3209]. Additional, but not required, capabilities are being defined as part of Segment Routing (SR) [I-D.ietf-spring-segment-routing].

#5 Flow duplication and merging (M/W)

MPLS as DetNet Transport layer supports the replication via point-to-multipoint LSPs. At the MPLS LSP level, there are mechanisms defined to provide 1+1 protection, which could help realizing the flow merging function. The current definitions [RFC6378] and [RFC7271] use OAM mechanisms to support and coordinate protection switching and packet loss is possible during a switch. While such this level of protection may be sufficient for many DetNet applications, when truly hitless (i.e., zero loss) switching is required, additional mechanisms will be needed. It is expected that these additional mechanisms will be defined at a DetNet layer.

#6 Operations, Administration and Maintenance (M)

MPLS already includes a rich set of OAM functions at both the Service and Transport Layers. This includes LSP ping [ref] and those enabled via the MPLS Generic Associated Channel [RFC5586] and registered by IANA [4].

#8 Class and quality of service capabilities (M/W)

As previously mentioned, Data plane Quality of Service capabilities are included in the MPLS in the form of Traffic Engineered (TE) LSPs [RFC3209] and the MPLS Differentiated

Services (DiffServ) architecture [RFC3270]. Both E-LSP and L-LSP MPLS DiffServ modes are defined. The Traffic Class field (formerly the EXP field) of an MPLS label follows the definition of [RFC5462] and [RFC3270]. One potential open area of work is synchronized, time based scheduling. Another is shaping, which is generally not supported in shipping MPLS hardware.

#9 Packet traceability (M)

MPLS supports multiple tracing mechanisms. A control based one is defined in [RFC3209]. An OAM based mechanism is defined in MPLS On-Demand Connectivity Verification and Route Tracing [RFC6426].

#10 Technical maturity (M)

MPLS as a mature technology that has been widely deployed in many networks for many years. Numerous vendor products and multiple generations of MPLS hardware have been built and deployed.

5.1.3.3. Summary

MPLS is a mature technology that has been widely deployed. Numerous vendor products and multiple generations of MPLS hardware have been built and deployed. MPLS LSPs support a significant portion of the identified DetNet data plane criteria today. Aspects of the DetNet data plane that are not fully supported can be incrementally added. It's worth noting that a number of limitations are in shipping hardware, versus at the protocol specification level, e.g., shaping.

5.1.4. Bit Indexed Explicit Replication (BIER)

Bit Indexed Explicit Replication [I-D.ietf-bier-architecture] (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 destinations that are intended for a particular packet, which means that 1) the source is aware of the individual destinations 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. Section 5.1.4 discusses the applicability of BIER for replication in the DetNet.

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 the more classical BIER is mainly a multicast technology that typically leverages a unicast distributed control plane through IGP extensions, BIER-TE is mainly a unicast technology that leverages a central computation to setup path, compute segments and install the mapping in the intermediate nodes. Section 5.1.5 discusses the applicability of BIER-TE for replication, traceability and OAM operations in DetNet.

Bit-Indexed Explicit Replication (BIER) layer may be considered to be included into Deterministic Networking data plane solution. Encapsulation of a BIER packet in MPLS network presented in Figure 8

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                   Label Stack Element                         |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                   Label Stack Element                         |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |              BIER-MPLS label          |     |1|               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |0 1 0 1|  Ver  |  Len  |              Entropy                  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                BitString  (first 32 bits)                     ~
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ~                                                               ~
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ~                BitString  (last 32 bits)                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |OAM|     Reserved      | Proto |            BFIR-id            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: BIER packet in MPLS encapsulation

5.1.4.1. Solution description

The DetNet may be presented in BIER as distinctive payload type with its own Proto(col) ID. Then it is likely that DetNet will have the header that would identify:

- Version;
- Sequence Number;
- Timestamp;
- Payload type, e.g. data vs. OAM.

DetNet node, collocated with BFIR, may use multiple BIER sub-domains to create replicated flows. Downstream DetNet nodes, collocated with BFER, would terminate redundant flows based on Sequence Number and/or

Timestamp information. Such DetNet may be BFER in one BIER sub-domain and BFIR in another. Thus DetNet flow would traverse several BIER sub-domains.

```
                                  +-----+
                                  |  A  |
                                  +-----+
                                   /   \
                                  .     .
                                 /       .
                                .         \
                               /           .
                              .             .
                             /               \
                        +-----+             +-----+
                        |  B  |             |  C  |
                        +-----+             +-----+
                         /   \               /   \
                        .     .             .     .
                       /       \           .       .
                      .         .         /         \
                     /           \       .           .
                    .             .     .             .
                   /               \   /               \
               +-----+            +-----+           +-----+
               |  D  |            |  E  |           |  F  |
               +-----+            +-----+           +-----+
                  \                .  .               /
                   .              .    .             .
                    \            .      .           .
                     .          .        .         /
                      \        .          .       .
                        .     .            .     .
                         \   .              .   /
                        +-----+            +-----+
                        |  G  |            |  H  |
                        +-----+            +-----+
```

Figure 9: DetNet in BIER domain

Consider DetNet flow that must traverse BIER enabled domain from A to G and H. DetNet may use three BIER subdomains:

- A-B-D-E-G (dash-dot): A is BFIR, E and G are BFERs,
- A-C-E-F-H (dash-double-dot): A is BFIR, E and H are BFERs,
- E-G-H (dotted): E is BFIR, G and H are BFERs.

DetNet node A sends DetNet into red and purple BIER sub-domains. DetNet node E receives DetNet packet and sends into green sub-domain while terminating duplicates and those that deemed too-late.

DetNet nodes G and H receive DetNet flows, terminate duplicates and those that are too-late.

