
Stream: Internet Engineering Task Force (IETF)
RFC: [9633](#)
Category: Standards Track
Published: September 2024
ISSN: 2070-1721
Authors: X. Geng Y. Ryoo D. Fedyk R. Rahman
Huawei Technologies ETRI LabN Consulting, L.L.C. Equinix
Z. Li
China Mobile

RFC 9633

Deterministic Networking (DetNet) YANG Data Model

Abstract

This document contains the specification for the Deterministic Networking (DetNet) YANG data model for configuration and operational data for DetNet flows. The model allows the provisioning of an end-to-end DetNet service on devices along the path without depending on any signaling protocol. It also specifies operational status for flows.

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9633>.

Copyright Notice

Copyright (c) 2024 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Abbreviations	3
3. Terminology	4
4. DetNet YANG Module	4
4.1. DetNet Application Flow YANG Attributes	5
4.2. DetNet Service Sub-layer YANG Attributes	5
4.3. DetNet Forwarding Sub-layer YANG Attributes	5
5. DetNet Flow Aggregation	6
6. DetNet YANG Structure Considerations	7
7. DetNet Configuration YANG Structures	7
8. DetNet Configuration YANG Data Model	8
9. IANA Considerations	35
10. Security Considerations	36
11. References	37
11.1. Normative References	37
11.2. Informative References	38
Appendix A. DetNet Configuration YANG Tree	39
Appendix B. Examples	47
B.1. Example A-1: Application Flow Aggregation	47
B.2. Example B-1: Aggregation Using a Forwarding Sub-layer	53
B.3. Example B-2: Service Aggregation	58
B.4. Example C-1: DetNet Relay Service Sub-layer	64
B.5. Example C-2: DetNet Relay Service Sub-layer Aggregation/Disaggregation	79
B.6. Example C-3: DetNet Relay Service Sub-layer Aggregation/Disaggregation	90

B.7. Example C-4: DetNet Relay Service Sub-layer Aggregation/Disaggregation	103
B.8. Example D-1: Transit Node Forwarding Sub-layer Aggregation/Disaggregation	118
Acknowledgments	125
Contributors	125
Authors' Addresses	126

1. Introduction

DetNet (Deterministic Networking) provides the ability 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 data model for DetNet based on YANG data types and modeling language defined in [\[RFC6991\]](#) and [\[RFC7950\]](#).

This document also includes the following:

- The DetNet service, which is designed to describe the characteristics of services being provided for application flows over a network.
- DetNet configuration, which is designed to provide DetNet flow path establishment, flow status reporting, and configuration of DetNet functions in order to achieve end-to-end bounded latency and zero congestion loss.

This YANG data model is scoped to the description of the aggregation/disaggregation and data plane capabilities of the DetNet data planes defined in "[Deterministic Networking Architecture](#)" [\[RFC8655\]](#) and "[Deterministic Networking \(DetNet\) Data Plane Framework](#)" [\[RFC8938\]](#). DetNet operates at the IP layer and delivers service over lower-layer technologies such as MPLS and IEEE 802.1 Time-Sensitive Networking (TSN).

2. Abbreviations

The following abbreviations are used in this document:

PEF: Packet Elimination Function

POF: Packet Ordering Function

PRF: Packet Replication Function

PREOF: Packet Replication, Elimination, and Ordering Functions

MPLS: Multiprotocol Label Switching

3. Terminology

This document uses the terminology defined in [RFC8655]. The terms "A-Label", "S-Label", and "F-Label" are used in this document as defined in [RFC8964].

4. DetNet YANG Module

The DetNet YANG module (Section 8) 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 Sections 4.1, 4.2, and 4.3, respectively.

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

Data Instance		
Edge Node	Relay Node	Transit Node
App-Flow Data Layer		
Service Sub-layer	Service Sub-layer	
Forwarding Sub-layer	Forwarding Sub-layer	Forwarding Sub-layer

Table 1: DetNet Layers and Node Types

All of the layers have ingress/incoming and egress/outgoing operations, but any instance may be configured as unidirectional only. "Ingress" refers to any DetNet layer where a DetNet context is applied. Ingress allows functions such as switching, aggregation, and encapsulation. "Egress" refers to any DetNet layer where a DetNet context is removed. Egress allows functions such as switching, disaggregation, and decapsulation. This means that each unidirectional flow identifier configuration is programmed starting at the ingress and flow status is reported at the ingress on each end. In the case of MPLS, once encapsulated, the IP 6-tuple parameters (see [RFC8938]) may not be required to be programmed again. In the case of IP, without encapsulation, various IP flow identification parameters must be configured along the flow path.

