
Workgroup: Network Working Group
Internet-Draft: draft-ietf-detnet-yang-15
Published: 6 December 2021
Intended Status: Standards Track
Expires: 9 June 2022
Authors:

X. Geng Y. Ryoo D. Fedyk R. Rahman Z. Li
Huawei Technologies *ETRI* *LabN Consulting, L.L.C.* *Individual* *China Mobile*

Deterministic Networking (DetNet) YANG Model

Abstract

This document contains the specification for the Deterministic Networking YANG Model for configuration and operational data for DetNet Flows. The model allows for provisioning of end-to-end DetNet service on devices along the path without dependency on any signaling protocol. It also specifies operational status for flows. An operator or network controller programs the configuration of the devices.

The YANG module defined in this document conforms to the Network Management Datastore Architecture (NMDA).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on 9 June 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
2. Abbreviations
3. Terminology
4. DetNet YANG Module
 - 4.1. DetNet Application Flow YANG Attributes
 - 4.2. DetNet Service Sub-layer YANG Attributes
 - 4.3. DetNet Forwarding Sub-layer YANG Attributes
5. DetNet Flow Aggregation
6. DetNet YANG Structure Considerations
7. DetNet Configuration YANG Structures
8. DetNet Configuration YANG Model
9. IANA Considerations
10. Security Considerations
11. Contributors
12. Acknowledgments
13. References
 - 13.1. Normative References
 - 13.2. Informative References
- Appendix A. DetNet Configuration YANG Tree
- Appendix B. Examples
 - B.1. Example A-1 JSON Configuration/Operational
 - B.2. Example B-1 XML Config: Aggregation using a Forwarding Sub-layer
 - B.3. Example B-2 JSON Service Aggregation Configuration
 - B.4. Example C-1 JSON Relay Aggregation/Disaggregation Configuration
 - B.5. Example C-2 JSON Relay Aggregation Service Sub-Layer
 - B.6. Example C-3 JSON Relay Service Sub-Layer Aggregation/Disaggregation

[B.7. Example C-4 JSON Relay Service Sub-Layer Aggregation/Disaggregation](#)

[B.8. Example D-1 JSON Transit Forwarding Sub-Layer Aggregation/Disaggregation](#)

[Authors' Addresses](#)

1. Introduction

DetNet (Deterministic Networking) provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low packet loss rates and assured maximum end-to-end delivery latency. A description of the general background and concepts of DetNet can be found in [\[RFC8655\]](#).

This document defines a YANG model for DetNet based on YANG data types and modeling language defined in [\[RFC6991\]](#) and [\[RFC7950\]](#). DetNet service, which is designed for describing the characteristics of services being provided for application flows over a network, and DetNet configuration, which is designed for DetNet flow path establishment, flow status reporting, and DetNet functions configuration in order to achieve end-to-end bounded latency and zero congestion loss, are both included in this document.

2. Abbreviations

The following abbreviations are used in this document:

PEF	Packet Elimination Function
PRF	Packet Replication Function
PEOF	Packet Elimination and Ordering Functions
PERF	Packet Elimination and Replication Functions
PREOF	Packet Replication, Elimination and Ordering Functions
MPLS	Multiprotocol Label Switching

3. Terminology

This document uses the terminology defined in [\[RFC8655\]](#).

4. DetNet YANG Module

The DetNet YANG module includes DetNet App-flow, DetNet Service Sub-layer, and DetNet Forwarding Sub-layer configuration and operational objects. The corresponding attributes used in different sub-layers are defined in Section 3.1, 3.2, 3.3 respectively.

Layers of the objects typically occur in the different data instances forming the node types defined in [RFC8655]. Figure 1 illustrates the relationship between data instance node types and the included layers. Node types typically are logical per DetNet service and one DetNet service can be one node type while another is another node type on same device. This model is a controller based model because a controller or operator configures all the devices to form a service.

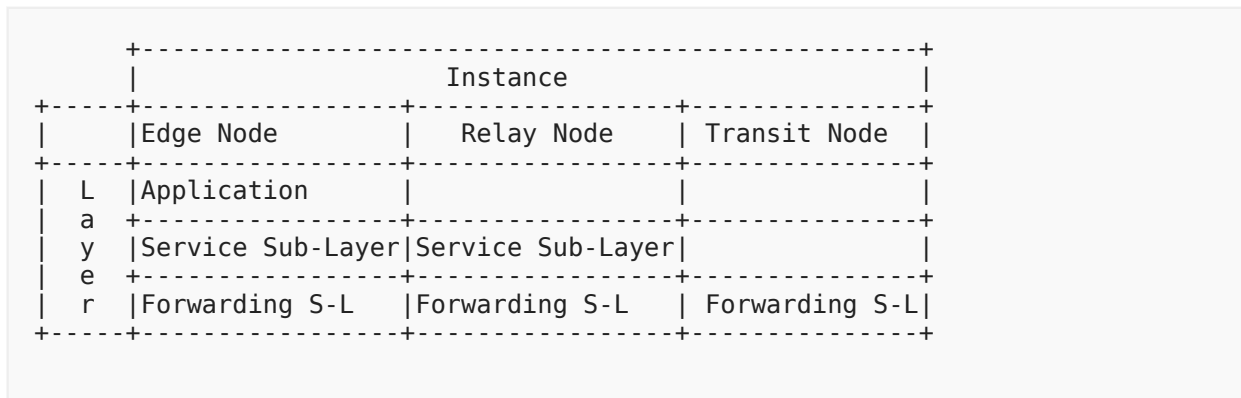


Figure 1: Detnet Layers and Node Types

All of the layers have ingress/incoming and egress/outgoing operations but any instance may be configured as only unidirectional. This means that each unidirectional flow identifier configuration is programmed starting at the ingress and flow status is reported at ingress on each end. In the MPLS cases once encapsulated, the IP six tuple parameters may not be required to be programmed again. In the IP case, without encapsulation, various IP flow id parameters must be configured along the flow path.

In the YANG model the terms source and destination are used as flow identifiers whereas ingress and egress refer to a DetNet application direction from the application edge. Ingress is to the DetNet application and egress is from the application. The terms incoming and outgoing generally represent the flow direction towards the remote application. Outgoing is viewed as going up the stack from Application to Service sub-layer to Forwarding sub-layer and incoming is the reverse. Although, in examples where there is aggregation and disaggregation outgoing relates to the aggregating output and incoming relates to the disaggregating flows.

At the egress the YANG model is handing off IP or MPLS to a next hop and a routing next hop structure is used.

4.1. DetNet Application Flow YANG Attributes

DetNet application flow is responsible for mapping between application flows and DetNet flows at the edge node(egress/ingress node). The application flows can be either layer 2 or layer 3 flows. To map a flow at the User Network Interface (UNI), the corresponding attributes are defined in [RFC9016].

4.2. DetNet Service Sub-layer YANG Attributes

DetNet service functions, e.g., DetNet tunnel initialization/termination and service protection, are provided in the DetNet service sub-layer. To support these functions, the following service attributes need to be configured:

- DetNet flow identification
- Service function indication, indicates which service function will be invoked at a DetNet edge, relay node or end station. (DetNet tunnel initialization or termination are default functions in DetNet service layer, so there is no need for explicit indication). The corresponding arguments for service functions also needs to be defined.

4.3. DetNet Forwarding Sub-layer YANG Attributes

As defined in [RFC8655], DetNet forwarding sub-layer optionally provides congestion protection for DetNet flows over paths provided by the underlying network. Explicit route is another mechanism that is used by DetNet to avoid temporary interruptions caused by the convergence of routing or bridging protocols, and it is also implemented at the DetNet forwarding sub-layer.

To support congestion protection and explicit route, the following transport layer related attributes are necessary:

- Flow Specification and Traffic Requirements, refers to [RFC9016]. These may used for resource reservation, flow shaping, filtering and policing by a control plane or other network management and control mechanisms.
- Since this model programs the data plane existing explicit route mechanisms can be reused. If a static MPLS tunnel is used as the transport tunnel, the configuration need to be at every transit node along the path. For an IP based path, the static configuration is similar to the static MPLS case. This document provides data-plane configuration of IP addresses or MPLS labels but it does not provide control plane mapping or other aspects.

5. DetNet Flow Aggregation

DetNet provides the capability of flow aggregation to improve scalability of DetNet data, management and control planes. Aggregated flows can be viewed by the some DetNet nodes as individual DetNet flows. When aggregating DetNet flows, the flows should be compatible: if bandwidth reservations are used, the reservation should be a reasonable representation of the individual reservations; if maximum delay bounds are used, the system should ensure that the aggregate does not exceed the delay bounds of the individual flows.

The DetNet YANG model defined in this document supports DetNet flow aggregation with the following functions:

- Aggregation flow encapsulation/decapsulation/identification
- Mapping individual DetNet flows to an aggregated flow

- Changing traffic specification parameters for aggregated flow

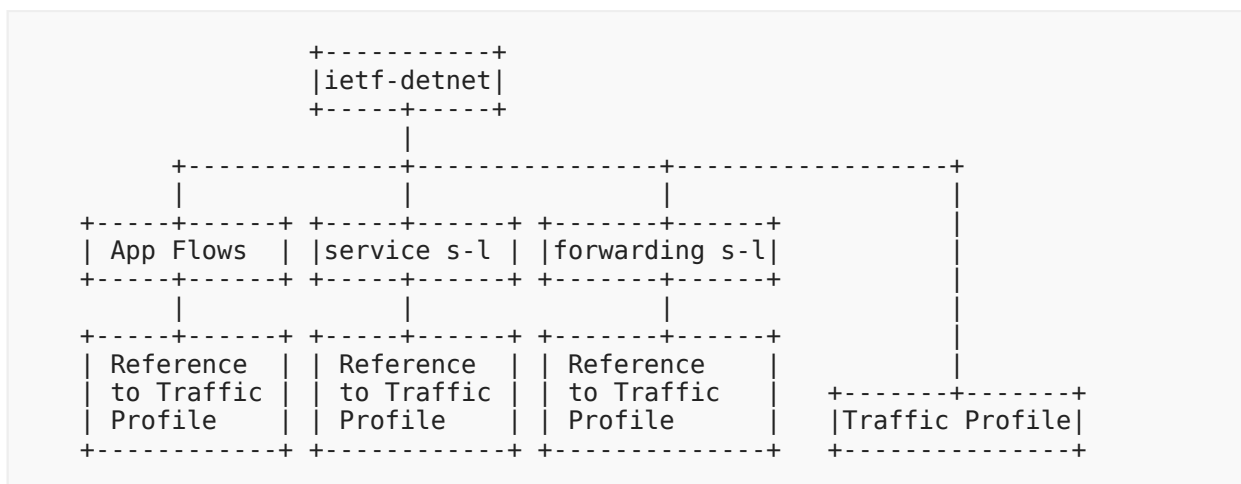
The following cases of DetNet aggregation are supported:

- Ingress node aggregates App flows into a service sub-layer of DetNet flow
- In ingress node, the service sub-layers of DetNet flows are aggregated into a forwarding sub-layer
- In ingress node, the service sub-layers of DetNet flows are aggregated into a service sub-layer of an aggregated DetNet flow
- Relay node aggregates the forwarding sub-layers DetNet flows into a forwarding sub-layer
- Relay node aggregates the service sub-layers of DetNet flows into a forwarding sub-layer
- Relay node aggregates the service sub-layers of DetNet flows into a service sub-layer of Aggregated DetNet flow
- Relay node aggregates the forwarding sub-layers of DetNet flow into a service sub-layer of Aggregated DetNet flow
- Transit node aggregates the forwarding sub-layers of DetNet flows into a forwarding sub-layer

Traffic requirements and traffic specification may be tracked for individual or aggregate flows but reserving resources and tracking the services in the aggregated flow is out of scope.

6. DetNet YANG Structure Considerations

The picture shows that the general structure of the DetNet YANG Model:



There are three layer types in DetNet YANG Model: App-flow data layer, service sub-layer and forwarding sub-layer. Additionally, the Traffic parameters are captured in a Traffic profile that can be referenced by any of the layers.

Below is a summary YANG tree showing the major items. A complete YANG tree is in section [Appendix A](#).

A traffic profile can be created for application, service sub-layer or forwarding sub-layer. A single profile may be shared by multiple applications/sub-layer. Each profile indicates the members that currently are using a profile.

Depending on which DetNet layers and functions are required, some or all of the components may be configured. Examples are shown in [Appendix B](#).

7. DetNet Configuration YANG Structures

The following is a partial tree representation of the YANG as defined in [\[RFC8340\]](#). This corresponds to the structure layout in the previous section.

```

module: ietf-detnet
  +--rw detnet
    +--rw traffic-profile* [name]
      | +--rw name string
      | +--rw traffic-requirements
      | +--rw traffic-spec
      | +--ro member-app* app-flow-ref
      | +--ro member-service* service-sub-layer-ref
      | +--ro member-fwd-sublayer* forwarding-sub-layer-ref
    +--rw app-flows
      | +--rw app-flow* [name]
      | | +--rw name string
      | | +--rw bidir-congruent? boolean
      | | +--ro outgoing-service? service-sub-layer-ref
      | | +--ro incoming-service? service-sub-layer-ref
      | | +--rw traffic-profile? traffic-profile-ref
      | | +--rw ingress
      | | | ...
      | | +--rw egress
      | | | ...
    +--rw service
      | +--rw sub-layer* [name]
      | | +--rw name string
      | | +--rw service-rank? uint8
      | | +--rw traffic-profile? traffic-profile-ref
      | | +--rw service-protection
      | | | ...
      | | +--rw operation? operation
      | | +--rw incoming
      | | | ...
      | | +--rw outgoing
      | | | ...
    +--rw forwarding
      | +--rw sub-layer* [name]
      | | +--rw name string
      | | +--rw traffic-profile? traffic-profile-ref
      | | +--rw operation? forwarding-operations
      | | +--rw incoming
      | | | ...
      | | +--rw outgoing
      | | | ...