5.1.4.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

BIER over MPLS network encapsulation (will refer as "BIER over MPLS" further for short), Figure 8, is being defined [I-D. ietf-bier-mpls-encapsulation] within the BIER working group.

#2 Flow identification (M)

Flow identification and separation can be achieved through use of BIER domains and/or Entropy value in the BIER over MPLS, Figure 8.

#4 Explicit routes (M)

Explicit routes may be used as underlay for BIER domain. BIER underlay may be calculated using PCE and instantiated using any southbound mechanism.

#5 Flow duplication and merging (M/W)

Packet replication, as indicated by its name, is core function of the Bit-Indexed Explicit Replication. Elimination of the duplicates and/or too-late packets cannot be done within BIER sub-domain but may be done at DetNet overlay at the edge of the BIER sub-domain.

[Editor's note: how about the flow merging?]

#6 Operations, Administration and Maintenance (M/W)

BIER over MPLS guarantees that OAM is fate-sharing, i.e. in-band with a data flow being monitored or measured. Additionally, BIER over MPLS enables passive performance measurement, e.g. with the marking method [I-D.mirsky-bier-pmmm-oam]. Some OAM protocols, e.g. can be applied and used in BIER over MPLS as demonstrated [I-D.ooamdt-rtgwg-oam-gap-analysis], while new protocols being worked on, e.g. ping/traceroute [I-D.kumarzheng-bier-ping] or Path MTU Discovery [I-D.mirsky-bier-path-mtu-discovery].

#8 Class and quality of service capabilities (M/W)

Class of Service can be inherited from the underlay of the particular BIER sub-domain. Quality of Service, i.e. scheduling and bandwidth reservations can be used among other constrains in calculating explicit path for the BIER sub-domain's underlay.

#9 Packet traceability (W)

Ability to do passive performance measurement by using OAM field of the BIER over MPLS, Figure 8, is unmatched and significantly simplifies truly passive tracing of selected flows and packets within them.

#10 Technical maturity (W)

The BIER over MPLS is nearing finalization within the BIER WG and several experimental implementations are expected soon.

5.1.4.3. Summary

BIER over MPLS supports a significant portion of the identified DetNet data plane requirements, including controlled packet replication, traffic engineering, while some requirements, e.g. duplicate and too-late packet elimination may be realized as function of the DetNet overlay. BIER over MPLS is a viable candidate as the DetNet Transport layer in MPLS networks.

5.1.5. BIER - Traffic Engineering (BIER-TE)

An alternate use of Bit-Indexed Explicit Replication (BIER) uses bits in the BitString to represent adjacencies as opposed to destinations, as discussed in BIER Traffic Engineering (TE) [I-D.eckert-bier-te-arch].

The proposed function of BIER-TE in the DetNet data plane is to control the process of replication and elimination, as opposed to the identification of the flows or and the sequencing of packets within a flow.

At the path ingress, BIER-TE identifies the adjacencies that are activated for this packet (under the rule of the controller). At the egress, BIER-TE is used to identify the adjacencies where transmission failed. This information is passed to the controller, which in turn can modify the active adjacencies for the next packets.

The value is that the replication can be controlled and monitored in a loop that may involve an external controller, with the granularity of a packet and an adjacency .

5.1.5.1. Solution description

BIER-TE enables to activate the replication and elimination functions in a manner that is abstract to the data plane forwarding information. An adjacency, which is represented by a bit in the BIER header, can correspond in the data plane to an Ethernet hop, a Label Switched Path, or it can correspond to an IPv6 loose or strict source routed path.

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

- 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 10: Ladder Shape with replication and elimination Points

- 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 11: Assigning Bits

- 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 12: 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. Figure 8 illustrates a BIER header as encapsulated within MPLS. 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).

5.1.5.2. Analysis and Discussion

#1 Encapsulation and overhead (W/M)

The size of the BIER header depends on the number of segments in the particular path. It is very concise considering the amount of information that is carried (control of replication, traceability, and measurement of the reliability of the segments).

#2 Flow identification (N)

Some fields in the BIER header could be used to identify the flows but they are not the primary purpose, so it's probably not a good idea.

#4 Explicit routes (N)

A separate procedure must be used to set up the paths and allocate the bits for the adjacencies. The bits should be distributed as a form of tag by the route setup protocol. This procedure requires more work and is separate from the data plane method that is described here.

#5 Flow duplication and merging M/W)

The bitmap expresses in a very concise fashion which replication and merging (and elimination) should take place for a given packet. It also enables to control that process on a per packet basis, depending on the loss that it enables to measure. The net result is that a complex path may be installed with all the possibilities and that the decision of which possibilities are used is controlled in the data plane.

#6 Operations, Administration and Maintenance (W)

The setting of the bits at arrival enables to determine which adjacencies worked and which did not, enabling a dynamic control of the replication and elimination process. This is a form of OAM that is in-band with the data stream as opposed to leveraging separate packets, which is a more accurate information on the reliability of the link for the user.

#8 Class and quality of service capabilities (N)

BIER-TE does not signal that explicitly.

#9 Packet traceability (W)

This is a strong point of the solution. The solution enables to determine which is the current segment that a given packet is expected to traverse, which node performed the replication and which should perform the elimination if any

#10 Technical maturity (W)

Some components of the technology are more mature, e.g. segment routing and BIER. Yet, the overall solution has never been deployed as is not fully defined. It should be noted that the definition of the BIER-TE solution is outside the scope of the DetNet WG charter.

5.1.5.3. Summary

BIER-TE occupies a particular position in the DetNet data plane. 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:

- control which replication take place on a per packet basis, so that replication points can be configured but not actually utilized
- trace the replication activity and determine which node replicated a particular packet
- 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.