In the YANG data model defined in this document, 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" means "to the DetNet application", and "egress" means "from the application". The terms "incoming" and "outgoing" represent the flow direction towards the remote application as a unidirectional flow. This means the terms are used at a sub-layer to represent "incoming" to the sub-layer function and "outgoing" is viewed as leaving the sub-layer.

For the service sub-layer, "incoming" is typically aggregating applications flows or other service sub-layers, etc. For the forwarding sub-layer, "incoming" is typically aggregating service sub-layers. However, this also means for both service and forwarding sub-layers at the egress DetNet node "incoming" also handles external flows "incoming" to the respective sub-layer. For MPLS, this would usually involve the removal of a label. For IP -- where the representative sub-layer is merely an aggregation of an IP prefix or IP tuple -- there may be no incoming/outgoing definitions, since the arriving packet can be handled directly by a standard next-hop routing decision. In examples ([Appendix B](#)) where both aggregation and disaggregation take place, at the egress of the flow "outgoing" relates to the aggregating output and "incoming" relates to the disaggregating flows.

At the egress point, forwarding information is determined by the App-flow type with all DetNet-related headers removed. In the case of IP, the forwarding information can specify an output port or set a next-hop address. In the case of MPLS, it can set an MPLS label.

4.1. DetNet Application Flow YANG Attributes

DetNet application flows are 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 defined in [\[RFC9016\]](#) are used.

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 type. Indicates which service function will be invoked at a DetNet edge, relay node, or end station. (DetNet tunnel initialization and termination are default functions in the DetNet service sub-layer, so there is no need to indicate them explicitly.) The corresponding arguments for service functions also need to be defined.

4.3. DetNet Forwarding Sub-layer YANG Attributes

As defined in [\[RFC8655\]](#), the DetNet forwarding sub-layer optionally provides congestion protection for DetNet flows over paths provided by the underlying network. Explicit routes provide another mechanism used by DetNet to avoid temporary interruptions caused by the convergence of routing or bridging protocols. Explicit routes are also implemented at the DetNet forwarding sub-layer.

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

- Flow specification and traffic requirements are as described in the information model provided in [RFC9016]. These may be 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 needs 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 techniques.

5. DetNet Flow Aggregation

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

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

- Aggregated flow encapsulation/decapsulation/identification.
- Mapping individual DetNet flows to an aggregated flow.
- Changing traffic specification parameters for aggregated flows.

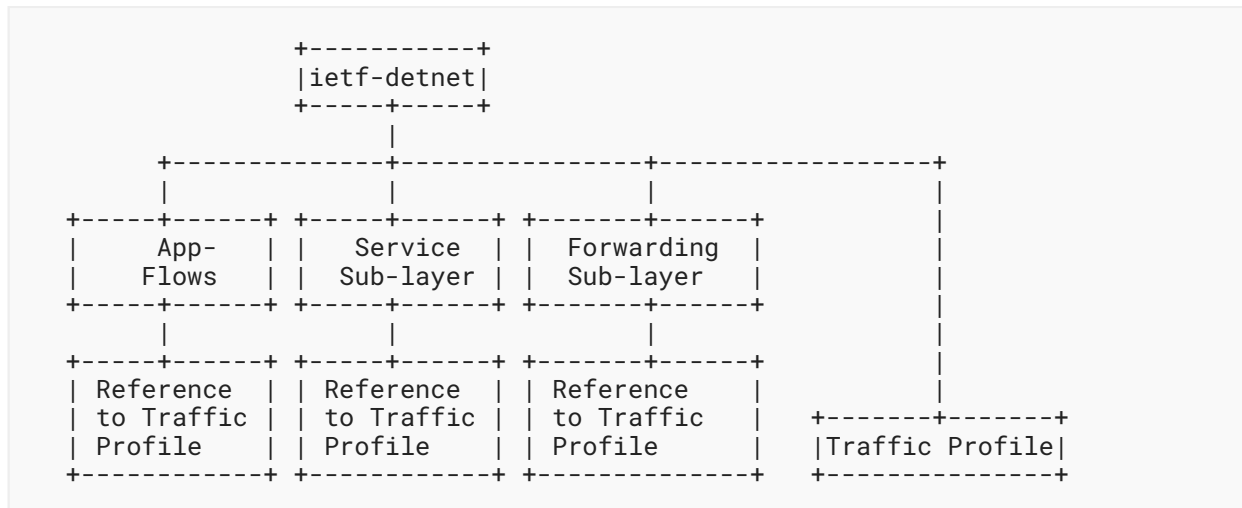
The following DetNet aggregation scenarios are supported:

- The ingress node aggregates App-flows into a service sub-layer of a DetNet flow.
- In the ingress node, the service sub-layers of DetNet flows are aggregated into a forwarding sub-layer.
- In the ingress node, the service sub-layers of DetNet flows are aggregated into a service sub-layer of an aggregated DetNet flow.
- The relay node aggregates the forwarding sub-layers of DetNet flows into a forwarding sub-layer.
- The relay node aggregates the service sub-layers of DetNet flows into a forwarding sub-layer.
- The relay node aggregates the service sub-layers of DetNet flows into a service sub-layer of an aggregated DetNet flow.
- The relay node aggregates the forwarding sub-layers of DetNet flows into a service sub-layer of an aggregated DetNet flow.
- The transit node aggregates the forwarding sub-layers of DetNet flows into a forwarding sub-layer.

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

6. DetNet YANG Structure Considerations

This diagram shows the general structure of the DetNet YANG data model:



There are three layer types in the DetNet YANG data model: the App-flow data layer, the service sub-layer, and the 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. The complete YANG tree is provided in [Appendix A](#).

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

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

7. DetNet Configuration YANG Structures

The following is a partial tree representation of the DetNet YANG data model, per the guidelines provided in [\[RFC8340\]](#). This corresponds to the layout of the diagram in [Section 6](#).

```

module: ietf-detnet
+--rw detnet
  +--rw traffic-profile* [name]
    | +--rw name string
    | +--rw traffic-requirements
    | +--rw traffic-spec
    | +--ro member-app-flow* app-flow-ref
    | +--ro member-svc-sublayer* 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? mpls-fwd-operation
      +--rw incoming
      |   ...
      +--rw outgoing
      |   ...

```

8. DetNet Configuration YANG Data Model

This YANG data model imports typedefs from [\[RFC6991\]](#), [\[RFC8519\]](#), [\[RFC8294\]](#), [\[RFC8343\]](#), and [\[IEEE8021Q-2022\]](#). This YANG data model also includes the following RFC references, which are not cited elsewhere in the body of this document: [\[RFC0791\]](#), [\[RFC4303\]](#), [\[RFC8200\]](#), [\[RFC8349\]](#), and [\[RFC8960\]](#).

```
<CODE BEGINS> file "ietf-detnet@2024-08-15.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 802.1Q-2022: IEEE Standard for Local and Metropolitan
    Area Networks--Bridges and Bridged Networks,
    Clause 48 ('YANG Data Models')";
}

organization
  "IETF DetNet Working Group";

contact
  "WG Web: <https://datatracker.ietf.org/wg/detnet/>
  WG List: <mailto:detnet@ietf.org>

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

  Author: Yeoncheol Ryoo
  <mailto:dbduscjf@etri.re.kr>

  Author: Don Fedyk
  <mailto:dfedyk@labn.net>
```

```
Author: Reshad Rahman
       <mailto:reshad@yahoo.com>

Author: 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) 2024 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 Revised 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).
```

```
    "RFC 9016: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.8";
  }

  identity failed {
    base app-status;
    description
      "The application is ingress/egress failed.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.8";
  }

  identity out-of-service {
    base app-status;
    description
      "The application is administratively blocked.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.8";
  }

  identity partial-failed {
    base app-status;
    description
      "This is an application with one or more egress-ready
      instances and one or more instances where egress failed.
      The DetNet flow can be used if the ingress's status is
      'ready'.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.8";
  }

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

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

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

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

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

typedef ipsec-spi {
  type uint32 {
    range "1..max";
  }
  description
    "IPsec Security Parameters Index. A 32-bit value,
    where some values are reserved.";
  reference
    "RFC 4303: IP Encapsulating Security Payload (ESP)";
}

typedef operation {
  type enumeration {
    enum initiation {
      description
        "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 the DetNet service sub-layer.";
    }
  }
  description
    "The operation type identifies this service sub-layer's
    behavior. Operations are described as unidirectional,
    but a service sub-layer may combine operation types.";
}

typedef mpls-fwd-operation {
  type enumeration {
    enum impose-and-forward {
```

```
    description
      "This operation imposes one or more outgoing labels and
      forwards to the next hop.";
    reference
      "RFC 8960: A YANG Data Model for MPLS Base";
  }
  enum pop-and-forward {
    description
      "This operation pops the incoming label and forwards to
      the next hop.";
    reference
      "RFC 8960: A YANG Data Model for MPLS Base";
  }
  enum pop-impose-and-forward {
    description
      "This operation pops the incoming label, imposes one or
      more outgoing labels, and forwards to the next hop.";
    reference
      "RFC 8960: A YANG Data Model for MPLS Base";
  }
  enum swap-and-forward {
    description
      "This operation swaps an incoming label with an outgoing
      label and forwards to the next hop.";
    reference
      "RFC 8960: A YANG Data Model for MPLS Base";
  }
  enum forward {
    description
      "This operation forwards to the next hop.";
  }
  enum pop-and-lookup {
    description
      "This operation pops an incoming label and performs a
      lookup.";
    reference
      "RFC 8960: A YANG Data Model for MPLS Base";
  }
}
description
  "MPLS operation types. This set of enums is modeled after
  the MPLS enums. With the exception of 'enum forward',
  these enums are the same as those provided in RFC 8960.";
reference
  "RFC 8960: A YANG Data Model for MPLS Base";
}

typedef service-protection {
  type enumeration {
    enum none {
      description
        "Service protection is not 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. A 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. A PEF can be
        implemented by an edge node, a relay node, or an
        end system.";
    }
    enum ordering {
        description
        "A Packet Ordering Function (POF) reorders 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 a 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 a 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 a 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 enumeration
values.";
}

typedef sequence-number-generation {
    type enumeration {
        enum copy-from-app-flow {
            description
            "'copy-from-app-flow' is used to utilize the sequence
            number present in the App-flow. This function is
            required when encapsulating App-flows that have been
            replicated and received through multiple ingress nodes
            into a member flow. When a relay node sees the same
            sequence number on an App-flow, it may be programmed
            to eliminate duplicate App-flow packets.";
        }
        enum generate-by-detnet-flow {
            description
            "'generate-by-detnet-flow' is used to create a new