```

8. DetNet Configuration YANG Model

This YANG model imports typedefs from [\[RFC6991\]](#), [\[RFC8519\]](#), [\[RFC8294\]](#), [\[RFC8343\]](#), and [\[IEEE8021Q\]](#). This YANG model also has the following references to RFCs that are not in the document text body [\[RFC0791\]](#), [\[RFC4303\]](#), [\[RFC8349\]](#), [\[RFC8960\]](#), [\[RFC8964\]](#), and [\[RFC8200\]](#).


```
<CODE BEGINS> file "ietf-detnet@2021-10-20.yang"

module ietf-detnet {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-detnet";
  prefix dnet;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991 - Common YANG Data Types.";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991 - Common YANG Data Types.";
  }
  import ietf-ethertypes {
    prefix ethertypes;
    reference
      "RFC 8519 - YANG Data Model for Network Access Control
        Lists (ACLs).";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294 - Common YANG Data Types for the Routing Area.";
  }
  import ietf-packet-fields {
    prefix packet-fields;
    reference
      "RFC 8519 - YANG Data Model for Network Access Control Lists
        (ACLs).";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343 - A YANG Data Model for Interface Management.";
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
    reference
      "IEEE 802.1Qcx-2020 - IEEE Standard for Local and Metropolitan
        Area Networks--Bridges and Bridged Networks Amendment 33: YANG
        Data Model for Connectivity Fault Management.";
  }
}

organization
  "IETF DetNet Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/detnet/>
  WG List: <mailto:detnet@ietf.org>

  Editor: Xuesong Geng
         <mailto:gengxuesong@huawei.com>

  Editor: Yeoncheol Ryoo
```

```
<mailto:dbduscjf@etri.re.kr>
Editor: Don Fedyk
       <mailto:dfedyk@labn.net>;
Editor: Reshad Rahman
       <mailto:reshad@yahoo.com>
Editor: Mach Chen
       <mailto:mach.chen@huawei.com>
Editor: Zhenqiang Li
       <mailto:lizhenqiang@chinamobile.com>";
description
  "This YANG module describes the parameters needed
  for DetNet flow configuration and flow status
  reporting. This YANG module conforms to the Network
  Management Datastore Architecture (NMDA).

  Copyright (c) 2021 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
  for full legal notices.";

revision 2021-11-18 {
  description
    "initial revision";
  reference
    "RFC XXXX: Deterministic Networking (DetNet) YANG Model";
}

identity app-status {
  description
    "Base identity from which all application-status
    status types are derived.";
  reference
    "RFC 9016 Section 5.8";
}

identity none {
  base app-status;
  description
    "This Application has no status. This type of status is
    expected when the configuration is incomplete.";
  reference
    "RFC 9016 Section 5.8";
}
```

```
identity ready {
  base app-status;
  description
    "Application ingress/egress ready.";
  reference
    "RFC 9016 Section 5.8";
}

identity failed {
  base app-status;
  description
    "Application ingres/egresss failed.";
  reference
    "RFC 9016 Section 5.8";
}

identity out-of-service {
  base app-status;
  description
    "Application Administratively blocked.";
  reference
    "RFC 9016 Section 5.8";
}

identity partial-failed {
  base app-status;
  description
    "This is an Application with one or more Egress ready, and one
    or more Egress failed. The DetNet flow can be used if the
    Ingress is Ready.";
  reference
    "RFC 9016 Section 5.8";
}

typedef app-flow-ref {
  type leafref {
    path "/dnet:detnet"
      + "/dnet:app-flows"
      + "/dnet:app-flow"
      + "/dnet:name";
  }
  description
    "This is an Application Reference.";
}

typedef service-sub-layer-ref {
  type leafref {
    path "/dnet:detnet"
      + "/dnet:service"
      + "/dnet:sub-layer"
      + "/dnet:name";
  }
  description
    "This is a Service sub-layer Reference.";
}

typedef forwarding-sub-layer-ref {
  type leafref {
```

```
    path "/dnet:detnet"
      + "/dnet:forwarding"
      + "/dnet:sub-layer"
      + "/dnet:name";
  }
  description
    "This is a Forwarding sub-layer Reference.";
}

typedef traffic-profile-ref {
  type leafref {
    path "/dnet:detnet"
      + "/dnet:traffic-profile"
      + "/dnet:name";
  }
  description
    "This is a Traffic Profile Reference.";
}

typedef ipsec-spi {
  type uint32 {
    range "1..max";
  }
  description
    "IPsec Security Parameters Index. A 32 bit value
     where 0 is reserved.";
  reference
    "IETF RFC 4303 Encapsulating Security Payload (ESP).";
}

typedef operation {
  type enumeration {
    enum initiation {
      description
        "This is an initiating service sub-layer encapsulation.";
    }
    enum termination {
      description
        "Operation for DetNet service sub-layer decapsulation.";
    }
    enum relay {
      description
        "Operation for DetNet service sub-layer swap.";
    }
    enum non-detnet {
      description
        "No operation for DetNet service sub-layer.";
    }
  }
  description
    "Operation type identifies the behavior for this service
     sub-layer. Operations are described as unidirectional
     but a service sub-layer may combine operation types.";
}

typedef forwarding-operations {
  type enumeration {
    enum impose-and-forward {
```

```
    description
      "This operation impose outgoing label(s) and forward to
      next-hop.";
    reference
      " A YANG Data Model for MPLS Base RFC 8960.";
  }
  enum pop-and-forward {
    description
      "This operation pops the incoming label and forwards to
      the next-hop.";
    reference
      " A YANG Data Model for MPLS Base RFC 8960.";
  }
  enum pop-impose-and-forward {
    description
      "This operation pops the incoming label, imposes one or
      more outgoing label(s) and forwards to the next-hop.";
    reference
      " A YANG Data Model for MPLS Base RFC 8960.";
  }
  enum swap-and-forward {
    description
      "This operation swaps incoming label, with an outgoing
      label and forwards to the next-hop.";
    reference
      " A YANG Data Model for MPLS Base RFC 8960.";
  }
  enum forward {
    description
      "This operation forward to next-hop.";
  }
  enum pop-and-lookup {
    description
      "This operation pops incoming label and performs a
      lookup.";
  }
}
description
  "MPLS operations types. This is an enum modeled after the
  MPLS enum. The enums are the same as A YANG Data Model
  for MPLS Base. RFC 8960.";
}

typedef service-protection {
  type enumeration {
    enum none {
      description
        "No service protection provided.";
    }
    enum replication {
      description
        "A Packet Replication Function (PRF) replicates DetNet
        flow packets and forwards them to one or more next hops in
        the DetNet domain. The number of packet copies sent to
        each next hop is a DetNet flow specific parameter at the
        node doing the replication. PRF can be implemented by an
        edge node, a relay node, or an end system.";
    }
  }
}
```

```
enum elimination {
  description
    "A Packet Elimination Function (PEF) eliminates duplicate
    copies of packets to prevent excess packets flooding the
    network or duplicate packets being sent out of the DetNet
    domain. PEF can be implemented by an edge node, a relay
    node, or an end system.";
}
enum ordering {
  description
    "A Packet Ordering Function (POF) re-orders packets within
    a DetNet flow that are received out of order. This
    function can be implemented by an edge node, a relay node,
    or an end system.";
}
enum elimination-ordering {
  description
    "A combination of PEF and POF that can be implemented by
    an edge node, a relay node, or an end system.";
}
enum elimination-replication {
  description
    "A combination of PEF and PRF that can be implemented by
    an edge node, a relay node, or an end system.";
}
enum elimination-ordering-replication {
  description
    "A combination of PEF, POF and PRF that can be implemented
    by an edge node, a relay node, or an end system.";
}
}
description
  "This typedef describes the service protection types.";
}

typedef sequence-number-generation {
  type enumeration {
    enum copy-from-app-flow {
      description
        "This type means copy the app-flow sequence number to the
        DetNet-flow.";
    }
    enum generate-by-detnet-flow {
      description
        "This type means generate the sequence number by the
        DetNet flow.";
    }
  }
  description
    "An enumeration for the sequence number behaviors supported.";
}

typedef sequence-number-field {
  type enumeration {
    enum zero-sn {
      description
        "No DetNet sequence number field is used.";
    }
  }
}
```

```
enum short-sn {
  value 16;
  description
    "A 16-bit DetNet sequence number field is used.";
}
enum long-sn {
  value 28;
  description
    "A 28-bit DetNet sequence number field is used.";
}
}
description
  "This type captures the sequence number behavior.";
}

grouping ip-header {
  description
    "This grouping captures the IPv4/IPv6 packet header
    information. it is modeled after existing fields.";
  leaf src-ip-address {
    type inet:ip-address-no-zone;
    description
      "The source IP address in the header.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf dest-ip-address {
    type inet:ip-address-no-zone;
    description
      "The destination IP address in the header.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf protocol-next-header {
    type uint8;
    description
      "Internet Protocol number. Refers to the protocol of the
      payload. In IPv6, this field is known as 'next-header',
      and if extension headers are present, the protocol is
      present in the 'upper-layer' header.";
    reference
      "RFC 791: Internet Protocol
      RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification.";
  }
  leaf dscp {
    type inet:dscp;
    description
      "The traffic class value in the header.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf flow-label {
    type inet:ipv6-flow-label;
    description
      "The flow label value of the header.IPV6 only.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
}
```

```
}
leaf source-port {
  type inet:port-number;
  description
    "The source port number.";
  reference
    "RFC 6991 Common YANG Data Types";
}
leaf destination-port {
  type inet:port-number;
  description
    "The destination port number.";
  reference
    "RFC 6991 Common YANG Data Types";
}
}

grouping l2-header {
  description
    "The Ethernet or TSN packet header information.";
  leaf source-mac-address {
    type yang:mac-address;
    description
      "The source MAC address value of the Ethernet header.";
  }
  leaf destination-mac-address {
    type yang:mac-address;
    description
      "The destination MAC address value of the Ethernet header.";
  }
  leaf ethertype {
    type ethertypes:ethertype;
    description
      "The Ethernet packet type value of the Ethernet header.";
  }
  leaf vlan-id {
    type dot1q-types:vlanid;
    description
      "The VLAN value of the Ethernet header.";
    reference
      "IEEE 802.1Qcx-2020.";
  }
  leaf pcp {
    type dot1q-types:priority-type;
    description
      "The priority value of the Ethernet header.";
    reference
      "IEEE 802.1Qcx-2020.";
  }
}

grouping destination-ip-port-id {
  description
    "The TCP/UDP port(source/destination) identification
    information.";
  container destination-port {
    uses packet-fields:port-range-or-operator;
    description

```



```
        "This grouping captures the destination port fields.";
    }
}

grouping source-ip-port-id {
  description
    "The TCP/UDP port(source/destination) identification
    information.";
  container source-port {
    uses packet-fields:port-range-or-operator;
    description
      "This grouping captures the source port fields.";
  }
}

grouping ip-flow-id {
  description
    "The IPv4/IPv6 packet header identification information.";
  leaf src-ip-prefix {
    type inet:ip-prefix;
    description
      "The source IP prefix.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf dest-ip-prefix {
    type inet:ip-prefix;
    description
      "The destination IP prefix.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf protocol-next-header {
    type uint8;
    description
      "Internet Protocol number. Refers to the protocol of the
      payload. In IPv6, this field is known as 'next-header', and
      if extension headers are present, the protocol is present in
      the 'upper-layer' header.";
    reference
      "RFC 791: Internet Protocol
      RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification.";
  }
  leaf dscp {
    type inet:dscp;
    description
      "The traffic class value in the header.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
  leaf flow-label {
    type inet:ipv6-flow-label;
    description
      "The flow label value of the header.";
    reference
      "RFC 6991 Common YANG Data Types";
  }
}
```

```
    uses source-ip-port-id;
    uses destination-ip-port-id;
    leaf ipsec-spi {
      type ipsec-spi;
      description
        "IPsec Security Parameters Index of the Security
        Association.";
      reference
        "IETF RFC 4303 Encapsulating Security Payload (ESP).";
    }
  }
}

grouping mpls-flow-id {
  description
    "The MPLS packet header identification information.";
  choice label-space {
    description
      "Designates the label space being used.";
    case context-label-space {
      uses rt-types:mpls-label-stack;
    }
    case platform-label-space {
      leaf label {
        type rt-types:mpls-label;
        description
          "This is the case for Platform label space.";
      }
    }
  }
}

grouping data-flow-spec {
  description
    "app-flow identification.";
  choice data-flow-type {
    description
      "The Application flow type choices.";
    container tsn-app-flow {
      uses l2-header;
      description
        "The L2 header for application.";
    }
    container ip-app-flow {
      uses ip-flow-id;
      description
        "The IP header for application.";
    }
    container mpls-app-flow {
      uses mpls-flow-id;
      description
        "The MPLS header for application.";
    }
  }
}

grouping detnet-flow-spec {
  description
    "detnet-flow identification.";
```

```
choice detnet-flow-type {
  description
    "The Detnet flow type choices.";
  case ip-detnet-flow {
    uses ip-flow-id;
  }
  case mpls-detnet-flow {
    uses mpls-flow-id;
  }
}

grouping app-flows-group {
  description
    "Incoming or outgoing app-flow reference group.";
  leaf-list flow {
    type app-flow-ref;
    description
      "List of ingress or egress app-flows.";
  }
}

grouping service-sub-layer-group {
  description
    "Incoming or outgoing service sub-layer reference group.";
  leaf-list sub-layer {
    type service-sub-layer-ref;
    description
      "List of incoming or outgoing service sub-layers that have
      to aggregate or disaggregate.";
  }
}

grouping forwarding-sub-layer-group {
  description
    "Incoming or outgoing forwarding sub-layer reference group.";
  leaf-list sub-layer {
    type forwarding-sub-layer-ref;
    description
      "List of incoming or outgoing forwarding sub-layers that
      have to aggregate or disaggregate.";
  }
}

grouping detnet-header {
  description
    "DetNet header info for DetNet encapsulation or swap.";
  choice header-type {
    description
      "The choice of DetNet header type.";
    case mpls {
      description
        "MPLS label stack for DetNet MPLS encapsulation or
        forwarding.";
      uses rt-types:mpls-label-stack;
    }
    case ip {
      description

```

```
        "IPv4/IPv6 packet header for DetNet IP encapsulation.";
        uses ip-header;
    }
}

grouping detnet-app-next-hop-content {
    description
        "Generic parameters of DetNet next hops. This follows the
        principles for next hops in RFC 8349";
    choice next-hop-options {
        mandatory true;
        description
            "Options for next hops. It is expected that further cases
            will be added through
            augments from other modules, e.g., for recursive
            next hops.";
        case simple-next-hop {
            description
                "This case represents a simple next hop consisting of the
                next-hop address and/or outgoing interface.";
            leaf outgoing-interface {
                type if:interface-ref;
                description
                    "The outgoing interface, when matching all flows to
                    the interface.";
            }
            choice flow-type {
                description
                    "The flow type choices.";
                case ip {
                    leaf next-hop-address {
                        type inet:ip-address-no-zone;
                        description
                            "The IP next hop case.";
                    }
                }
                case mpls {
                    uses rt-types:mpls-label-stack;
                    description
                        "The MPLS Label stack next hop case.";
                }
            }
        }
    }
    case next-hop-list {
        description
            "Container for multiple next hops.";
        list next-hop {
            key "hop-index";
            description
                "An entry in a next-hop list.";
            leaf hop-index {
                type uint8;
                description
                    "A user-specified identifier utilized to uniquely
                    reference the next-hop entry in the next-hop list.
                    The value of this index has no semantic meaning other
                    than for referencing the entry.";
            }
        }
    }
}
```

```
    }
    leaf outgoing-interface {
      type if:interface-ref;
      description
        "The outgoing interface, when matching all flows to
         the interface.";
    }
    choice flow-type {
      description
        "The flow types supported.";
      case ip {
        leaf next-hop-address {
          type inet:ip-address-no-zone;
          description
            "This is the IP flow type next hop.";
        }
      }
      case mpls {
        uses rt-types:mpls-label-stack;
      }
    }
  }
}
}

grouping detnet-forwarding-next-hop-content {
  description
    "Generic parameters of DetNet next hops. This follows the
     principles for next hops in RFC 8349";
  choice next-hop-options {
    mandatory true;
    description
      "Options for next hops.
       It is expected that further cases will be added through
       augments from other modules, e.g., for recursive
       next hops.";
    case simple-next-hop {
      description
        "This case represents a simple next hop consisting of the
         next-hop address and/or outgoing interface.";
      leaf outgoing-interface {
        type if:interface-ref;
        description
          "The outgoing interface, when matching all flows to
           the interface.";
      }
      choice flow-type {
        description
          "These are the flow type next hop choices.";
        case ip {
          choice operation-type {
            description
              "This is the IP forwarding operation choices.";
            case ip-forwarding {
              leaf next-hop-address {
                type inet:ip-address-no-zone;
                description