However, as noted earlier, BIER-TE is not yet fully specified and the required specification work is outside the scope of the current DetNet WG charter.

5.2. DetNet Service layer technologies

5.2.1. Generic Routing Encapsulation (GRE)

5.2.1.1. Solution description

Generic Routing Encapsulation (GRE) [RFC2784] provides an encapsulation of an arbitrary network layer protocol over another arbitrary network layer protocol. The encapsulation of a GRE packet can be found in Figure 13.

```
                  +--------------------------------+
                  |        Delivery Header         |
                  +--------------------------------+
                  |          GRE Header            |
                  +--------------------------------+
                  |         Payload packet         |
                  +--------------------------------+
```

Figure 13: Encapsulation of a GRE packet

Based on RFC2784, [RFC2890] further includes sequencing number and Key in optional fields of the GRE header, which may help to transport DetNet traffic flows over IP networks. The format of a GRE header is presented in Figure 14.

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |C| |K|S|  Reserved0      | Ver |          Protocol Type          |
    +-----------------------------------------------------------------+
    |      Checksum (optional)      |        Reserved1 (Optional)     |
    +-----------------------------------------------------------------+
    |                        Key (optional)                           |
    +-----------------------------------------------------------------+
    |                  Sequence Number (optional)                     |
    +-----------------------------------------------------------------+
```

Figure 14: Format of a GRE header

5.2.1.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

GRE can provide encapsulation at the service layer over the transport layer. A new protocol type for DetNet traffic should be allocated as an "Ether Type" in [RFC3232] and in IANA Ethernet Numbers [5]. The fixed header of a GRE packet is 4 octets while the maximum header is 16 octets with optional fields in Figure 14.

#2 Flow identification (W)

There is no flow identification field in GRE header. However, it can rely on the flow identification mechanism applied in the delivery protocols, such as flow identification stated in IP Sections 5.1.1 and 5.1.2 when the delivery protocols are IPv6 and IPv4 respectively. Alternatively, the Key field can also be extended to carry the flow identification. The size of Key field is 4 octets.

#3 Packet sequencing and duplicate elimination (M/W)

As stated in Section 5.2.1, GRE provides an optional sequencing number in its header to provide sequencing services for packets. The size of the sequencing number is 32 bits. The GRE header could be extended to indicate the duplicated packets by defining a flag in reserved fields or using the sequencing number of a flow.

#5 Flow duplication and merging (W/N)

GRE has no flow/packet replication and merging support in its header. It can use the transport IPv4/IPv6 protocols at the transport layer to replicate the packets and take the different routes as discussed in Section 5.1.1 and Section 5.1.2.

#6 Operations, Administration and Maintenance (M)

GRE uses the network management provided by the IP protocols as transport layer.

#8 Class and quality of service capabilities (W)

For the class of service capability, an optional code point field to indicate CoS of the traffic could be added into the GRE header. Otherwise, GRE can reuse the class and quality of service of delivery protocols at transport layer such as IPv6 and IPv4 stated in Section 5.1.1 and Section 5.1.2.

#10 Technical maturity (M)

GRE has been developed over 20 years. The delivery protocol mostly used is IPv4, while the IPv6 support for GRE is to be standardized now in IETF as [RFC7676]. Due to its good extensibility, GRE has also been extended to support network virtualization in Data Center, which is NVGRE [RFC7637].

5.2.1.3. Summary

As a tunneling protocol, GRE can encapsulate a wide variety of network layer protocols over another network layer, which can naturally serve as the service layer protocol for DetNet. Currently, it supports a portion of the Detnet service layer criteria, and still some are not fully supported but can be incrementally added or supported by delivery protocols at as the transport layer. In general, GRE can be a choice as the DetNet service layer and can work with IPv6 and IPv4 as the DetNet Transport layer.

5.2.2. MPLS-based Services for DetNet

MPLS based technologies supports both the DetNet Service and DetNet Transport layers. This, as well as a general overview of MPLS, is covered above in Section 5.1.3. These sections focus on the DetNet Service Layer it provides client service adaption, via Pseudowires Section 5.2.3 and via native and other label-like mechanisms such as EPVN in Section 5.2.4. A representation of these options was previously discussed and is shown in Figure 7.

The following text is adapted from [RFC5921]:

> The MPLS native service adaptation functions interface the client layer network service to MPLS. For Pseudowires, these adaptation functions are the payload encapsulation described in Section 4.4 of [RFC3985] and Section 6 of [RFC5659]. For network layer client

services, the adaptation function uses the MPLS encapsulation format as defined in [RFC3032].

The purpose of this encapsulation is to abstract the data plane of the client layer network from the MPLS data plane, thus contributing to the independent operation of the MPLS network.

MPLS may itself be a client of an underlying server layer. MPLS can thus also bounded by a set of adaptation functions to this server layer network, which may itself be MPLS. These adaptation functions provide encapsulation of the MPLS frames and for the transparent transport of those frames over the server layer network.

While MPLS service can provided on and true end-system to end-system basis, it's more likely that DetNet service will be provided over Pseudowires as described in Section 5.2.3 or via an EPVN-based service described in Section 5.2.4 .

MPLS labels in the label stack may be used to identify transport paths, see Section 5.1.3, or as service identifiers. Typically a single label is used for service identification.

Packet sequencing mechanisms are added in client-related adaptation processing, see Sections 5.2.3 and 5.2.4.

The MPLS client inherits its Quality of Service (QoS) from the MPLS transport layer, which in turn inherits its QoS from the server (sub-network) layer. The server layer therefore needs to provide the necessary QoS to ensure that the MPLS client QoS commitments can be satisfied.

5.2.3. Pseudo Wire Emulation Edge-to-Edge (PWE3)

5.2.3.1. Solution description

Pseudo Wire Emulation Edge-to-Edge (PWE3) [RFC3985] or simply PseudoWires (PW) provide means of emulating the essential attributes and behaviour of a telecommunications service over a packet switched network (PSN) using IP or MPLS transport. In addition to traditional telecommunications services such as T1 line or Frame Relay, PWs also provide transport for Ethernet service [RFC4448] and for generic packet service [RFC6658]. Figure 15 illustrate the reference PWE3 stack model.