```

```
        sequence number for a DetNet flow at the ingress node.
        Care must be taken when using this option to ensure
        that there is only one source for generating sequence
        numbers.";
    }
}
description
  "This typedef defines how to generate sequence numbers to
  be used in DetNet encapsulation.";
}
typedef sequence-number-field {
  type enumeration {
    enum zero-sn {
      description
        "The DetNet sequence number field is not 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
  "These enums configure the behavior of the
  sequence number field.";
}

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
      "In IPv4, this field refers to the protocol of the
      payload. In IPv6, this field is known as
      'next-header'; it identifies the type of header
      immediately following the IPv6 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 in 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 Time-Sensitive Networking (TSN) packet
        header information.";
    leaf source-mac-address {
        type yang:mac-address;
        description
            "The source Media Access Control (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 802.1Q-2022: IEEE Standard for Local and
        Metropolitan Area Networks--Bridges and Bridged
        Networks";
    }
    leaf pcq {
        type dot1q-types:priority-type;
        description
            "The priority value of the Ethernet header.";
        reference
            "IEEE 802.1Q-2022: IEEE Standard for Local and
            Metropolitan Area Networks--Bridges and Bridged
            Networks";
    }
}

grouping destination-ip-port-id {
    description
        "The TCP/UDP port 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 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';
        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 in the header. IPv6 only.";
    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
        "RFC 4303: IP 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 the 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 the application.";
    }
    container ip-app-flow {
      uses ip-flow-id;
      description
        "The IP header for the application.";
    }
    container mpls-app-flow {
      uses mpls-flow-id;
      description
        "The MPLS header for the 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
    "Reference group for incoming or outgoing App-flows.";
  leaf-list flow {
    type app-flow-ref;
    description
      "List of ingress or egress App-flows.";
  }
}

grouping service-sub-layer-group {
  description
    "Reference group for incoming or outgoing
    service sub-layers.";
  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
```

```
    "Reference group for incoming or outgoing
    forwarding sub-layers.";
  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 information 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 for DetNet next hops. These follow the
    principles for next hops as discussed in RFC 8349.";
  reference
    "RFC 8349: A YANG Data Model for Routing Management
    (NMDA Version)";
  choice next-hop-options {
    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 {
```



```
(NMDA Version)";
choice next-hop-options {
  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 {
      description
        "Use the IP data plane for forwarding.";
      leaf next-hop-address {
        type inet:ip-address;
        description
          "This is an IP address as a next hop.";
      }
      uses ip-header;
    }
    case mpls {
      description
        "Use the MPLS data plane for forwarding.";
      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 next 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 {
```

```

        description
          "Use the IP data plane for forwarding.";
        leaf next-hop-address {
          type inet:ip-address;
          description
            "This is an IP address as a next hop.";
        }
        uses ip-header;
      }
    case mpls {
      description
        "Use the MPLS data plane for forwarding.";
      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
        "The name of the traffic profile that is used as a
        reference to this profile.";
    }
  }
  container traffic-requirements {
    description
      "This defines the attributes of the App-flow
      regarding bandwidth, latency, latency variation, loss,
      and misordering tolerance.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.9";
    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: Flow and Service Information Model for
        Deterministic Networking (DetNet), Section 5.9.1";
    }
    leaf max-latency {
      type uint32;
      units "nanoseconds";
      description