```

```
        "This is an IP address as a next hop.";
    }
    }
    case mpls-over-ip-encapsulation {
        uses ip-header;
    }
    }
    case mpls {
        uses rt-types:mpls-label-stack;
    }
}
case next-hop-list {
    description
    "Container for multiple next hops.";
    list next-hop {
        key "hop-index";
        description
        "An entry in a next-hop list.";
        leaf hop-index {
            type uint8;
            description
            "The value of the index for a hop.";
        }
        leaf outgoing-interface {
            type if:interface-ref;
            description
            "The outgoing interface, when matching all flows to
            the interface.";
        }
        choice flow-type {
            description
            "These are the flow type next hop choices.";
            case ip {
                choice operation-type {
                    description
                    "These are the next hop choices.";
                    case ip-forwarding {
                        leaf next-hop-address {
                            type inet:ip-address-no-zone;
                            description
                            "This is an IP address as a next hop.";
                        }
                    }
                    case mpls-over-ip-encapsulation {
                        uses ip-header;
                    }
                }
            }
            case mpls {
                uses rt-types:mpls-label-stack;
            }
        }
    }
}
}
```

```
container detnet {
  description
    "The top level DetNet container. This contains
    applications, service sub-layers and forwarding sub-layers
    as well as the traffic profiles.";
  list traffic-profile {
    key "name";
    description
      "A traffic profile.";
    leaf name {
      type string;
      description
        "An Aggregation group ID.";
    }
  }
  container traffic-requirements {
    description
      "This defines the attributes of the App-flow
      regarding bandwidth, latency, latency variation, loss, and
      misordering tolerance.";
    reference
      "RFC 9016 Section 4.2";
    leaf min-bandwidth {
      type uint64;
      units 'octets per second';
      description
        "This is the minimum bandwidth that has to be
        guaranteed for the DetNet service. MinBandwidth is
        specified in octets per second.";
      reference
        "RFC 9016 Section 4.2";
    }
    leaf max-latency {
      type uint32;
      units "nanoseconds";
      description
        "This is the maximum latency from Ingress to
        Egress(es) for a single packet of the DetNet flow.
        MaxLatency is specified as an integer number of
        nanoseconds. Any value above the MAX 4,294,967,295
        is displayed as MAX";
      reference
        "RFC 9016 Section 4.2";
    }
    leaf max-latency-variation {
      type uint32;
      units "nanoseconds";
      description
        "This is the difference between the
        minimum and the maximum end-to-end one-way latency.
        MaxLatencyVariation is specified as an integer number of
        nanoseconds.";
      reference
        "RFC 9016 Section 4.2";
    }
    leaf max-loss {
      type uint32;
      description
```

```
        "This defines the maximum Packet Loss Ratio (PLR)
        parameter for the DetNet service between the Ingress and
        Egress(es) of the DetNet domain.";
    reference
        "RFC 9016 Section 4.2";
}
leaf max-consecutive-loss-tolerance {
    type uint32;
    units "packets";
    description
        "Some applications have special loss requirement, such
        as MaxConsecutiveLossTolerance. The maximum consecutive
        loss tolerance parameter describes the maximum number of
        consecutive packets whose loss can be tolerated. The
        maximum consecutive loss tolerance can be measured for
        example based on sequence number.";
    reference
        "RFC 9016 Section 4.2";
}
leaf max-misordering {
    type uint32;
    units "packets";
    description
        "This describes the tolerable maximum number
        of packets that can be received out of order. The
        maximum allowed misordering can be measured for example
        based on sequence number. The value zero for the
        maximum allowed misordering indicates that in order
        delivery is required, misordering cannot be tolerated.";
    reference
        "RFC 9016 Section 4.2";
}
}
container traffic-spec {
    description
        "Traffic-specification specifies how the Source transmits
        packets for the flow. This is the promise/request of the
        Source to the network. The network uses this flow
        specification to allocate resources and adjust queue
        parameters in network nodes.";
    reference
        "RFC 9016 Section 5.5";
    leaf interval {
        type uint32;
        units "nanoseconds";
        description
            "The period of time in which the traffic
            specification should not be exceeded.";
        reference
            "RFC 9016 Section 5.5, IEEE802.1Q";
    }
    leaf max-pkts-per-interval {
        type uint32;
        description
            "The maximum number of packets that the
            source will transmit in one interval.";
        reference
            "RFC 9016 Section 5.5, IEEE802.1Q";
    }
}
```



```
    }
    leaf max-payload-size {
      type uint32;
      description
        "The maximum payload size that the source
         will transmit.";
      reference
        "RFC 9016 Section 5.5, IEEE802.1Q";
    }
    leaf min-payload-size {
      type uint32;
      description
        "The minimum payload size that the source
         will transmit., IEEE802.1Q";
    }
    leaf min-pkts-per-interval {
      type uint32;
      description
        "The minimum number of packets that the
         source will transmit in one interval.";
      reference
        "RFC 9016 Section 5.5, IEEE802.1Q";
    }
  }
  leaf-list member-app {
    type app-flow-ref;
    config false;
    description
      "A list of Applications attached to this profile. Each
       application that uses a profile has an automatically
       populated reference.";
    reference
      "RFC XXXX: Deterministic Networking (DetNet) YANG Model
       Section 5";
  }
  leaf-list member-service {
    type service-sub-layer-ref;
    config false;
    description
      "A list of Service Sub-layers attached to this profile.
       Each Service Sub-layers that uses a profile has an
       automatically populated reference.";
    reference
      "RFC XXXX: Deterministic Networking (DetNet) YANG Model
       Section 5";
  }
  leaf-list member-fwd-sublayer {
    type forwarding-sub-layer-ref;
    config false;
    description
      "A list of Forwarding Sub-layers attached to this profile.
       Each Forwarding Sub-layers that uses a profile has an
       automatically populated reference.";
    reference
      "RFC XXXX: Deterministic Networking (DetNet) YANG Model
       Section 5";
  }
}
```

```
container app-flows {
  description
    "The DetNet app-flow configuration.";
  reference
    "RFC 9016 Section 4.1";
  list app-flow {
    key "name";
    description
      "A unique (management) identifier of the App-flow.";
    leaf name {
      type string;
      description
        "A unique (management) identifier of the App-flow.";
      reference
        "RFC 9016
         Sections 4.1, 5.1";
    }
    leaf bidir-congruent {
      type boolean;
      default false;
      description
        "Defines the data path requirement of the App-flow
         whether it must share the same data path and physical
         path for both directions through the network, e.g., to
         provide congruent paths in the two directions.";
      reference
        "RFC 9016
         Section 4.2";
    }
    leaf outgoing-service {
      type service-sub-layer-ref;
      config false;
      description
        "Binding to this applications outgoing
         service.";
    }
    leaf incoming-service {
      type service-sub-layer-ref;
      config false;
      description
        "Binding to this applications incoming service.";
    }
    leaf traffic-profile {
      type traffic-profile-ref;
      description
        "The Traffic Profile for this group.";
    }
  }
  container ingress {
    description
      "Ingress DetNet application flows or a compound flow.";
    leaf app-flow-status {
      type identityref {
        base app-status;
      }
      default none;
      config false;
      description
        "Status of ingress application flow. This is an
```



```
    reference
      "RFC 9016 Section 5.7.";
  }
  leaf traffic-profile {
    type traffic-profile-ref;
    description
      "The Traffic Profile for this service.";
  }
  container service-protection {
    description
      "This is the service protection type and sequence number
      options.";
    leaf protection {
      type service-protection;
      description
        "The DetNet service protection type such as
        Packet Replication Function (PRF),
        Packet Elimination Function (PEF),
        Packet Replication, Elimination, and Ordering Functions
        (PREOF).";
      reference
        "RFC 8938 Section 4.3";
    }
    leaf sequence-number-length {
      type sequence-number-field;
      default zero-sn;
      description
        "Sequence number field length can be one of 0 (none),
        16-bits or 28-bits. The default is none.";
    }
  }
  leaf operation {
    type operation;
    description
      "This is the service operation type for this service
      sub-layer;";
  }
  container incoming {
    description
      "The DetNet service sub-layer incoming configuration.";
    choice incoming {
      mandatory true;
      description
        "A service sub-layer may have App flows or other
        service sub-layers.";
      container app-flow {
        description
          "This service sub-layer is related to the app-flows
          of the upper layer and provide ingress proxy or
          ingress aggregation at the ingress node.";
        uses app-flows-group;
      }
      container service-aggregation {
        description
          "This service sub-layer is related to the service
          sub-layer of the upper layer and provide
          service-to-service aggregation at the ingress node
          or relay node.";
      }
    }
  }
}
```



```
}
container service-sub-layer {
  description
    "This service sub-layer is sent to the service
    sub-layers of the lower layer for service-to-service
    aggregation at the ingress node or relay node. The
    service sub-layer encapsulates the DetNet
    Control-Word and S-label when the operation type is
    service-initiation, and swaps the S-label when the
    operation type is service-relay.";
  reference
    "RFC 8964 Section 4.2.1 and 4.2.2.";
  leaf aggregation-sub-layer {
    type service-sub-layer-ref;
    description
      "reference point of the service-sub-layer
      at which this service will be aggregated.";
  }
}
container service-label {
  description
    "This is the MPLS service sub-layer label. This
    is optional and only used when the service
    sublayer uses MPLS. It is an MPLS stack since
    more than a single label may be used.";
  uses rt-types:mpls-label-stack;
}
}
container app-flow {
  description
    "This service sub-layer is sent to the app-flow of
    the upper layer for egress proxy at the egress node,
    and decapsulates the DetNet Control-Word and S-label
    for individual DetNet service. This outgoing type
    only can be chosen when the operation type is
    service-termination.";
  reference
    "RFC 8964 Section 4.2.1 and 4.2.2.";
  uses app-flows-group;
}
container service-disaggregation {
  description
    "This service sub-layer is sent to the service
    sub-layer of the upper layer for service-to-service
    disaggregation at the relay node or egress node, and
    decapsulates the DetNet Control-Word and A-label for
    aggregated DetNet service. This outgoing type only
    can be chosen when the operation type is
    service-termination.";
  reference
    "RFC 8964 Section 4.2.1 and 4.2.2.";
  uses service-sub-layer-group;
}
container forwarding-disaggregation {
  description
    "This service sub-layer is sent to the forwarding
    sub-layer of the upper layer for
    forwarding-to-service disaggregation at the relay
    node or egress node, and decapsulates the DetNet
```



```
    }
    container forwarding-id {
      description
        "This forwarding sub-layer is related to all of the
        lower layer and provide DetNet forwarding swap or
        termination at the transit node or relay node or
        egress node.";
      leaf interface {
        type if:interface-ref;
        description
          "This is the interface associated with the
          forwarding sub-layer.";
      }
      uses detnet-flow-spec;
    }
  }
}
container outgoing {
  description
    "The DetNet forwarding sub-layer outbound
    configuration.";
  choice outgoing {
    mandatory true;
    description
      "This is when a service connected directly to an
      interface with no forwarding sub-layer.";
    container
      interface {
        description
          "This forwarding sub-layer is sent to the interface
          for send to next-hop at the ingress node or relay
          node or transit node.";
        uses detnet-forwarding-next-hop-content;
      }
    container service-aggregation {
      description
        "This forwarding sub-layer is sent to the service
        sub-layers of the lower layer for
        forwarding-to-service aggregation at the ingress
        node or relay node.";
      leaf aggregation-sub-layer {
        type service-sub-layer-ref;
        description
          "This is a reference to the service sub-layer.";
      }
    }
    container optional-forwarding-label {
      description
        "This is the optional forwarding label for service
        aggregation.";
      uses rt-types:mpls-label-stack;
    }
  }
}
container forwarding-sub-layer {
  description
    "This forwarding sub-layer is sent to the forwarding
    sub-layers of the lower layer for
    forwarding-to-forwarding aggregation at the ingress
    node or relay node or transit node.";
```


Name: ietf-detnet
Maintained by IANA: N
Namespace: urn:ietf:params:xml:ns:yang:ietf-detnet
Prefix: dnet
Reference: This RFC when published.

10. Security Considerations

Security considerations for DetNet are covered in the DetNet Architecture [RFC8655] and DetNet Security Considerations [RFC9055].

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols, such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can break or incorrectly connect DetNet flows. Since this is a configured Data Plane any changes that are not coordinated with all devices along the path the whole DetNet module is considered vulnerable and should have authorized access only.

Similarly, the data nodes in these YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data node and their sensitivity/vulnerability:

/detnet/app-flows: This controls the application details so it could be considered sensitive.

/detnet/traffic-profile/member-app: This links traffic profiles to applications, so this also could be considered more sensitive. The traffic profiles linked to service sub-layer and forwarding sub-layer are less sensitive.

/detnet/service/sub-layer/incoming/app-flow: This links applications to services.

/detnet/service/sub-layer/outgoing/app-flow: This links applications to services.

11. Contributors

The editors of this document wish to thank and acknowledge the following people who contributed substantially to the content of this document and should be considered coauthors:

Mach(Guoyi) Chen
Huawei Technologies
Email: mach.chen@huawei.com

12. Acknowledgments

The editors of this document would like to thank Lou Berger, Tom Petch and Xufeng Lui for their detailed comments.

13. References

13.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8960] Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", RFC 8960, DOI 10.17487/RFC8960, December 2020, <<https://www.rfc-editor.org/info/rfc8960>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.
- [RFC9016] Varga, B., Farkas, J., Cummings, R., Jiang, Y., and D. Fedyk, "Flow and Service Information Model for Deterministic Networking (DetNet)", RFC 9016, DOI 10.17487/RFC9016, March 2021, <<https://www.rfc-editor.org/info/rfc9016>>.
- [RFC9055] Grossman, E., Ed., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", RFC 9055, DOI 10.17487/RFC9055, June 2021, <<https://www.rfc-editor.org/info/rfc9055>>.