```
    +----------------+                      +----------------+
    |Emulated Service|                      |Emulated Service|
    |(e.g., Eth, ...)|<= Emulated Service =>|(e.g., Eth, ...)|
    +----------------+                      +----------------+
    |    Payload     |                      |    Payload     | CW,
    |  Encapsulation |<=== Pseudo Wire ====>|  Encapsulation | Timing,
    |                |                      |                | Seq., ..
    +----------------+                      +----------------+
    |PW Demultiplexer|                      |PW Demultiplexer|
    |   PSN Tunnel,  |<==== PSN Tunnel ====>|  PSN Tunnel,   | MPLS.
    | PSN & Physical |                      | PSN & Physical | L2TP,
    |     Layers     |                      |    Layers      | IP, ..
    +-------+--------+     ___________      +---------+------+
            |             /           \               |
            +============/     PSN     \==============+
                         \             /
                          \___________/
```

Figure 15: PWE3 protocol stack reference model

PWs appear as a good data plane solution alternative for a number of reasons. PWs are a proven and deployed technology with a rich OAM control plane [RFC4447], and enjoy the toolbox developed for MPLS networks in a case of MPLS-based PSN. Furthermore, PWs may have an optional Control Word (CW) as part of the payload encapsulation between the PSN and the emulated service that is, for example, capable of frame sequencing and duplicate detection. The encapsulation layer may also provide timing using RTP as described in Sections 5.2.2, 5.4.1 and 5.4.2 of [RFC3985] and utilized by [RFC4553][RFC5087]. Furthermore, advanced DetNet node functions are conceptually already supported by PW framework (with some added functional required), such as the DetNet Relay node modeled after the Multi-Segment PWE3 [RFC5254].

PWs can be also used if the PSN is IP, which enables the application of PWs in networks that do not have MPLS enabled in their core routers. One approach to provide PWs over IP is to provide MPLS over IP in some way and then leverage what is available for PWs over MPLS. The following standard solutions are available both for IPv4 and IPv6 to follow this approach. The different solutions have different overhead as discussed in the following subsection. The MPLS-in-IP encapsulation is specified by [RFC4023]. The IPv4 Protocol Number field or the IPv6 Next Header field is set to 137, which indicates an MPLS unicast packet. (The use of the MPLS-in-IP encapsulation for MPLS multicast packets is not supported.) The MPLS-in-GRE encapsulation is specified in [RFC4023], where the IP header (either IPv4 or IPv6) is followed by a GRE header, which is followed by an

MPLS label stack. The protocol type field in the GRE header is set to MPLS Unicast (0x8847) or Multicast (0x8848). MPLS over L2TPv3 over IP encapsulation is specified by [RFC4817]. The MPLS-in-UDP encapsulation is specified by [RFC7510], where the UDP Destination Port indicates tunneled MPLS packet and the UDP Source Port is an entropy value that is generated by the encapsulator to uniquely identify a flow. MPLS-in-UDP encapsulation can be applied to enable UDP-based ECMP (Equal-Cost Multipath) or Link Aggregation. All these solutions can be secured with IPsec [RFC4303].

5.2.3.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

PWs offer encapsulation services practically for any types of payloads over any PSN. New PW types need a code point allocation [RFC4446] and in some cases an emulated service specific document.

Specifically in the case of the MPLS PSN the PW encapsulation overhead is minimal. Typically minimum two labels and a CW is needed, which totals to 12 octets. PW type specific handling might, however, allow optimizations on the emulated service in the provider edge (PE) device's native service processing (NSP) / forwarder function. These optimizations could be used, for example, to reduce header overhead. Ethernet PWs already have rather low overhead [RFC4448]. Without a CW and VLAN tags the Ethernet header gets reduced to 14 octets (minimum Ethernet header overhead is 26).

The overhead is somewhat bigger in case of IP PSN if an MPLS over IP solution is applied to provide PWs. IP adds at least 20 (IPv4) or 40 (IPv6) bytes overhead to the PW over MPLS overhead; furthermore, the GRE, L2TPv3, or UDP header has to be taken into account if any of these further encapsulations is used.

#2 Flow identification (M)

PWs provide multiple layers of flow identification, especially in the case of the MPLS PSN. The PWs are typically prepended with an endpoint specific PW label that can be used to identify a specific PW per endpoint. Furthermore, the MPLS PSN also uses one or more labels to transport packets over a specific label switched paths (that then would carry PWs). So, a DetNet flow can be identified in this example by the service and transport layer labels. IP (and other) PSNs may need other mechanisms, such as, UDP port numbers, upper layer protocol header (like RTP) or some IP extension header to provide required flow identification.

#3 Packet sequencing and duplicate elimination (M)

As mentioned earlier PWs may contain an optional CW that is able to provide sequencing services. The size of the sequence number in the generic CW is 16 bits, which might be, depending on the used link and DetNet flow speed be too little. The PW duplicate detection mechanism is already conceptually specified [RFC3985] but no emulated service makes use of it currently.

#5 Flow duplication and merging (W)

PWs could use a (extended) version of existing transport layer provided protection mechanisms (e.g., hitless 1+1 protection) for both flow duplication and flow merging. The service layer has to provide the functionality to map DetNet flows into appropriate transport leyer connection, though.

#6 Operations, Administration and Maintenance (M/W)