```

```
        "This is the maximum latency from the ingress to
        one or more egresses for a single packet of the
        DetNet flow. MaxLatency is specified as an
        integer number of nanoseconds. The maximum value
        for this parameter is 4,294,967,295 nanoseconds.";
    reference
    "RFC 9016: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.9.2";
}
leaf max-latency-variation {
    type uint32;
    units "nanoseconds";
    description
    "This is the difference between the
    minimum and maximum end-to-end one-way latency.
    MaxLatencyVariation is specified as an integer
    number of nanoseconds.";
    reference
    "RFC 9016: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.9.3";
}
leaf max-loss {
    type decimal64 {
        fraction-digits 10;
        range "0 .. 100";
    }
    units "percent";
    description
    "This defines the maximum Packet Loss Rate (PLR)
    parameter for the DetNet service between the ingress
    and one or more egresses of the DetNet domain. The
    PLR is calculated by the number of transmitted
    packets minus the number of received packets divided
    by the number of transmitted packets, expressed as a
    percentage.";
    reference
    "RFC 9016: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.9.4";
}
leaf max-consecutive-loss-tolerance {
    type uint32;
    units "packets";
    description
    "Some applications have special loss requirements
    and use such parameters as
    MaxConsecutiveLossTolerance.
    'max-consecutive-loss-tolerance' 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: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.9.5";
}
leaf max-misordering {
    type uint32;
    units "packets";
}
```



```
    description
      "This describes the maximum tolerable number of
      packets that can be received out of order. The
      maximum allowed misordering can be measured, for
      example, based on sequence number. A value of '0'
      for the maximum allowed misordering indicates that
      in-order delivery is required and misordering cannot
      be tolerated.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.9.6";
  }
}
container traffic-spec {
  description
    "'traffic-spec' 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: Flow and Service Information Model for
    Deterministic Networking (DetNet), Section 5.5";
  leaf interval {
    type uint32;
    units "nanoseconds";
    description
      "The period of time during which the traffic
      specification should not be exceeded.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.5
      IEEE 802.1Q-2022: IEEE Standard for Local and
      Metropolitan Area Networks--Bridges and Bridged
      Networks";
  }
  leaf max-pkts-per-interval {
    type uint32;
    description
      "The maximum number of packets that the
      source will transmit in one interval.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.5
      IEEE 802.1Q-2022: IEEE Standard for Local and
      Metropolitan Area Networks--Bridges and Bridged
      Networks";
  }
  leaf max-payload-size {
    type uint32;
    description
      "The maximum payload size that the source
      will transmit.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.5
      IEEE 802.1Q-2022: IEEE Standard for Local and
      Metropolitan Area Networks--Bridges and Bridged
```

```
        Networks";
    }
    leaf min-payload-size {
        type uint32;
        description
            "The minimum payload size that the source
            will transmit.";
        reference
            "RFC 9016: Flow and Service Information Model for
            Deterministic Networking (DetNet), Section 5.5
            IEEE 802.1Q-2022: IEEE Standard for Local and
            Metropolitan Area Networks--Bridges and Bridged
            Networks";
    }
    leaf min-pkts-per-interval {
        type uint32;
        description
            "The minimum number of packets that the
            source will transmit in one interval.";
        reference
            "RFC 9016: Flow and Service Information Model for
            Deterministic Networking (DetNet), Section 5.5
            IEEE 802.1Q-2022: IEEE Standard for Local and
            Metropolitan Area Networks--Bridges and Bridged
            Networks";
    }
}
leaf-list member-app-flow {
    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 9633: Deterministic Networking (DetNet) YANG Data
        Model, Section 6";
}
leaf-list member-svc-sublayer {
    type service-sub-layer-ref;
    config false;
    description
        "A list of service sub-layers attached to this profile.
        Each service sub-layer that uses a profile has an
        automatically populated reference.";
    reference
        "RFC 9633: Deterministic Networking (DetNet) YANG Data
        Model, Section 6";
}
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-layer that uses a profile has an
        automatically populated reference.";
    reference
        "RFC 9633: Deterministic Networking (DetNet) YANG Data