13.2. Informative References

- [IEEE8021Q] IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks", DOI 10.1109/IEEESTD.2018.8403927, IEEE 802.1Q-2018, July 2018, <<https://ieeexplore.ieee.org/document/8403927>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

Appendix A. DetNet Configuration YANG Tree

This is the full YANG tree as described in [\[RFC8340\]](#).

```

module: ietf-detnet
+--rw detnet
  +--rw traffic-profile* [name]
    +--rw name string
    +--rw traffic-requirements
      +--rw min-bandwidth? uint64
      +--rw max-latency? uint32
      +--rw max-latency-variation? uint32
      +--rw max-loss? uint32
      +--rw max-consecutive-loss-tolerance? uint32
      +--rw max-misordering? uint32
    +--rw traffic-spec
      +--rw interval? uint32
      +--rw max-pkts-per-interval? uint32
      +--rw max-payload-size? uint32
      +--rw min-payload-size? uint32
      +--rw min-pkts-per-interval? uint32
    +--ro member-app* app-flow-ref
    +--ro member-service* service-sub-layer-ref
    +--ro member-fwd-sublayer* forwarding-sub-layer-ref
  +--rw app-flows
    +--rw app-flow* [name]
      +--rw name string
      +--rw bidir-congruent? boolean
      +--ro outgoing-service? service-sub-layer-ref
      +--ro incoming-service? service-sub-layer-ref
      +--rw traffic-profile? traffic-profile-ref
      +--rw ingress
        +--ro app-flow-status? identityref
        +--rw interface if:interface-ref
        +--rw (data-flow-type)?
          +--:(tsn-app-flow)
            +--rw tsn-app-flow
              +--rw source-mac-address?
                | yang:mac-address
              +--rw destination-mac-address?
                | yang:mac-address
              +--rw ethertype?
                | ethertypes:ethertype
              +--rw vlan-id?
                | dot1q-types:vlanid
              +--rw pcp?
                | dot1q-types:priority-type
          +--:(ip-app-flow)
            +--rw ip-app-flow
              +--rw src-ip-prefix? inet:ip-prefix
              +--rw dest-ip-prefix? inet:ip-prefix
              +--rw protocol-next-header? uint8
              +--rw dscp? inet:dscp
              +--rw flow-label?
                | inet:ipv6-flow-label
            +--rw source-port
              +--rw (port-range-or-operator)?
                +--:(range)
                  +--rw lower-port
                    | inet:port-number
                  +--rw upper-port

```

```

|         inet:port-number
+---:(operator)
|         +--rw operator?      operator
|         +--rw port          inet:port-number
+---rw destination-port
|         +--rw (port-range-or-operator)?
|         +---:(range)
|         |         +--rw lower-port
|         |         |         inet:port-number
|         |         +--rw upper-port
|         |         |         inet:port-number
|         +---:(operator)
|         |         +--rw operator?      operator
|         |         +--rw port          inet:port-number
+---rw ipsec-spi?          ipsec-spi
+---:(mpls-app-flow)
+---rw mpls-app-flow
+---rw (label-space)?
+---:(context-label-space)
|         +--rw mpls-label-stack
|         |         +--rw entry* [id]
|         |         |         +--rw id          uint8
|         |         |         +--rw label?
|         |         |         |         rt-types:mpls-label
|         |         |         +--rw ttl?          uint8
|         |         |         +--rw traffic-class? uint8
+---:(platform-label-space)
|         +--rw label?
|         |         rt-types:mpls-label
+---rw egress
+---rw (application-type)?
+---:(ethernet)
|         +--rw ethernet
|         |         +--rw interface?  if:interface-ref
+---:(ip-mpls)
+---rw ip-mpls
+---rw (next-hop-options)
+---:(simple-next-hop)
|         +--rw outgoing-interface?
|         |         if:interface-ref
+---rw (flow-type)?
+---:(ip)
|         +--rw next-hop-address?
|         |         inet:ip-address-no-zone
+---:(mpls)
|         +--rw mpls-label-stack
|         |         +--rw entry* [id]
|         |         |         +--rw id          uint8
|         |         |         +--rw label?
|         |         |         |         rt-types:mpls-label
|         |         |         +--rw ttl?          uint8
|         |         |         +--rw traffic-class? uint8
+---:(next-hop-list)
+---rw next-hop* [hop-index]
|         +--rw hop-index          uint8
|         +--rw outgoing-interface?
|         |         if:interface-ref
+---rw (flow-type)?

```



```

+--rw service-sub-layer
  +--rw aggregation-sub-layer?
  |   service-sub-layer-ref
  +--rw service-label
    +--rw mpls-label-stack
      +--rw entry* [id]
        +--rw id          uint8
        +--rw label?
        |   rt-types:mpls-label
        +--rw ttl?       uint8
        +--rw traffic-class?  uint8
+--:(app-flow)
  +--rw app-flow
    +--rw flow*   app-flow-ref
+--:(service-disaggregation)
  +--rw service-disaggregation
    +--rw sub-layer*   service-sub-layer-ref
+--:(forwarding-disaggregation)
  +--rw forwarding-disaggregation
    +--rw sub-layer*   forwarding-sub-layer-ref
+--rw forwarding
  +--rw sub-layer* [name]
    +--rw name          string
    +--rw traffic-profile?  traffic-profile-ref
    +--rw operation?     forwarding-operations
    +--rw incoming
      +--rw (incoming)
        +--:(service-sub-layer)
          +--rw service-sub-layer
            +--rw sub-layer*   service-sub-layer-ref
        +--:(forwarding-aggregation)
          +--rw forwarding-aggregation
            +--rw sub-layer*   forwarding-sub-layer-ref
        +--:(forwarding-id)
          +--rw forwarding-id
            +--rw interface?
            |   if:interface-ref
            +--rw (detnet-flow-type)?
              +--:(ip-detnet-flow)
                +--rw src-ip-prefix?
                |   inet:ip-prefix
                +--rw dest-ip-prefix?
                |   inet:ip-prefix
                +--rw protocol-next-header?  uint8
                +--rw dscp?                  inet:dscp
                +--rw flow-label?
                |   inet:ipv6-flow-label
                +--rw source-port
                |   +--rw (port-range-or-operator)?
                |   |   +--:(range)
                |   |   |   +--rw lower-port
                |   |   |   |   inet:port-number
                |   |   |   +--rw upper-port
                |   |   |   |   inet:port-number
                |   |   +--:(operator)
                |   |   +--rw operator?   operator
                |   +--rw port
                |   |   inet:port-number

```

```

+--rw destination-port
|   +--rw (port-range-or-operator)?
|   |   +--:(range)
|   |   |   +--rw lower-port
|   |   |   |   inet:port-number
|   |   |   +--rw upper-port
|   |   |   |   inet:port-number
|   |   +--:(operator)
|   |   +--rw operator?      operator
|   |   +--rw port
|   |   |   inet:port-number
+--rw ipsec-spi?              ipsec-spi
+--:(mpls-detnet-flow)
+--rw (label-space)?
+--:(context-label-space)
|   +--rw mpls-label-stack
|   |   +--rw entry* [id]
|   |   |   +--rw id              uint8
|   |   |   +--rw label?
|   |   |   |   rt-types:mpls-label
|   |   |   +--rw ttl?          uint8
|   |   |   +--rw traffic-class? uint8
+--:(platform-label-space)
+--rw label?
|   rt-types:mpls-label
+--rw outgoing
+--rw (outgoing)
+--:(interface)
|   +--rw interface
|   |   +--rw (next-hop-options)
|   |   +--:(simple-next-hop)
|   |   |   +--rw outgoing-interface?
|   |   |   |   if:interface-ref
|   |   +--rw (flow-type)?
|   |   +--:(ip)
|   |   |   +--rw (operation-type)?
|   |   |   +--:(ip-forwarding)
|   |   |   |   +--rw next-hop-address?
|   |   |   |   |   inet:ip-address-no-zone
|   |   |   +--:(mpls-over-ip-encapsulation)
|   |   |   +--rw src-ip-address?
|   |   |   |   inet:ip-address-no-zone
|   |   |   +--rw dest-ip-address?
|   |   |   |   inet:ip-address-no-zone
|   |   |   +--rw protocol-next-header?
|   |   |   |   uint8
|   |   |   +--rw dscp?
|   |   |   |   inet:dscp
|   |   |   +--rw flow-label?
|   |   |   |   inet:ipv6-flow-label
|   |   |   +--rw source-port?
|   |   |   |   inet:port-number
|   |   |   +--rw destination-port?
|   |   |   |   inet:port-number
+--:(mpls)
+--rw mpls-label-stack
+--rw entry* [id]

```

```

+---rw id uint8
+---rw label?
|   rt-types:mpls-label
+---rw ttl? uint8
+---rw traffic-class? uint8
+---:(next-hop-list)
+---rw next-hop* [hop-index]
+---rw hop-index
|   uint8
+---rw outgoing-interface?
|   if:interface-ref
+---rw (flow-type)?
+---:(ip)
+---rw (operation-type)?
+---:(ip-forwarding)
|   +---rw next-hop-address?
|       inet:ip-address-no-
|       zone
+---:(mpls-over-ip-
encapsulation)
+---rw src-ip-address?
|   inet:ip-address-no-
|   zone
+---rw dest-ip-address?
|   inet:ip-address-no-
|   zone
+---rw protocol-next-
header?
|   uint8
+---rw dscp?
|   inet:dscp
+---rw flow-label?
|   inet:ipv6-flow-
|   label
+---rw source-port?
|   inet:port-number
+---rw destination-port?
|   inet:port-number
+---:(mpls)
+---rw mpls-label-stack
+---rw entry* [id]
+---rw id
|   uint8
+---rw label?
|   rt-types:mpls-
|   label
+---rw ttl?
|   uint8
+---rw traffic-class?
|   uint8
+---:(service-aggregation)
+---rw service-aggregation
+---rw aggregation-sub-layer?
|   service-sub-layer-ref
+---rw optional-forwarding-label
+---rw mpls-label-stack
+---rw entry* [id]
+---rw id uint8

```

```

    +-rw label?
    |   rt-types:mpls-label
    +-rw ttl?      uint8
    +-rw traffic-class?  uint8
+--:(forwarding-sub-layer)
  +-rw forwarding-sub-layer
  +-rw aggregation-sub-layer?
  |   forwarding-sub-layer-ref
  +-rw forwarding-label
  +-rw mpls-label-stack
  +-rw entry* [id]
  +-rw id      uint8
  +-rw label?
  |   rt-types:mpls-label
  +-rw ttl?      uint8
  +-rw traffic-class?  uint8
+--:(service-sub-layer)
  +-rw service-sub-layer
  +-rw sub-layer*  service-sub-layer-ref
+--:(forwarding-disaggregation)
  +-rw forwarding-disaggregation
  +-rw sub-layer*  forwarding-sub-layer-ref

```

Appendix B. Examples

The following examples are provided. These examples are tested with Yanglint and use operational output to exercise both config true and config false objects. Note that IPv4 and IPv6 addresses are supported but for clarity in the examples and diagrams IPv4 has been used in most examples. The IP types are imported from [\[RFC6991\]](#) and these support both IPv4 and IPv6.

The following are examples of aggregation and disaggregation at various points in Detnet. Figures are provided in the PDF version of this document.

B.1. Example A-1 JSON Configuration/Operational

This illustrates simple aggregation. Ingress node 1 aggregates App flows 0 and 1 into a service sub-layer of DetNet flow 1. Two ways of illustrating this follow, then the JSON operational data model corresponding to the diagrams follows. This example uses IPv6 address format.

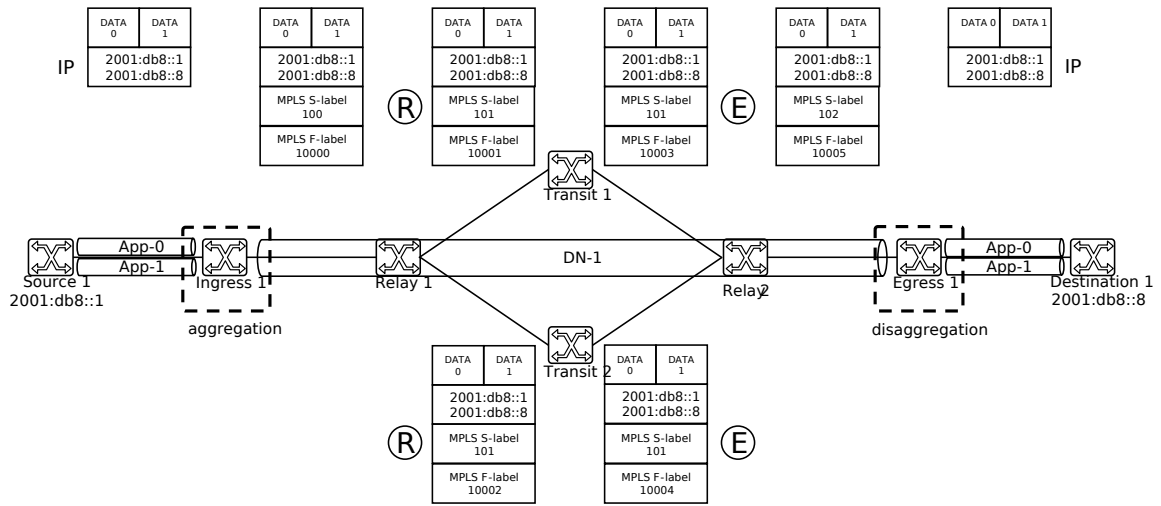


Figure 2: Case A-1 Example JSON Operational/Configuration

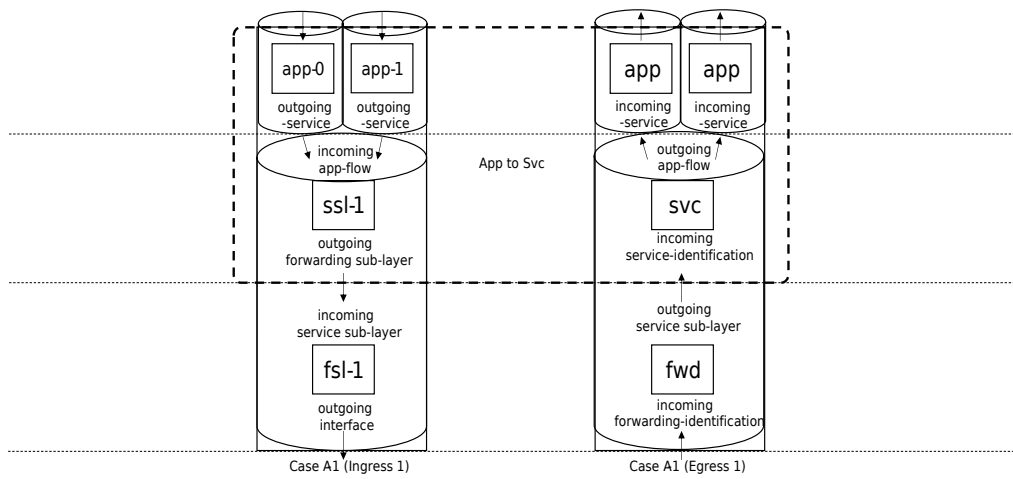


Figure 3: Case A-1 Example JSON Operational/Configuration


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 200000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500,
          "min-payload-size": 100,
          "min-pkts-per-interval": 1
        },
        "member-app": [
          "app-0",
          "app-1"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 200000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500,
          "min-payload-size": 100,
          "min-pkts-per-interval": 1
        },
        "member-service": [
          "ssl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500
        },
        "member-fw-sublayer": [
          "fsl-1"
        ]
      }
    ]
  }
}
```