PWs have rich control plane for OAM and in a case of the MPLS PSN enjoy the full control plane toolbox developed for MPLS network OAM likewise IP PSN have the full toolbox of IP network OAM tools. There could be, however, need for deterministic networking specific extensions for the mentioned control planes.

#8 Class and quality of service capabilities (M/W)

In a case of IP PSN the 6-bit differentiated services code point (DSCP) field can be used for indicating the class of service [RFC2474] and 2-bit field reserved for the explicit congestion notification (ECN) [RFC3168]. Similarly, in a case of MPLS PSN, there are 3-bit traffic class field (TC) [RFC5462] in the label reserved for for both Explicitly TC-encoded-PSC LSPs (E-LSP) [RFC3270] and ECN [RFC5129]. Due to the limited number of bits in the TC field, their use for QoS and ECN functions restricted and intended to be flexible. Although the QoS/CoS mechanism is already in place some clarifications may be required in the context of deterministic networking flows, for example, if some specific mapping between bit fields have to be done.

When PWs are used over MPLS, MPLS LSPs can be used to provide both CoS (E-LSPs and L-LSPs) and QoS (dedicated TE LSPS).

#10 Technical maturity (M)

PWs, IP and MPLS are proven technologies with wide variety of deployments and years of operational experience. Furthermore, the estimated work for missing functionality (packet replication and

elimination) does not appear to be extensive, since the existing protection mechanism already get close to what is needed from the deterministic networking data plane solution.

5.2.3.3. Summary

PseudoWires appear to be a strong candidate as the deterministic networking data plane solution alternative for the DetNet Service layer. The strong points are the technical maturity and the extensive control plane for OAM. This holds specifically for MPLS-based PSN.

Extensions are required to realize the packet replication and duplicate detection features of the deterministic networking data plane.

5.2.4. MPLS-Based Ethernet VPN (EVPN)

5.2.4.1. Solution description

MPLS-Based Ethernet VPN (EVPN), in the form documented in [RFC7432] and [RFC7209], is an increasingly popular approach to delivering MPLS-based Ethernet services and is designed to be the successor to Virtual Private LAN Service (VPLS), [RFC4664].

EVPN provides client adaptation and reuses the MPLS data plane discussed above in Section 5.2.2. While not required, the PW Control Word is also used. EVPN control is via BGP, [RFC7432], and may use TE-LSPs, e.g., controlled via [RFC3209] for MPLS transport. Additional EVPN related RFCs and in progress drafts are being developed by the BGP Enabled Services Working Group [6].

5.2.4.2. Analysis and Discussion

#1 Encapsulation and overhead (M)

EVPN generally uses a single MPLS label stack entry to support its client adaptation service. The optional addition of a second label is also supported. In certain cases PW Control Word may also be used.

#2 Flow identification (W)

EVPN currently uses labels to identify flows per {Ethernet Segment Identifier, VLAN} or per MAC level. Additional definition will be

needed to standardize identification of finer granularity DetNet flows as well as mapping of TSN services to DetNet Services.

#3 Packet sequencing and duplicate elimination (M)

Like MPLS, EVPN generally orders packets similar to Ethernet. Reordering is possible primarily during path changes and protection switching. In order to avoid misordering due to ECMP, EVPN uses the "Preferred PW MPLS Control Word" [RFC4385] (in which case EVPN inherits this function from PWs) or the entropy labels [RFC6790].

If additional ordering mechanisms are required, such mechanisms will need to be defined.

#5 Flow duplication and merging (M/W)

EVPN relies on the MPLS layer for all protection functions. See Section 5.1.3 and Section 5.2.2. Some extensions, either at the EVPN or MPLS levels, will be need to support those DetNet applications which require true hitless (i.e., zero loss) 1+1 protection switching. (Network coding may be an interesting alternative to investigate to delivering such hitless loss protection capability.)

#6 Operations, Administration and Maintenance (M/W)

Nodes supporting EVPN may participate in either or both Ethernet level and MPLS level OAM. It is likely that it may make sense to map or adapt the OAM functions at the different levels, but such has yet to be defined. [RFC6371] provides some useful background on this topic.

#8 Class and quality of service capabilities (M/W)

EVPN is largely silent on the topics of CoS and QoS, but the 802.1 TSN Ethernet and existing MPLS TE mechanisms can be directly used. The inter-working of such is new work and within the scope of DetNet. The existing MPLS mechanisms include both CoS (E-LSPs and L-LSPs) and QoS (dedicated TE LSPs).

#10 Technical maturity (M)

EVPN is a second (or third) generation MPLS-based L2VPN service standard. From a data plane standpoint it makes uses of existing MPLS data plane mechanisms. The mechanisms have been widely implemented and deployed.

5.2.4.3. Summary

EVPN is the emerging successor to VPLS. EVPN is standardized, implemented and deployed. It makes use of the mature MPLS data plane. While offering a mature and very comprehensive set of features, certain DetNet required features are not fully/directly supported and additional standardization in these areas are needed. Examples include: mapping CoS and QoS; use of labels per DetNet flow, and hitless 1+1 protection.

5.2.5. Higher layer header fields

Fields of headers belonging to higher OSI layers can be used to implement functionality that is not provided e.g., by the IPv6 or IPv4 header fields. However, this approach cannot be always applied, e.g., due to encryption. Furthermore, even if this approach is applicable, it requires deep packet inspection from the routers and switches. There are implementation dependent limits how far into the packet the lookup can be done efficiently in the fast path. When encryption is not used, a safe bet is generally between 128 and 256 octets for the maximum lookup depth. Various higher layer protocols can be applied. Some examples are provided here for the sequence numbering feature (Section 4.3).

5.2.5.1. TCP

The TCP header includes a sequence number parameter, which can be applied to detect and eliminate duplicate packets if DetNet service protection is used. As the TCP header is right after the IP header, it does not require very deep packet inspection; the 4-byte sequence number is conveyed by bits 32 through 63 of the TCP header. In addition to sequencing, the TCP header also contain source and destination port information that can be used for assisting the flow identification.

5.2.5.2. RTP

5.2.5.2.1. Solution Description

Real-time Transport Protocol (RTP) [RFC3550] is often used to deliver time critical traffic in IP networks. RTP is typically carried on top of UDP/IP. However, as noted earlier in Section 5.2.3 PseudoWires also have a well-defined way of embedding and

transposting RTP header as part of its payload encapsulation headers/ sub-layer. RTP is also augmented by its own control protocol RTCP, which monitors of the data delivery and provides minimal control and identification functionality. RTCP packets do not carry "media payload". Although both RTP and RTCP are typically used with UDP/IP transport they are designed to be independent of the underlying transport and network layers.

The RTP header includes a 2-byte sequence number, which can be used to detect and eliminate duplicate packets if DetNet service protection is used. The sequence number is conveyed by bits 16 through 31 of the RTP header. In addition to the sequence number the RTP header has also timestamp field (bits 32 through 63) that can be useful for time synchronization purposes. Furthermore, the RTP header has also one or more synchronization sources (bits starting from 64) that can potentially be useful for flow identification purposes.

5.2.5.2.2. **Analysis and Discussion**

#1 Encapsulation and overhead (M)

RTP adds minimum 12 octets of header overhead. Typically 8 octets overhead of UDP header has to be also added, at least in a case when RTP is transported over IP. Although RTCP packets do not contribute to the media payload transport they still consume overall network capacity, since all participants to an RTP session including sourcess and multicast session destinations are expected to send RTCP reports.

#2 Flow identification (M)

The RTP header contains a synchronization source (SSRC) identifier. The intent is that no two synchronization sources within the same RTP session has the same SSRC identifier.

#3 Packet sequencing and duplicate elimination (M)

The RTP header contains a 16 bit sequence number. The sequence number can be also used to detect duplicate packets.

#5 Flow duplication and merging (M/W)

RTP has precedence of being used for hitless protection switching [ST20227], which essentially is equivalent to DetNet service protection. Furthermore, recent work in IETF for RTP stream duplication [RFC7198] as a mechanism to protect media flows from packet loss is again equivalent to Detnet service protection.

#6 Operations, Administration and Maintenance (M)

RTP has its own control protocol RTCP for (minimal) management and stream monitoring purposes. Existing IP OAM tools can directly leveraged when RTP is deployed over IP transport.

#8 Class and quality of service capabilities (M/W)

TBD. [Editor's note: relies on lower layers to provide CoS/QoS]

#10 Technical maturity (M)

RTP has been deployed and used in large commercial systems for over ten years and can be considered a mature technology.

5.2.5.2.3. Summary

RTP appears to be a good candidate as the deterministic networking data plane solution alternative for the DetNet Service layer. The strong points are the technical maturity and the fact it was designed for transporting time-sensitive payload from the beginning. RTP is specifically well suited to be used with (UDP)/IP transport.

Extensions may be required to realize the packet replication and duplicate detection features of the deterministic networking data plane. However, there is already precedence of similar solutions that could potentially be leveraged [ST20227][RFC7198].

6. Summary of data plane alternatives

The following table summarizes the criteria (Section 4) used for the evaluation of data plane options.

Applicability per Alternative

Item #	Meaning
#1	Encapsulation and overhead
#2	Flow identification
#3	Packet sequencing and duplicate elimination
#4	Explicit routes
#5	Flow duplication and merging
#6	Operations, Administration and Maintenance
#8	Class and quality of service capabilities
#9	Packet traceability
#10	Technical maturity

Table 5: Evaluation criteria (#7 obsoleted)

There is no single technology that could meet all the criteria on its own. Distinguishing the DetNet Service and the DetNet Transport, as explained in (Section 3), allows a number of combinations, which can meet most of the criteria. There is no room here to evaluate all possible combinations. Therefore, only some combinations are highlighted here, which are selected based on the number of criteria that are met and the maturity of the technology (#10).

The following table summarizes the evaluation of the data plane options that can be used for the DetNet Transport Layer against the evaluation criteria. Each value in the table is from the corresponding section.

Applicability per Transport Alternative

Solution	#1	#2	#4	#5	#6	#8	#9	#10
IPv6	M	W	W	W	M	W	W	M/W
IPv4	M	W	W	W/N	M	M/W	W	M/W
MPLS	M	M	M	M/W	M	M/W	M	M
BIER	M	M	M	M/W	M/W	M/W	M	W
BIER-TE	W/M	N	N	M/W	W	N	W	W

Summarizing Transport capabilities

Table 6: DetNet Transport Layer

The following table summarizes the evaluation of the data plane options that can be used for the DetNet Service Layer against the criteria evaluation criteria. Each value in the table is from the corresponding section.

Applicability per Service Alternative

Solution	#1	#2	#3	#5	#6	#8	#10
GRE	M	W	M/W	W/N	M	W	M