```
],
"app-flows": {
  "app-flow": [
    {
      "name": "app-0",
      "bidir-congruent": false,
      "outgoing-service": "ssl-1",
      "traffic-profile": "pf-1",
      "ingress": {
        "app-flow-status": "ietf-detnet:ready",
        "interface": "eth0",
        "ip-app-flow": {
          "src-ip-prefix": "2001:db8::1/128",
          "dest-ip-prefix": "2001:db8::8/128",
          "dscp": 6
        }
      }
    }
  ],
  {
    "name": "app-1",
    "bidir-congruent": false,
    "outgoing-service": "ssl-1",
    "traffic-profile": "pf-1",
    "ingress": {
      "app-flow-status": "ietf-detnet:ready",
      "interface": "eth0",
      "ip-app-flow": {
        "src-ip-prefix": "2001:db8::1/128",
        "dest-ip-prefix": "2001:db8::8/128",
        "dscp": 7
      }
    }
  }
]
},
"service": {
  "sub-layer": [
    {
      "name": "ssl-1",
      "service-rank": 10,
      "traffic-profile": "pf-2",
      "service-protection": {
        "protection": "none",
        "sequence-number-length": "long-sn"
      },
      "operation": "initiation",
      "incoming": {
        "app-flow": {
          "flow": [
            "app-0",
            "app-1"
          ]
        }
      }
    }
  ],
  "outgoing": {
    "forwarding-sub-layer": {
      "service-outgoing": [
        {

```

```
        "index": 0,
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "100"
            }
          ]
        },
        "sub-layer": [
          "fsl-1"
        ]
      ]
    }
  ]
},
"forwarding": {
  "sub-layer": [
    {
      "name": "fsl-1",
      "traffic-profile": "pf-3",
      "operation": "impose-and-forward",
      "incoming": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-1"
          ]
        }
      },
      "outgoing": {
        "interface": {
          "outgoing-interface": "eth2",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "10000"
              }
            ]
          }
        }
      }
    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
```

```
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
```

Figure 4: Example A-1 DetNet JSON configuration

B.2. Example B-1 XML Config: Aggregation using a Forwarding Sub-layer

This illustrates aggregation in the service sub-layers of DetNet. Flows 1 and 2 are aggregated into a forwarding sub-layer. A diagram illustrating this case is shown and then the corresponding XML operational data follows.

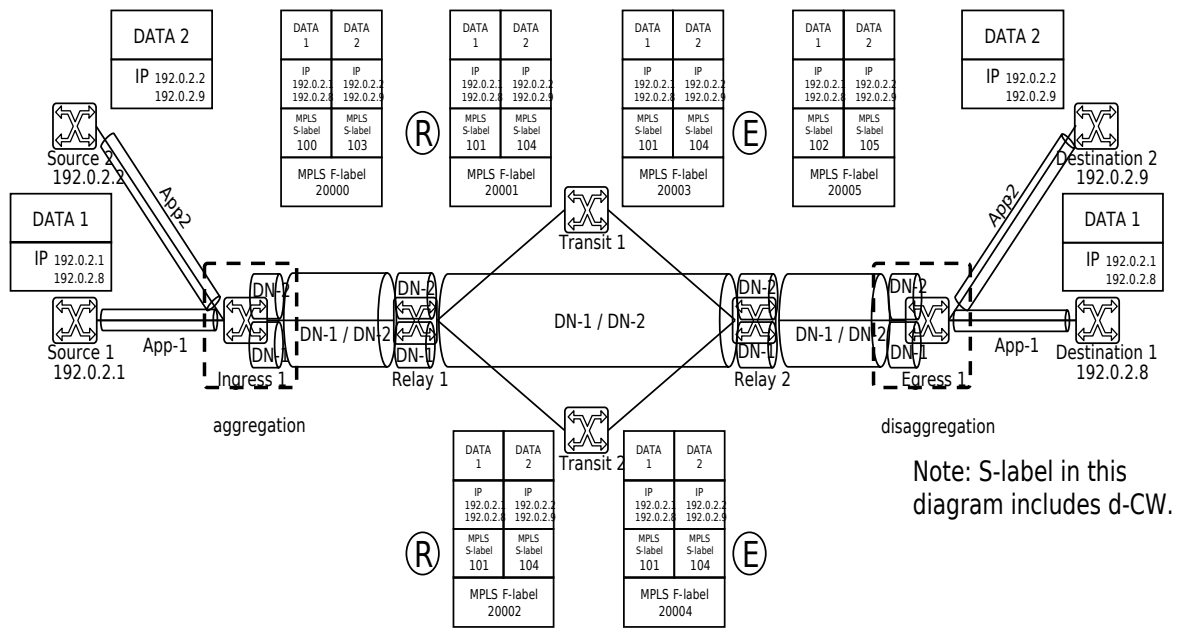


Figure 5: Case B-1 Example XML Config: Aggregation using a Forwarding Sub-layer


```
<interfaces
  xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>eth0</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time
        >2020-12-18T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth1</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time
        >2020-12-18T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth2</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time
        >2020-12-18T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth3</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time
        >2020-12-18T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth4</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time
        >2020-12-18T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
</interfaces>
<detnet
  xmlns="urn:ietf:params:xml:ns:yang:ietf-detnet">
  <app-flows>
    <app-flow>
      <name>app-1</name>
      <bidir-congruent>>false</bidir-congruent>
      <outgoing-service>ssl-1</outgoing-service>
      <traffic-profile>1</traffic-profile>
    </app-flow>
  </app-flows>
</detnet>
```

```
<ingress>
  <app-flow-status>ready</app-flow-status>
  <interface>eth0</interface>
  <ip-app-flow>
    <src-ip-prefix>192.0.2.1/32</src-ip-prefix>
    <dest-ip-prefix>192.0.2.8/32</dest-ip-prefix>
    <dscp>6</dscp>
  </ip-app-flow>
</ingress>
</app-flow>
<app-flow>
  <name>app-2</name>
  <bidir-congruent>false</bidir-congruent>
  <outgoing-service>ssl-2</outgoing-service>
  <traffic-profile>1</traffic-profile>
  <ingress>
    <app-flow-status>ready</app-flow-status>
    <interface>eth1</interface>
    <ip-app-flow>
      <src-ip-prefix>192.0.2.2/32</src-ip-prefix>
      <dest-ip-prefix>192.0.2.9/32</dest-ip-prefix>
      <dscp>7</dscp>
    </ip-app-flow>
  </ingress>
</app-flow>
</app-flows>
<traffic-profile>
  <name>1</name>
  <traffic-requirements>
    <min-bandwidth>1000000000</min-bandwidth>
    <max-latency>1000000000</max-latency>
    <max-latency-variation>2000000000</max-latency-variation>
    <max-loss>2</max-loss>
    <max-consecutive-loss-tolerance
      >5</max-consecutive-loss-tolerance>
    <max-misordering>0</max-misordering>
  </traffic-requirements>
  <member-app>app-1</member-app>
  <member-app>app-2</member-app>
</traffic-profile>
<traffic-profile>
  <name>2</name>
  <traffic-requirements>
    <min-bandwidth>1000000000</min-bandwidth>
    <max-latency>1000000000</max-latency>
    <max-latency-variation>2000000000</max-latency-variation>
    <max-loss>2</max-loss>
    <max-consecutive-loss-tolerance
      >5</max-consecutive-loss-tolerance>
    <max-misordering>0</max-misordering>
  </traffic-requirements>
  <member-service>ssl-1</member-service>
  <member-service>ssl-2</member-service>
</traffic-profile>
<traffic-profile>
  <name>3</name>
  <traffic-spec>
    <interval>5</interval>
```

```
<max-pkts-per-interval>10</max-pkts-per-interval>
<max-payload-size>1500</max-payload-size>
</traffic-spec>
<member-fwd-sublayer>afl-1</member-fwd-sublayer>
</traffic-profile>
<service>
  <sub-layer>
    <name>ssl-1</name>
    <service-rank>10</service-rank>
    <traffic-profile>2</traffic-profile>
    <operation>initiation</operation>
    <service-protection>
      <protection>none</protection>
      <sequence-number-length>long-sn</sequence-number-length>
    </service-protection>
    <incoming>
      <app-flow>
        <flow>app-1</flow>
      </app-flow>
    </incoming>
    <outgoing>
      <forwarding-sub-layer>
        <service-outgoing>
          <index>0</index>
          <mpls-label-stack>
            <entry>
              <id>0</id>
              <label>100</label>
            </entry>
          </mpls-label-stack>
          <sub-layer>afl-1</sub-layer>
        </service-outgoing>
      </forwarding-sub-layer>
    </outgoing>
  </sub-layer>
  <sub-layer>
    <name>ssl-2</name>
    <service-rank>10</service-rank>
    <traffic-profile>2</traffic-profile>
    <operation>initiation</operation>
    <service-protection>
      <protection>none</protection>
      <sequence-number-length>long-sn</sequence-number-length>
    </service-protection>
    <incoming>
      <app-flow>
        <flow>app-2</flow>
      </app-flow>
    </incoming>
    <outgoing>
      <forwarding-sub-layer>
        <service-outgoing>
          <index>0</index>
          <mpls-label-stack>
            <entry>
              <id>0</id>
              <label>103</label>
            </entry>
          </mpls-label-stack>
        </service-outgoing>
      </forwarding-sub-layer>
    </outgoing>
  </sub-layer>
</service>
```



```
        </mpls-label-stack>
        <sub-layer>afl-1</sub-layer>
      </service-outgoing>
    </forwarding-sub-layer>
  </outgoing>
</sub-layer>
</service>
<forwarding>
<sub-layer>
  <name>afl-1</name>
  <traffic-profile>3</traffic-profile>
  <operation>impose-and-forward</operation>
  <incoming>
    <service-sub-layer>
      <sub-layer>ssl-1</sub-layer>
      <sub-layer>ssl-2</sub-layer>
    </service-sub-layer>
  </incoming>
  <outgoing>
    <interface>
      <outgoing-interface>eth2</outgoing-interface>
      <mpls-label-stack>
        <entry>
          <id>0</id>
          <label>10000</label>
        </entry>
      </mpls-label-stack>
    </interface>
  </outgoing>
</sub-layer>
</forwarding>
</detnet>
```

Figure 6: Example B-1 DetNet XML configuration

B.3. Example B-2 JSON Service Aggregation Configuration

This illustrates the service sub-layers of DetNet. Flows 1 and 2 are aggregated into a service sub-layer of aggregated DetNet flow 1. A diagram illustrating this case is shown and then the corresponding JSON operational data follows.

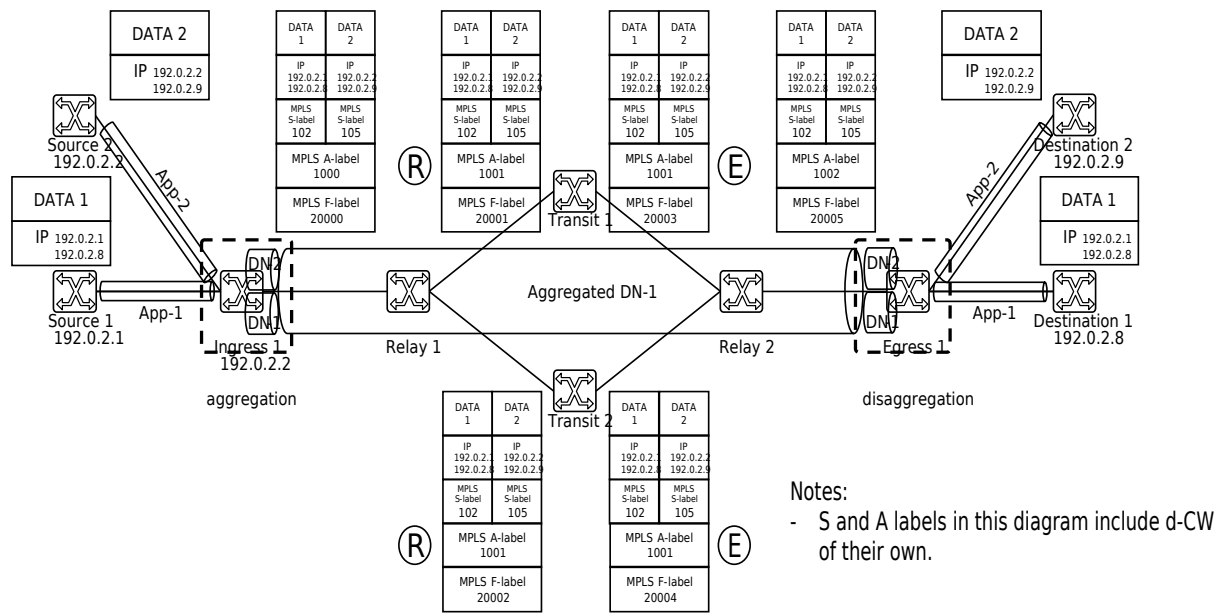


Figure 7: Case B-2 Example JSON Service Aggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 200000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-app": [
          "app-1",
          "app-2"
        ]
      },
      {
        "name": "2",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 200000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "3",
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500
        },
        "member-fwd-sublayer": [
          "afl-1"
        ]
      }
    ],
    "app-flows": {
      "app-flow": [
        {
          "name": "app-1",
          "bidir-congruent": false,
          "outgoing-service": "ssl-1",
          "traffic-profile": "1",
          "ingress": {
            "app-flow-status": "ietf-detnet:ready",
            "interface": "eth0",
            "ip-app-flow": {
              "src-ip-prefix": "192.0.2.1/32",

```

```
        "dest-ip-prefix": "192.0.2.8/32",
        "dscp": 6
    }
}
},
{
    "name": "app-2",
    "bidir-congruent": false,
    "outgoing-service": "ssl-2",
    "traffic-profile": "1",
    "ingress": {
        "app-flow-status": "ietf-detnet:ready",
        "interface": "eth0",
        "ip-app-flow": {
            "src-ip-prefix": "192.0.2.2/32",
            "dest-ip-prefix": "192.0.2.9/32",
            "dscp": 7
        }
    }
}
]
},
"service": {
    "sub-layer": [
        {
            "name": "ssl-1",
            "service-rank": 10,
            "traffic-profile": "2",
            "service-protection": {
                "protection": "none",
                "sequence-number-length": "long-sn"
            },
            "operation": "initiation",
            "incoming": {
                "app-flow": {
                    "flow": [
                        "app-1"
                    ]
                }
            },
            "outgoing": {
                "service-sub-layer": {
                    "aggregation-sub-layer": "asl-1",
                    "service-label": {
                        "mpls-label-stack": {
                            "entry": [
                                {
                                    "id": 0,
                                    "label": "102"
                                }
                            ]
                        }
                    }
                }
            }
        }
    ]
},
{
    "name": "ssl-2",
```

```
"service-rank": 10,
"traffic-profile": "2",
"service-protection": {
  "protection": "none",
  "sequence-number-length": "long-sn"
},
"operation": "initiation",
"incoming": {
  "app-flow": {
    "flow": [
      "app-2"
    ]
  }
},
"outgoing": {
  "service-sub-layer": {
    "aggregation-sub-layer": "asl-1",
    "service-label": {
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "105"
          }
        ]
      }
    }
  }
},
},
{
  "name": "asl-1",
  "service-rank": 10,
  "traffic-profile": "2",
  "service-protection": {
    "protection": "none",
    "sequence-number-length": "long-sn"
  },
  "operation": "initiation",
  "incoming": {
    "service-aggregation": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "service-outgoing": [
        {
          "index": 0,
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "1000"
              }
            ]
          }
        }
      ]
    }
  }
}
```

```
    ]
    },
    "sub-layer": [
      "afl-1"
    ]
  ]
}
}
}
}
}
}
},
"forwarding": {
  "sub-layer": [
    {
      "name": "afl-1",
      "traffic-profile": "3",
      "operation": "impose-and-forward",
      "incoming": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      },
      "outgoing": {
        "interface": {
          "outgoing-interface": "eth2",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20000"
              }
            ]
          }
        }
      }
    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-10-02T19:59:00-04:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-10-02T19:59:00-04:00"
      }
    }
  ]
}
```

```
    }
  },
  {
    "name": "eth2",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-10-02T19:59:00-04:00"
    }
  },
  {
    "name": "eth3",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-10-02T19:59:00-04:00"
    }
  },
  {
    "name": "eth4",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-10-02T19:59:00-04:00"
    }
  }
]
}
```

Figure 8: Example B-2 DetNet JSON Service Aggregation

B.4. Example C-1 JSON Relay Aggregation/Disaggregation Configuration

This illustrates the Relay node 1 aggregating the forwarding sub-layers of DetNet flows 1 and 2 into a forwarding sub-layer. A diagram illustrating both aggregation and disaggregation is shown and then the corresponding JSON operational data follows.

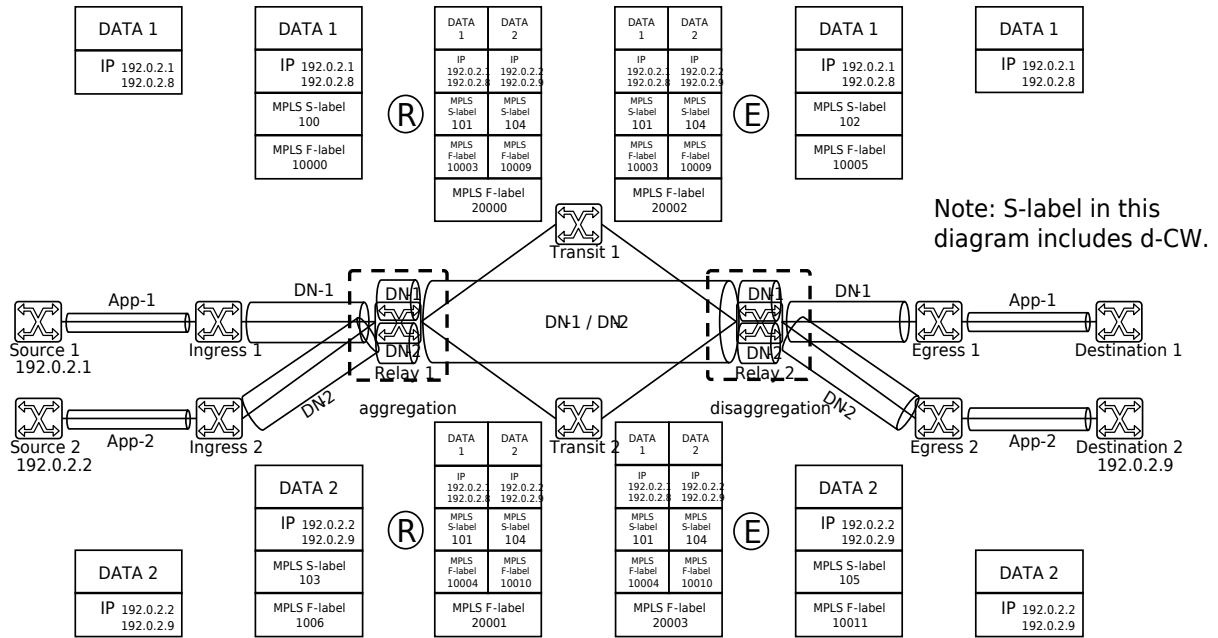


Figure 9: Case C-1 Example JSON Service Aggregation/Disaggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1",
          "afl-2"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2",
          "fsl-3",
          "fsl-4",
          "fsl-5",
          "fsl-6"
        ]
      }
    ],
    "service": {
      "sub-layer": [
        {
          "name": "ssl-1",
          "service-rank": 10,
          "traffic-profile": "pf-1",
          "service-protection": {
            "protection": "replication",
            "sequence-number-length": "long-sn"
          }
        }
      ]
    }
  }
}
```

```
"operation": "relay",
"incoming": {
  "service-id": {
    "mpls-label-stack": {
      "entry": [
        {
          "id": 0,
          "label": "100"
        }
      ]
    }
  }
},
"outgoing": {
  "forwarding-sub-layer": {
    "service-outgoing": [
      {
        "index": 0,
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "101"
            }
          ]
        },
        "sub-layer": [
          "fsl-2",
          "fsl-3"
        ]
      }
    ]
  }
},
{
  "name": "ssl-2",
  "service-rank": 10,
  "traffic-profile": "pf-1",
  "service-protection": {
    "protection": "replication",
    "sequence-number-length": "long-sn"
  },
  "operation": "relay",
  "incoming": {
    "service-id": {
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "103"
          }
        ]
      }
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
```

```
        "service-outgoing": [
          {
            "index": 0,
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "104"
                }
              ]
            },
            "sub-layer": [
              "fsl-5",
              "fsl-6"
            ]
          }
        ]
      },
    ],
    "forwarding": {
      "sub-layer": [
        {
          "name": "fsl-1",
          "traffic-profile": "pf-3",
          "operation": "pop-and-lookup",
          "incoming": {
            "forwarding-id": {
              "interface": "eth0",
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "10000"
                  }
                ]
              }
            }
          },
          "outgoing": {
            "service-sub-layer": {
              "sub-layer": [
                "ssl-1"
              ]
            }
          }
        }
      ],
      {
        "name": "fsl-2",
        "traffic-profile": "pf-3",
        "operation": "impose-and-forward",
        "incoming": {
          "service-sub-layer": {
            "sub-layer": [
              "ssl-1"
            ]
          }
        }
      }
    ]
  }
}
```

```
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "aggregation-sub-layer": "afl-1",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10003"
            }
          ]
        }
      }
    }
  }
},
{
  "name": "fsl-3",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1"
      ]
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "aggregation-sub-layer": "afl-2",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10004"
            }
          ]
        }
      }
    }
  }
},
{
  "name": "fsl-4",
  "traffic-profile": "pf-3",
  "operation": "pop-and-lookup",
  "incoming": {
    "forwarding-id": {
      "interface": "eth1",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10006"
          }
        ]
      }
    }
  }
}
```

```
    ]
  }
}
},
"outgoing": {
  "service-sub-layer": {
    "sub-layer": [
      "ssl-2"
    ]
  }
}
},
{
  "name": "fsl-5",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "aggregation-sub-layer": "afl-1",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10009"
            }
          ]
        }
      }
    }
  }
}
},
{
  "name": "fsl-6",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "aggregation-sub-layer": "afl-2",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
```

```
        "label": "10010"
      }
    ]
  }
}
},
{
  "name": "afl-1",
  "traffic-profile": "pf-2",
  "operation": "impose-and-forward",
  "incoming": {
    "forwarding-aggregation": {
      "sub-layer": [
        "fsl-2",
        "fsl-5"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth2",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20000"
          }
        ]
      }
    }
  }
},
{
  "name": "afl-2",
  "traffic-profile": "pf-2",
  "operation": "impose-and-forward",
  "incoming": {
    "forwarding-aggregation": {
      "sub-layer": [
        "fsl-3",
        "fsl-6"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20001"
          }
        ]
      }
    }
  }
}
```



```
    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 10: Example C-1 DetNet JSON Relay Service Aggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1",
          "afl-2"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2",
          "fsl-3",
          "fsl-4",
          "fsl-5",
          "fsl-6"
        ]
      }
    ],
    "service": {
      "sub-layer": [
        {
          "name": "ssl-1",
          "service-rank": 10,
          "traffic-profile": "pf-1",
          "service-protection": {
            "protection": "elimination",
            "sequence-number-length": "long-sn"
          }
        }
      ]
    }
  }
}
```

```
"operation": "relay",
"incoming": {
  "service-id": {
    "mpls-label-stack": {
      "entry": [
        {
          "id": 0,
          "label": "101"
        }
      ]
    }
  }
},
"outgoing": {
  "forwarding-sub-layer": {
    "service-outgoing": [
      {
        "index": 0,
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "102"
            }
          ]
        }
      ],
      "sub-layer": [
        "fsl-3"
      ]
    ]
  }
},
{
  "name": "ssl-2",
  "service-rank": 10,
  "traffic-profile": "pf-1",
  "service-protection": {
    "protection": "elimination",
    "sequence-number-length": "long-sn"
  },
  "operation": "relay",
  "incoming": {
    "service-id": {
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "104"
          }
        ]
      }
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "service-outgoing": [
```

```
    {
      "index": 0,
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "105"
          }
        ]
      },
      "sub-layer": [
        "fsl-6"
      ]
    }
  ]
},
"forwarding": {
  "sub-layer": [
    {
      "name": "afl-1",
      "traffic-profile": "pf-2",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20002"
              }
            ]
          }
        }
      },
      "outgoing": {
        "forwarding-disaggregation": {
          "sub-layer": [
            "fsl-1",
            "fsl-4"
          ]
        }
      }
    }
  ],
  {
    "name": "afl-2",
    "traffic-profile": "pf-2",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
```

```
        "id": 0,
        "label": "20003"
      }
    ]
  }
},
"outgoing": {
  "forwarding-disaggregation": {
    "sub-layer": [
      "fsl-2",
      "fsl-5"
    ]
  }
},
{
  "name": "fsl-1",
  "traffic-profile": "pf-3",
  "operation": "pop-and-lookup",
  "incoming": {
    "forwarding-id": {
      "interface": "eth0",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10003"
          }
        ]
      }
    }
  },
  "outgoing": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1"
      ]
    }
  }
},
{
  "name": "fsl-2",
  "traffic-profile": "pf-3",
  "operation": "pop-and-lookup",
  "incoming": {
    "forwarding-id": {
      "interface": "eth1",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10004"
          }
        ]
      }
    }
  },
}
```

```
    "outgoing": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-1"
        ]
      }
    },
  },
  {
    "name": "fsl-3",
    "traffic-profile": "pf-3",
    "operation": "impose-and-forward",
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-1"
        ]
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth2",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10005"
            }
          ]
        }
      }
    }
  },
  {
    "name": "fsl-4",
    "traffic-profile": "pf-3",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth0",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10009"
            }
          ]
        }
      }
    },
    "outgoing": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-2"
        ]
      }
    }
  },
},
```

```
{
  "name": "fsl-5",
  "traffic-profile": "pf-3",
  "operation": "pop-and-lookup",
  "incoming": {
    "forwarding-id": {
      "interface": "eth1",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10010"
          }
        ]
      }
    }
  },
  "outgoing": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  }
},
{
  "name": "fsl-6",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10011"
          }
        ]
      }
    }
  }
}
]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
```