PWE3	M	M	M	W	M/W	M/W	M
EVPN	M	W	M	M/W	M/W	M/W	M
RTP	M	M	M	M/W	M	M/W	M

Summarizing Service capabilities

Table 7: DetNet Service Layer

PseudoWire (Section 5.2.3) is a technology that is mature and meets most of the criteria for the DetNet Service layer as shown in the table above. From upper layer protocols PWs or RTP can be a candidate for non-MPLS PSNs. The identified work for PWs is to figure out how to implement duplicate detection for these protocols (e.g., based on [RFC3985]). In a case of RTP there is precedence of implementing packet duplication and duplicate elimination [ST20227][RFC7198].

PWs can be carried over MPLS or IP. MPLS is the most common technology that is used as PSN for PseudoWires; furthermore, MPLS is a mature technology and meets most DetNet Transport layer criteria. IPv[46] can be also used as PSN and both are mature technologies, although both generally only support CoS (DiffServ) in deployed networks. RTP is independent of the underlying transport technology and network. However, it is well suited for UDP/IP transport or embedded as a part of the PseudoWire timing sub-layer.

7. Security considerations

This document does not add any new security considerations beyond what the referenced technologies already have.

8. IANA Considerations

This document has no IANA considerations.

9. Acknowledgements

The author(s) ACK and NACK.

The following people were part of the DetNet Data Plane Design Team:

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

Substantial contributions were received from:

Balazs Varga (service model)

The DetNet chairs serving during the DetNet Data Plane Design Team:

Lou Berger
Pat Thaler

10. References

10.1. Informative References

[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.finn-detnet-architecture]
Finn, N., Thubert, P., and M. Teener, "Deterministic Networking Architecture", draft-finn-detnet-architecture-07 (work in progress), July 2016.

[I-D.ietf-6man-rfc2460bis]
Deering, D. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", draft-ietf-6man-rfc2460bis-05 (work in progress), June 2016.

[I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., and D. Lebrun, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-01 (work in progress), March 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.

[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-mpls-residence-time]
Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and S. Vainshtein, "Residence Time Measurement in MPLS network", draft-ietf-mpls-residence-time-11 (work in progress), July 2016.

[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.

[I-D.ietf-sunset4-gapanalysis]
Perreault, S., Tsou, T., Zhou, C., and P. Fan, "Gap Analysis for IPv4 Sunset", draft-ietf-sunset4-gapanalysis-07 (work in progress), April 2015.

[I-D.kumarzheng-bier-ping]
Kumar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M., and G. Mirsky, "BIER Ping and Trace", draft-kumarzheng-bier-ping-03 (work in progress), July 2016.

[I-D.mirsky-bier-path-mtu-discovery]
Mirsky, G., Przygienda, T., and A. Dolganow, "Path Maximum Transmission Unit Discovery (PMTUD) for Bit Index Explicit Replication (BIER) Layer", draft-mirsky-bier-path-mtu-discovery-01 (work in progress), April 2016.

[I-D.mirsky-bier-pmmm-oam]
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-mirsky-bier-pmmm-oam-01 (work in progress), March 2016.

[I-D.ooamdt-rtgwg-oam-gap-analysis]
Mirsky, G., Nordmark, E., Pignataro, C., Kumar, N., Kumar, D., Chen, M., Yizhou, L., Mozes, D., Networks, J., and i. ibagdona@gmail.com, "Operations, Administration and Maintenance (OAM) for Overlay Networks: Gap Analysis", draft-ooamdt-rtgwg-oam-gap-analysis-02 (work in progress), July 2016.

[I-D.ooamdt-rtgwg-ooam-requirement]
Kumar, N., Pignataro, C., Kumar, D., Mirsky, G., Chen, M., Nordmark, E., Networks, J., and D. Mozes, "Overlay OAM Requirements", draft-ooamdt-rtgwg-ooam-requirement-01 (work in progress), July 2016.

[IEEE802.1Qbv]
IEEE, "Enhancements for Scheduled Traffic", 2016, <http://www.ieee802.org/1/files/private/bv-drafts/>.

[IEEE802.1Qca]
IEEE 802.1, "IEEE 802.1Qca Bridges and Bridged Networks - Amendment 24: Path Control and Reservation", IEEE P802.1Qca/D2.1 P802.1Qca, June 2015, <https://standards.ieee.org/findstds/standard/802.1Qca-2015.html>.

[IEEE802.1Qch]
IEEE, "Cyclic Queuing and Forwarding", 2016, <http://www.ieee802.org/1/files/private/ch-drafts/>.

[IEEE8021CB]
Finn, N., "Draft Standard for Local and metropolitan area networks - Seamless Redundancy", IEEE P802.1CB /D2.1 P802.1CB, December 2015, <http://www.ieee802.org/1/files/private/cb-drafts/d2/802-1CB-d2-1.pdf>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <http://www.rfc-editor.org/info/rfc2205>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <http://www.rfc-editor.org/info/rfc2474>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <http://www.rfc-editor.org/info/rfc2784>.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <http://www.rfc-editor.org/info/rfc2890>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <http://www.rfc-editor.org/info/rfc3032>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <http://www.rfc-editor.org/info/rfc3168>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <http://www.rfc-editor.org/info/rfc3232>.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <http://www.rfc-editor.org/info/rfc3270>.

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <http://www.rfc-editor.org/info/rfc3443>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <http://www.rfc-editor.org/info/rfc3985>.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <http://www.rfc-editor.org/info/rfc4023>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <http://www.rfc-editor.org/info/rfc4385>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <http://www.rfc-editor.org/info/rfc4446>.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, DOI 10.17487/RFC4447, April 2006, <http://www.rfc-editor.org/info/rfc4447>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <http://www.rfc-editor.org/info/rfc4448>.