```
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-12-18T18:59:00-05:00"
    }
  },
  {
    "name": "eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-12-18T18:59:00-05:00"
    }
  },
  {
    "name": "eth2",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-12-18T18:59:00-05:00"
    }
  },
  {
    "name": "eth3",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-12-18T18:59:00-05:00"
    }
  },
  {
    "name": "eth4",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2020-12-18T18:59:00-05:00"
    }
  }
]
}
```

Figure 11: Example C-1 DetNet JSON Relay Service Disaggregation

B.5. Example C-2 JSON Relay Aggregation Service Sub-Layer

This illustrates the Relay node 1 aggregating the service sub-layers of DetNet flows 1 and 2 into a forwarding sub-layer A diagram illustrating both aggregation and disaggregation is shown and then the corresponding JSON operational data follows.

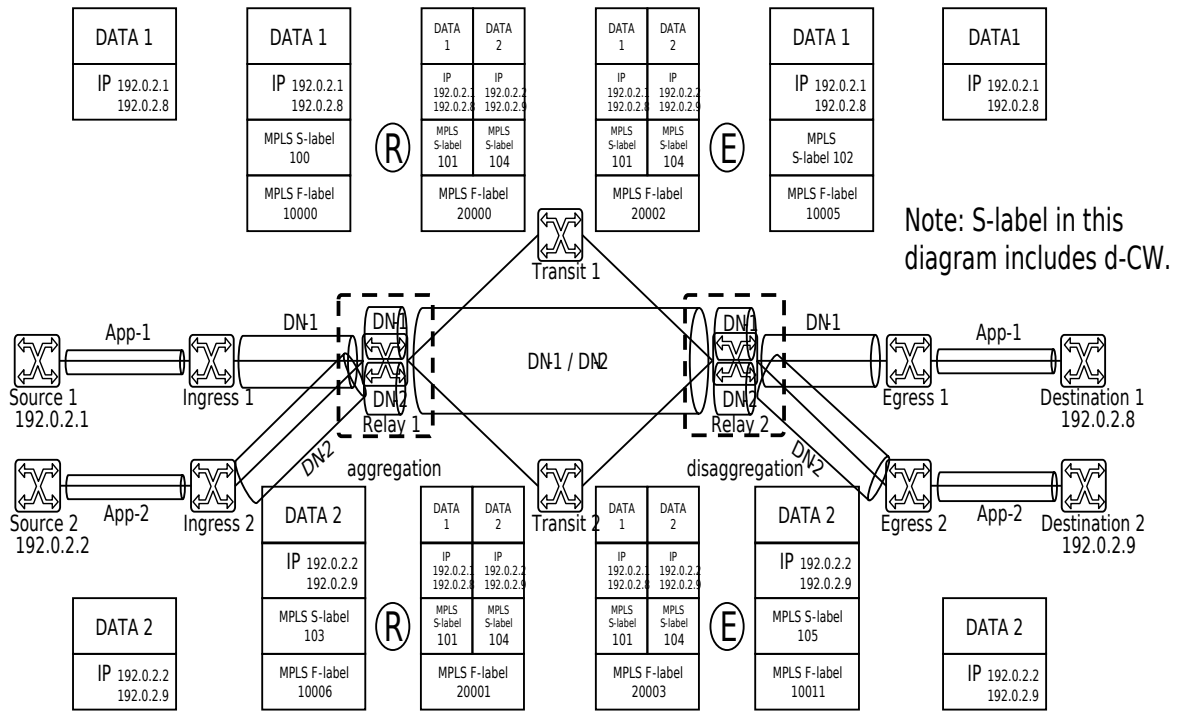


Figure 12: Case C-2 Example JSON Service Aggregation/Disaggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1",
          "afl-2"
        ]
      }
    ],
    "service": {
      "sub-layer": [
        {
          "name": "ssl-1",
          "service-rank": 10,
          "traffic-profile": "pf-1",
          "service-protection": {
            "protection": "replication",
            "sequence-number-length": "long-sn"
          },
          "operation": "relay",
          "incoming": {
            "service-id": {
              "mpls-label-stack": {
```

```
        "entry": [
          {
            "id": 0,
            "label": "100"
          }
        ]
      }
    },
    "outgoing": {
      "forwarding-sub-layer": {
        "service-outgoing": [
          {
            "index": 0,
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "101"
                }
              ]
            },
            "sub-layer": [
              "afl-1",
              "afl-2"
            ]
          }
        ]
      }
    },
    {
      "name": "ssl-2",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "replication",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "103"
              }
            ]
          }
        }
      },
      "outgoing": {
        "forwarding-sub-layer": {
          "service-outgoing": [
            {
              "index": 0,
              "mpls-label-stack": {
```

```
        "entry": [
          {
            "id": 0,
            "label": "104"
          }
        ]
      },
      "sub-layer": [
        "afl-1",
        "afl-2"
      ]
    ]
  }
}
],
},
"forwarding": {
  "sub-layer": [
    {
      "name": "fsl-1",
      "traffic-profile": "pf-2",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "10000"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-2",
    "traffic-profile": "pf-2",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10006"
            }
          ]
        }
      }
    }
  }
}
```

```
    ]
  }
}
},
"outgoing": {
  "service-sub-layer": {
    "sub-layer": [
      "ssl-2"
    ]
  }
}
},
{
  "name": "afl-1",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth2",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20000"
          }
        ]
      }
    }
  }
}
},
{
  "name": "afl-2",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20001"
          }
        ]
      }
    }
  }
}
```



```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1",
          "afl-2"
        ]
      }
    ],
    "service": {
      "sub-layer": [
        {
          "name": "ssl-1",
          "service-rank": 10,
          "traffic-profile": "pf-1",
          "service-protection": {
            "protection": "elimination",
            "sequence-number-length": "long-sn"
          },
          "operation": "relay",
          "incoming": {
            "service-id": {
              "mpls-label-stack": {
```

```
        "entry": [
          {
            "id": 0,
            "label": "101"
          }
        ]
      }
    },
    "outgoing": {
      "forwarding-sub-layer": {
        "service-outgoing": [
          {
            "index": 0,
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "102"
                }
              ]
            },
            "sub-layer": [
              "fsl-1"
            ]
          }
        ]
      }
    },
    {
      "name": "ssl-2",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "elimination",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "104"
              }
            ]
          }
        }
      },
      "outgoing": {
        "forwarding-sub-layer": {
          "service-outgoing": [
            {
              "index": 0,
              "mpls-label-stack": {
                "entry": [
```

```
        {
          "id": 0,
          "label": "105"
        }
      ]
    },
    "sub-layer": [
      "fsl-2"
    ]
  }
]
},
"forwarding": {
  "sub-layer": [
    {
      "name": "afl-1",
      "traffic-profile": "pf-3",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20002"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-1",
            "ssl-2"
          ]
        }
      }
    }
  ],
  {
    "name": "afl-2",
    "traffic-profile": "pf-3",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20003"
            }
          ]
        }
      }
    }
  ]
}
```

```
    }
  },
  "outgoing": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  }
},
{
  "name": "fsl-1",
  "traffic-profile": "pf-2",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth2",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10005"
          }
        ]
      }
    }
  }
},
{
  "name": "fsl-2",
  "traffic-profile": "pf-2",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10011"
          }
        ]
      }
    }
  }
}
```

```
    }
  }
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
```

Figure 14: Example C-2 DetNet JSON Relay Disaggregation Service Sub-Layer

B.6. Example C-3 JSON Relay Service Sub-Layer Aggregation/Disaggregation

This illustrates the Relay node 1 aggregating the service sub-layers of DetNet flows 1 and 2 into a service sub-layer of Aggregated DetNet flow 1. It also illustrates the Relay node 2 disaggregating the aggregated DetNet flow 1 into the DetNet flows 1 and 2 service sub-layers. A diagram illustrating both aggregation and disaggregation is shown and then the corresponding JSON operational data follows.

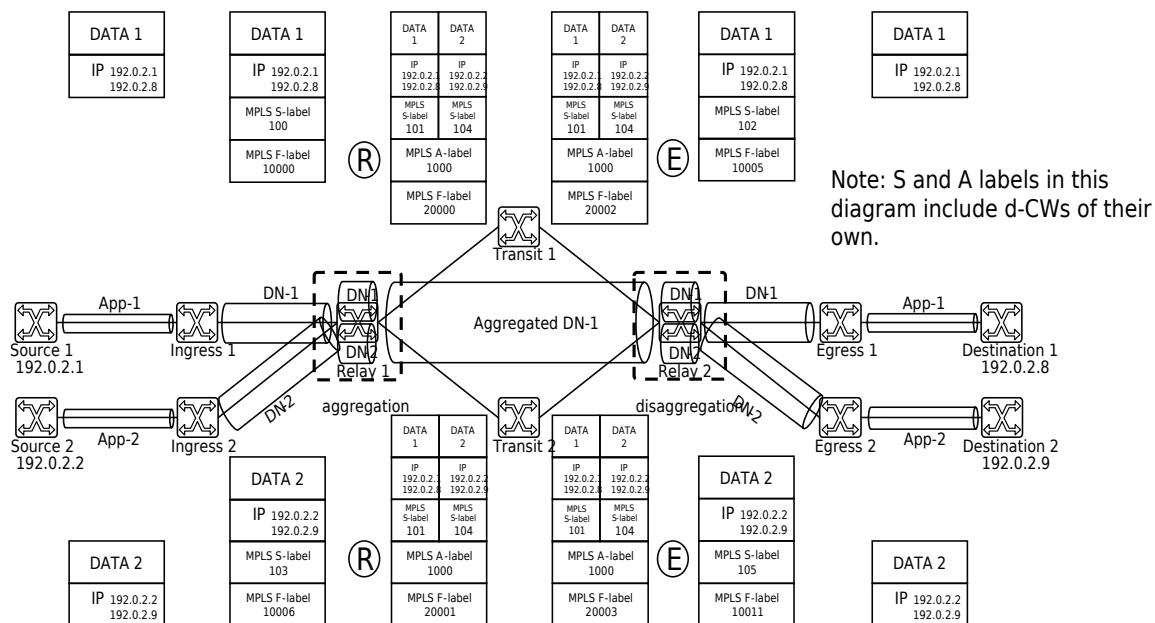


Figure 15: Case C-3 Example JSON Service Aggregation/Disaggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "asl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      },
      {
        "name": "pf-4",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-3",
          "fsl-4"
        ]
      }
    ]
  }
},
],
```

```
"service": {
  "sub-layer": [
    {
      "name": "ssl-1",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "none",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "100"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "aggregation-sub-layer": "asl-1",
          "service-label": {
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "101"
                }
              ]
            }
          }
        }
      }
    },
    {
      "name": "ssl-2",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "none",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "103"
              }
            ]
          }
        }
      }
    }
  ]
}
```

```
    }
  },
  "outgoing": {
    "service-sub-layer": {
      "aggregation-sub-layer": "asl-1",
      "service-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "104"
            }
          ]
        }
      }
    }
  }
},
{
  "name": "asl-1",
  "service-rank": 10,
  "traffic-profile": "pf-2",
  "service-protection": {
    "protection": "replication",
    "sequence-number-length": "long-sn"
  },
  "operation": "initiation",
  "incoming": {
    "service-aggregation": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "service-outgoing": [
        {
          "index": 0,
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "1000"
              }
            ]
          },
          "sub-layer": [
            "fsl-3",
            "fsl-4"
          ]
        }
      ]
    }
  }
}
]
```

```
    },
    "forwarding": {
      "sub-layer": [
        {
          "name": "fsl-1",
          "traffic-profile": "pf-3",
          "operation": "pop-and-lookup",
          "incoming": {
            "forwarding-id": {
              "interface": "eth0",
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "10000"
                  }
                ]
              }
            }
          },
          "outgoing": {
            "service-sub-layer": {
              "sub-layer": [
                "ssl-1"
              ]
            }
          }
        }
      ],
      {
        "name": "fsl-2",
        "traffic-profile": "pf-3",
        "operation": "pop-and-lookup",
        "incoming": {
          "forwarding-id": {
            "interface": "eth1",
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "10006"
                }
              ]
            }
          },
          "outgoing": {
            "service-sub-layer": {
              "sub-layer": [
                "ssl-2"
              ]
            }
          }
        }
      ],
      {
        "name": "fsl-3",
        "traffic-profile": "pf-4",
        "operation": "impose-and-forward",
        "incoming": {
```

```
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      },
      "outgoing": {
        "interface": {
          "outgoing-interface": "eth2",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20000"
              }
            ]
          }
        }
      }
    },
    {
      "name": "fsl-4",
      "traffic-profile": "pf-4",
      "operation": "impose-and-forward",
      "incoming": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      },
      "outgoing": {
        "interface": {
          "outgoing-interface": "eth3",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20001"
              }
            ]
          }
        }
      }
    }
  ]
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ],
},
```

```
{
  "name": "eth1",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth4",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
}
]
}
```

Figure 16: Example C-3 DetNet JSON Relay Service Sub-Layer Aggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "asl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-3",
          "fsl-4"
        ]
      },
      {
        "name": "pf-4",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      }
    ]
  }
},
],
```



```
"service": {
  "sub-layer": [
    {
      "name": "ssl-1",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "none",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "101"
              }
            ]
          }
        }
      },
      "outgoing": {
        "forwarding-sub-layer": {
          "service-outgoing": [
            {
              "index": 0,
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "102"
                  }
                ]
              }
            }
          ],
          "sub-layer": [
            "fsl-3"
          ]
        }
      }
    }
  ],
  {
    "name": "ssl-2",
    "service-rank": 10,
    "traffic-profile": "pf-1",
    "service-protection": {
      "protection": "none",
      "sequence-number-length": "long-sn"
    },
    "operation": "relay",
    "incoming": {
      "service-id": {
        "mpls-label-stack": {
          "entry": [
            {

```

```
        "id": 0,
        "label": "104"
      }
    ]
  }
},
"outgoing": {
  "forwarding-sub-layer": {
    "service-outgoing": [
      {
        "index": 0,
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "105"
            }
          ]
        },
        "sub-layer": [
          "fsl-4"
        ]
      }
    ]
  }
},
{
  "name": "asl-1",
  "service-rank": 10,
  "traffic-profile": "pf-2",
  "service-protection": {
    "protection": "elimination",
    "sequence-number-length": "long-sn"
  },
  "operation": "termination",
  "incoming": {
    "service-id": {
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "1000"
          }
        ]
      }
    }
  },
  "outgoing": {
    "service-disaggregation": {
      "sub-layer": [
        "ssl-1",
        "ssl-2"
      ]
    }
  }
}
}
```

```
]
},
"forwarding": {
  "sub-layer": [
    {
      "name": "fsl-1",
      "traffic-profile": "pf-4",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20002"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-2",
    "traffic-profile": "pf-4",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20003"
            }
          ]
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-3",
    "traffic-profile": "pf-3",
    "operation": "impose-and-forward",
```

```
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-1"
        ]
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth2",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10005"
            }
          ]
        }
      }
    }
  },
  {
    "name": "fsl-4",
    "traffic-profile": "pf-3",
    "operation": "impose-and-forward",
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-2"
        ]
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth3",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10011"
            }
          ]
        }
      }
    }
  }
]
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
```

```
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
```

Figure 17: Example C-3 DetNet JSON Relay Service Sub-Layer Disaggregation

B.7. Example C-4 JSON Relay Service Sub-Layer Aggregation/Disaggregation

This illustrates the Relay node 1 aggregating the forwarding sub-layers of DetNet flow 1 and 2 into a service sub-layer of Aggregated DetNet flow 1. This also illustrates the Relay node 2 disaggregating the service sub-layer of Aggregated DetNet flow 1 to forwarding sub-layers of DetNet flow 1 and 2. A diagram illustrating both aggregation and disaggregation is shown and then the corresponding JSON operational data follows.

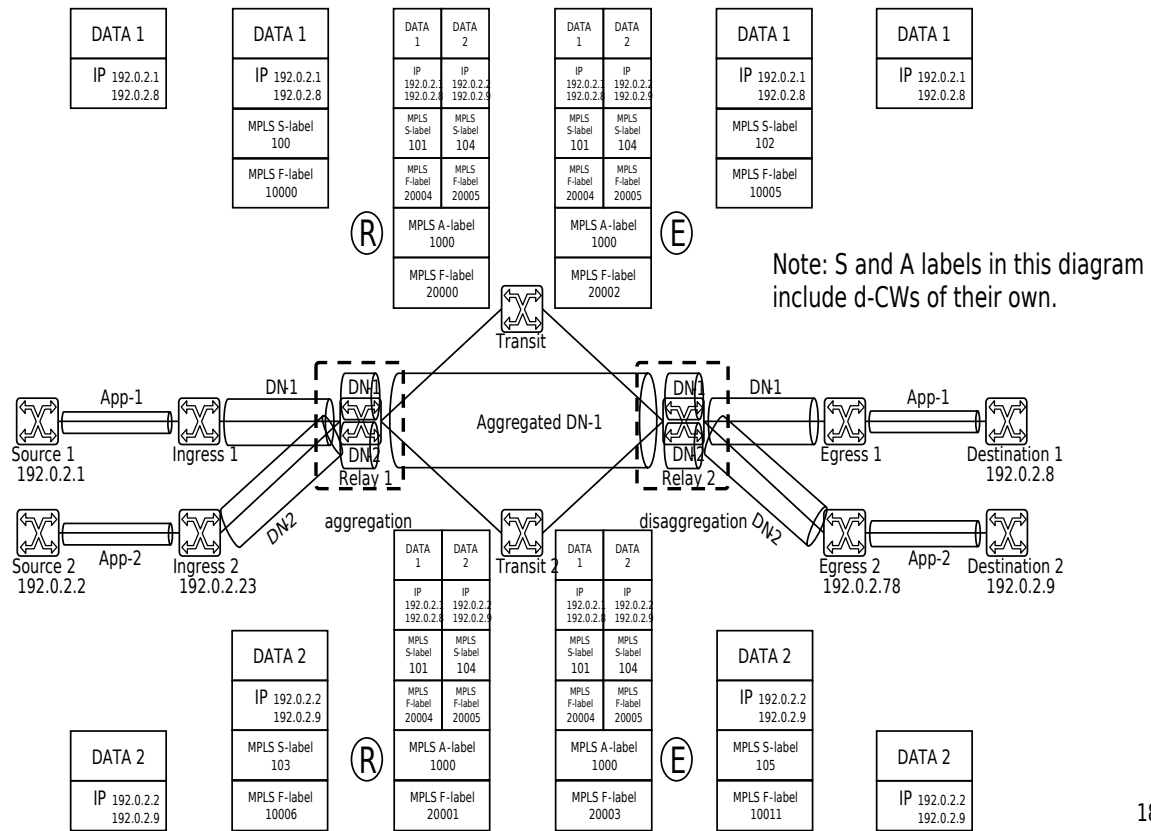


Figure 18: Case C-4 Example JSON Service Aggregation/Disaggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "asl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2",
          "fsl-3",
          "fsl-4"
        ]
      },
      {
        "name": "pf-4",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-5",
          "fsl-6"
        ]
      }
    ]
  }
}
```