[RFC4553] Vainshtein, A., Ed. and YJ. Stein, Ed., "Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)", RFC 4553, DOI 10.17487/RFC4553, June 2006, <http://www.rfc-editor.org/info/rfc4553>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <http://www.rfc-editor.org/info/rfc4664>.

[RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <http://www.rfc-editor.org/info/rfc4817>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <http://www.rfc-editor.org/info/rfc4875>.

[RFC5087] Stein, Y(J)., Shashoua, R., Insler, R., and M. Anavi, "Time Division Multiplexing over IP (TDMoIP)", RFC 5087, DOI 10.17487/RFC5087, December 2007, <http://www.rfc-editor.org/info/rfc5087>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <http://www.rfc-editor.org/info/rfc5129>.

[RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, DOI 10.17487/RFC5254, October 2008, <http://www.rfc-editor.org/info/rfc5254>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <http://www.rfc-editor.org/info/rfc5331>.

[RFC5332] Eckert, T., Rosen, E., Ed., Aggarwal, R., and Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, DOI 10.17487/RFC5332, August 2008, <http://www.rfc-editor.org/info/rfc5332>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <http://www.rfc-editor.org/info/rfc5440>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <http://www.rfc-editor.org/info/rfc5462>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <http://www.rfc-editor.org/info/rfc5586>.

[RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <http://www.rfc-editor.org/info/rfc5659>.

[RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <http://www.rfc-editor.org/info/rfc5921>.

[RFC5960] Frost, D., Ed., Bryant, S., Ed., and M. Bocci, Ed., "MPLS Transport Profile Data Plane Architecture", RFC 5960, DOI 10.17487/RFC5960, August 2010, <http://www.rfc-editor.org/info/rfc5960>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <http://www.rfc-editor.org/info/rfc6275>.

[RFC6371] Busi, I., Ed. and D. Allan, Ed., "Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks", RFC 6371, DOI 10.17487/RFC6371, September 2011, <http://www.rfc-editor.org/info/rfc6371>.

[RFC6373] Andersson, L., Ed., Berger, L., Ed., Fang, L., Ed., Bitar, N., Ed., and E. Gray, Ed., "MPLS Transport Profile (MPLS-TP) Control Plane Framework", RFC 6373, DOI 10.17487/RFC6373, September 2011, <http://www.rfc-editor.org/info/rfc6373>.

[RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <http://www.rfc-editor.org/info/rfc6378>.

[RFC6426] Gray, E., Bahadur, N., Boutros, S., and R. Aggarwal, "MPLS On-Demand Connectivity Verification and Route Tracing", RFC 6426, DOI 10.17487/RFC6426, November 2011, <http://www.rfc-editor.org/info/rfc6426>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <http://www.rfc-editor.org/info/rfc6437>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <http://www.rfc-editor.org/info/rfc6564>.

[RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <http://www.rfc-editor.org/info/rfc6621>.

[RFC6658] Bryant, S., Ed., Martini, L., Swallow, G., and A. Malis, "Packet Pseudowire Encapsulation over an MPLS PSN", RFC 6658, DOI 10.17487/RFC6658, July 2012, <http://www.rfc-editor.org/info/rfc6658>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <http://www.rfc-editor.org/info/rfc6790>.

[RFC6814] Pignataro, C. and F. Gont, "Formally Deprecating Some IPv4 Options", RFC 6814, DOI 10.17487/RFC6814, November 2012, <http://www.rfc-editor.org/info/rfc6814>.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <http://www.rfc-editor.org/info/rfc6864>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <http://www.rfc-editor.org/info/rfc7045>.

[RFC7198] Begen, A. and C. Perkins, "Duplicating RTP Streams", RFC 7198, DOI 10.17487/RFC7198, April 2014, <http://www.rfc-editor.org/info/rfc7198>.

[RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, DOI 10.17487/RFC7209, May 2014, <http://www.rfc-editor.org/info/rfc7209>.

[RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <http://www.rfc-editor.org/info/rfc7271>.

[RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <http://www.rfc-editor.org/info/rfc7399>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <http://www.rfc-editor.org/info/rfc7426>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <http://www.rfc-editor.org/info/rfc7432>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <http://www.rfc-editor.org/info/rfc7637>.

[RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <http://www.rfc-editor.org/info/rfc7676>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <http://www.rfc-editor.org/info/rfc7752>.

[RFC7813] Farkas, J., Ed., Bragg, N., Unbehagen, P., Parsons, G., Ashwood-Smith, P., and C. Bowers, "IS-IS Path Control and Reservation", RFC 7813, DOI 10.17487/RFC7813, June 2016, <http://www.rfc-editor.org/info/rfc7813>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <http://www.rfc-editor.org/info/rfc7872>.

[ST20227] SMPTE 2022, "Seamless Protection Switching of SMPTE ST 2022 IP Datagrams", ST 2022-7:2013, 2013, <https://www.smpte.org/digital-library>.

[TSNTG] IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", 2013, <http://www.IEEE802.org/1/pages/avbridges.html>.

10.2. URIs

[1] http://6lab.cisco.com/stats/

[2] https://www.google.com/intl/en/ipv6/statistics.html

[3] https://datatracker.ietf.org/wg/spring/charter/

[4] http://www.iana.org/assignments/g-ach-parameters/g-ach-parameters.xhtml

[5] http://ftp.isi.edu/in-notes/iana/assignments/ethernet-numbers

[6] https://tools.ietf.org/wg/bess/

Appendix A. Examples of combined DetNet Service and Transport layers

Authors' Addresses

Jouni Korhonen (editor)
Broadcom
3151 Zanker Road
San Jose, CA 95134
USA

Email: jouni.nospam@gmail.com

Janos Farkas
Ericsson
Konyves Kalman krt. 11/B
Budapest 1097
Hungary

Email: janos.farkas@ericsson.com

Gregory Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Pascal Thubert
Cisco

Email: pthubert@cisco.com

Yan Zhuang
Huawei

Email: zhuangyan.zhuang@huawei.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net