```
    }
  ],
  "service": {
    "sub-layer": [
      {
        "name": "ssl-1",
        "service-rank": 10,
        "traffic-profile": "pf-1",
        "service-protection": {
          "protection": "none",
          "sequence-number-length": "long-sn"
        },
        "operation": "relay",
        "incoming": {
          "service-id": {
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "100"
                }
              ]
            }
          }
        },
        "outgoing": {
          "forwarding-sub-layer": {
            "service-outgoing": [
              {
                "index": 0,
                "mpls-label-stack": {
                  "entry": [
                    {
                      "id": 0,
                      "label": "101"
                    }
                  ]
                }
              ]
            },
            "sub-layer": [
              "fsl-3"
            ]
          }
        }
      }
    ],
    {
      "name": "ssl-2",
      "service-rank": 10,
      "traffic-profile": "pf-1",
      "service-protection": {
        "protection": "none",
        "sequence-number-length": "long-sn"
      },
      "operation": "relay",
      "incoming": {
        "service-id": {
          "mpls-label-stack": {
```

```
        "entry": [
          {
            "id": 0,
            "label": "103"
          }
        ]
      }
    },
    "outgoing": {
      "forwarding-sub-layer": {
        "service-outgoing": [
          {
            "index": 0,
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "104"
                }
              ]
            },
            "sub-layer": [
              "fsl-4"
            ]
          }
        ]
      }
    },
    {
      "name": "asl-1",
      "service-rank": 10,
      "traffic-profile": "pf-2",
      "service-protection": {
        "protection": "replication",
        "sequence-number-length": "long-sn"
      },
      "operation": "initiation",
      "incoming": {
        "forwarding-aggregation": {
          "sub-layer": [
            "fsl-3",
            "fsl-4"
          ]
        }
      },
      "outgoing": {
        "forwarding-sub-layer": {
          "service-outgoing": [
            {
              "index": 0,
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "1000"
                  }
                ]
              }
            }
          ]
        }
      }
    }
  ]
}
```

```
    ],
    },
    "sub-layer": [
      "fsl-5",
      "fsl-6"
    ]
  ]
}
}
}
}
},
"forwarding": {
  "sub-layer": [
    {
      "name": "fsl-1",
      "traffic-profile": "pf-3",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "10000"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-2",
    "traffic-profile": "pf-3",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10006"
            }
          ]
        }
      },
      "outgoing": {
```

```
        "service-sub-layer": {
          "sub-layer": [
            "ssl-2"
          ]
        }
      },
    },
    {
      "name": "fsl-3",
      "traffic-profile": "pf-3",
      "operation": "impose-and-forward",
      "incoming": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-1"
          ]
        }
      },
      "outgoing": {
        "service-aggregation": {
          "aggregation-sub-layer": "asl-1",
          "optional-forwarding-label": {
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "20004"
                }
              ]
            }
          }
        }
      }
    },
    {
      "name": "fsl-4",
      "traffic-profile": "pf-3",
      "operation": "impose-and-forward",
      "incoming": {
        "service-sub-layer": {
          "sub-layer": [
            "ssl-2"
          ]
        }
      },
      "outgoing": {
        "service-aggregation": {
          "aggregation-sub-layer": "asl-1",
          "optional-forwarding-label": {
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "20005"
                }
              ]
            }
          }
        }
      }
    }
  ]
}
```

```
    }
  },
  {
    "name": "fsl-5",
    "traffic-profile": "pf-4",
    "operation": "impose-and-forward",
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "asl-1"
        ]
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth2",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20000"
            }
          ]
        }
      }
    }
  },
  {
    "name": "fsl-6",
    "traffic-profile": "pf-4",
    "operation": "impose-and-forward",
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "asl-1"
        ]
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth3",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20001"
            }
          ]
        }
      }
    }
  ]
},
"ietf-interfaces:interfaces": {
  "interface": [
```

```
{
  "name": "eth0",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth1",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth4",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
}
]
}
```

Figure 19: Example C-4 DetNet JSON Relay Service Sub-Layer Aggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": 2,
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-service": [
          "asl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-3",
          "fsl-4",
          "fsl-5",
          "fsl-6"
        ]
      },
      {
        "name": "pf-4",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      }
    ]
  }
}
```



```
    }
  ],
  "service": {
    "sub-layer": [
      {
        "name": "ssl-1",
        "service-rank": 10,
        "traffic-profile": "pf-1",
        "service-protection": {
          "protection": "none",
          "sequence-number-length": "long-sn"
        },
        "operation": "relay",
        "incoming": {
          "service-id": {
            "mpls-label-stack": {
              "entry": [
                {
                  "id": 0,
                  "label": "101"
                }
              ]
            }
          }
        },
        "outgoing": {
          "forwarding-sub-layer": {
            "service-outgoing": [
              {
                "index": 0,
                "mpls-label-stack": {
                  "entry": [
                    {
                      "id": 0,
                      "label": "102"
                    }
                  ]
                }
              ],
              "sub-layer": [
                "fsl-5"
              ]
            }
          }
        }
      },
      {
        "name": "ssl-2",
        "service-rank": 10,
        "traffic-profile": "pf-1",
        "service-protection": {
          "protection": "none",
          "sequence-number-length": "long-sn"
        },
        "operation": "relay",
        "incoming": {
          "service-id": {
            "mpls-label-stack": {
```



```
    }
  ]
},
"forwarding": {
  "sub-layer": [
    {
      "name": "fsl-1",
      "traffic-profile": "pf-4",
      "operation": "pop-and-lookup",
      "incoming": {
        "forwarding-id": {
          "interface": "eth0",
          "mpls-label-stack": {
            "entry": [
              {
                "id": 0,
                "label": "20002"
              }
            ]
          }
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-2",
    "traffic-profile": "pf-4",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20003"
            }
          ]
        }
      },
      "outgoing": {
        "service-sub-layer": {
          "sub-layer": [
            "asl-1"
          ]
        }
      }
    }
  ],
  {
    "name": "fsl-3",
```

```
"traffic-profile": "pf-3",
"operation": "pop-and-lookup",
"incoming": {
  "forwarding-id": {
    "interface": "eth0",
    "mpls-label-stack": {
      "entry": [
        {
          "id": 0,
          "label": "20004"
        }
      ]
    }
  }
},
"outgoing": {
  "service-sub-layer": {
    "sub-layer": [
      "ssl-1"
    ]
  }
},
{
  "name": "fsl-4",
  "traffic-profile": "pf-3",
  "operation": "pop-and-lookup",
  "incoming": {
    "forwarding-id": {
      "interface": "eth1",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20005"
          }
        ]
      }
    }
  },
  "outgoing": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-2"
      ]
    }
  }
},
{
  "name": "fsl-5",
  "traffic-profile": "pf-3",
  "operation": "impose-and-forward",
  "incoming": {
    "service-sub-layer": {
      "sub-layer": [
        "ssl-1"
      ]
    }
  }
}
```

```
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth2",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10005"
            }
          ]
        }
      }
    }
  },
  {
    "name": "fsl-6",
    "traffic-profile": "pf-3",
    "operation": "impose-and-forward",
    "incoming": {
      "service-sub-layer": {
        "sub-layer": [
          "ssl-2"
        ]
      }
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10011"
          }
        ]
      }
    }
  }
}
],
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ],
  {
    "name": "eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
```

```
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
},
{
  "name": "eth4",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2020-12-18T18:59:00-05:00"
  }
}
]
}
```

Figure 20: Example C-4 DetNet JSON Relay Service Sub-Layer Disaggregation

B.8. Example D-1 JSON Transit Forwarding Sub-Layer Aggregation/Disaggregation

This illustrates the Transit node 1 aggregating the forwarding sub-layers of DetNet flow 1 and 2 into a forwarding sub-layer. This also illustrates a Transit node 4 disaggregating a forwarding sub-layer into DetNet flow 1 and 2 forwarding sub-layers.

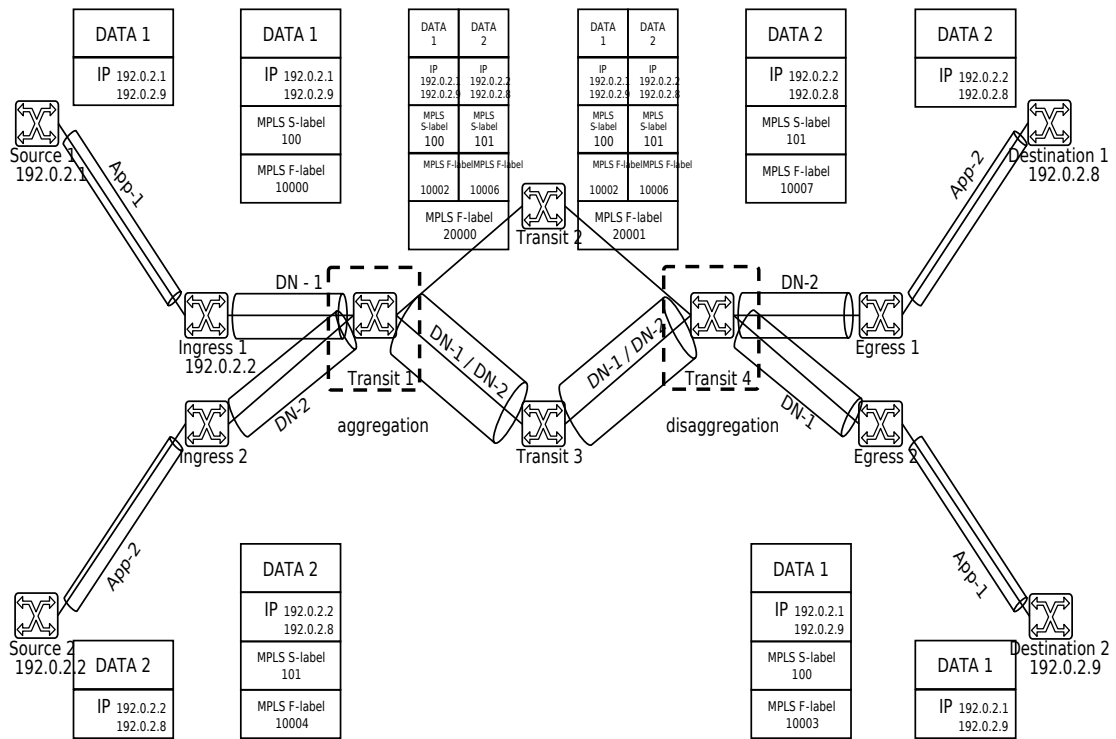


Figure 21: Case D-1 Example Service Aggregation/Disaggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1"
        ]
      }
    ],
    "forwarding": {
      "sub-layer": [
        {
          "name": "fsl-1",
          "traffic-profile": "pf-1",
          "operation": "pop-impose-and-forward",
          "incoming": {
            "forwarding-id": {
              "interface": "eth0",
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "10000"
                  }
                ]
              }
            }
          },
          "outgoing": {
            "forwarding-sub-layer": {
              "aggregation-sub-layer": "afl-1",
              "forwarding-label": {
                "mpls-label-stack": {
                  "entry": [
                    {
                      "id": 0,
                      "label": "10002"
                    }
                  ]
                }
              }
            }
          }
        ]
      }
    ]
  }
}
```

```
    }
  }
},
{
  "name": "fsl-2",
  "traffic-profile": "pf-1",
  "operation": "pop-impose-and-forward",
  "incoming": {
    "forwarding-id": {
      "interface": "eth1",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "10004"
          }
        ]
      }
    }
  },
  "outgoing": {
    "forwarding-sub-layer": {
      "aggregation-sub-layer": "afl-1",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10006"
            }
          ]
        }
      }
    }
  }
},
{
  "name": "afl-1",
  "traffic-profile": "pf-2",
  "operation": "impose-and-forward",
  "incoming": {
    "forwarding-aggregation": {
      "sub-layer": [
        "fsl-1",
        "fsl-2"
      ]
    }
  },
  "outgoing": {
    "interface": {
      "outgoing-interface": "eth3",
      "mpls-label-stack": {
        "entry": [
          {
            "id": 0,
            "label": "20000"
          }
        ]
      }
    }
  }
}
```

```
    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 22: Example D-1 DetNet JSON Relay Service Sub-Layer Aggregation


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 1,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "fsl-1",
          "fsl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "afl-1"
        ]
      }
    ],
    "forwarding": {
      "sub-layer": [
        {
          "name": "fsl-1",
          "traffic-profile": "pf-1",
          "operation": "swap-and-forward",
          "incoming": {
            "forwarding-id": {
              "interface": "eth1",
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "10002"
                  }
                ]
              }
            }
          },
          "outgoing": {
            "interface": {
              "outgoing-interface": "eth3",
              "mpls-label-stack": {
                "entry": [
                  {
                    "id": 0,
                    "label": "10003"
                  }
                ]
              }
            }
          }
        }
      ]
    }
  }
}
```

```
    }
  },
  {
    "name": "fsl-2",
    "traffic-profile": "pf-1",
    "operation": "swap-and-forward",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10006"
            }
          ]
        }
      }
    },
    "outgoing": {
      "interface": {
        "outgoing-interface": "eth2",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "10007"
            }
          ]
        }
      }
    }
  },
  {
    "name": "afl-1",
    "traffic-profile": "pf-2",
    "operation": "pop-and-lookup",
    "incoming": {
      "forwarding-id": {
        "interface": "eth1",
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,
              "label": "20001"
            }
          ]
        }
      }
    },
    "outgoing": {
      "forwarding-disaggregation": {
        "sub-layer": [
          "fsl-1",
          "fsl-2"
        ]
      }
    }
  }
}
```

```
    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    },
    {
      "name": "eth4",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2020-12-18T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 23: Example D-1 DetNet JSON Relay Service Sub-Layer Disaggregation

Authors' Addresses

Xuesong Geng

Huawei Technologies

Email: gengxuesong@huawei.com**Yeoncheol Ryoo**

ETRI

Email: dbduscjf@etri.re.kr**Don Fedyk**

LabN Consulting, L.L.C.

Email: dfedyk@labn.net**Reshad Rahman**

Individual

Email: reshad@yahoo.com**Zhenqiang Li**

China Mobile

Email: lizhenqiang@chinamobile.com