```

```
        Model, Section 6";
    }
}
container app-flows {
  description
    "Configuration information for DetNet App-flows.";
  reference
    "RFC 9016: Flow and Service Information Model for
    Deterministic Networking (DetNet), 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: Flow and Service Information Model for
        Deterministic Networking (DetNet), Sections 4.1
        and 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: Flow and Service Information Model for
        Deterministic Networking (DetNet), Section 4.2";
    }
    leaf outgoing-service {
      type service-sub-layer-ref;
      config false;
      description
        "Binding to this application's outgoing service.";
    }
    leaf incoming-service {
      type service-sub-layer-ref;
      config false;
      description
        "Binding to this application's 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 an ingress application flow. This is an  
        operational status and defaults to 'none' if  
        incomplete.";  
    reference  
        "RFC 9016: Flow and Service Information Model for  
        Deterministic Networking (DetNet), Sections 4.1  
        and 5.8";  
    }  
    leaf-list interface {  
        type if:interface-ref;  
        description  
            "An interface is optional for a service type.  
            When matching a flow to a single interface,  
            one interface is specified. This list allows  
            the matching of a subset of interfaces.  
            When more than one interface is specified, these  
            flows are simply aggregated, and the service  
            sub-layer is unaware of the aggregation.";  
    }  
    uses data-flow-spec;  
    }  
    container egress {  
        description  
            "Egress DetNet application flows or a compound flow.";  
        uses data-flow-spec;  
        choice application-type {  
            description  
                "The application type choices.";  
            container ethernet {  
                description  
                    "Ethernet or TSN traffic that maps to an  
                    interface.";  
                leaf-list interface {  
                    type if:interface-ref;  
                    description  
                        "One or more Ethernet or TSN interfaces.  
                        If multiple interfaces are specified, this  
                        application flow is replicated to those  
                        interfaces. DetNet application flow filtering  
                        applies to the whole list of interfaces.  
                        For fine-grained flow filtering, use a single  
                        interface per application.";  
                }  
            }  
            container ip-mpls {  
                description  
                    "IP or MPLS DetNet application types.";  
                uses detnet-app-next-hop-content;  
            }  
        }  
    }  
    }  
    }  
    }  
    }  
    }  
    container service {
```

```
description
  "The DetNet service sub-layer configuration.";
list sub-layer {
  key "name";
  description
    "Services are indexed by name.";
  leaf name {
    type string;
    description
      "The name of the DetNet service sub-layer.";
  }
  leaf service-rank {
    type uint8;
    default "255";
    description
      "The DetNet rank for this service. Defaults to '255'
      (lowest rank) if not specified.";
    reference
      "RFC 9016: Flow and Service Information Model for
      Deterministic Networking (DetNet), Section 5.7";
  }
  leaf traffic-profile {
    type traffic-profile-ref;
    description
      "The traffic profile for this service.";
  }
  container service-protection {
    description
      "The service protection type and sequence number
      options.";
    leaf protection {
      type service-protection;
      description
        "The DetNet service protection type, such as
        the Packet Replication Function (PRF), the
        Packet Elimination Function (PEF), or the
        Packet Replication, Elimination, and Ordering
        Functions (PREOF).";
      reference
        "RFC 8938: Deterministic Networking (DetNet)
        Data Plane Framework, Section 4.3";
    }
    leaf sequence-number-length {
      type sequence-number-field;
      default "zero-sn";
      description
        "The sequence number field length can be one of
        0 (none), 16 bits, or 28 bits. The default is
        0 (none).";
    }
  }
  leaf operation {
    type operation;
    description
      "This is the service operation type for this service
      sub-layer.";
  }
  container incoming {
```



```

    "The outgoing type may be a forwarding sub-layer, a
    service sub-layer, or an aggregation type.";
  container forwarding-sub-layer {
    description
      "This service sub-layer is sending to the
      forwarding sub-layer of the lower layer
      for DetNet service forwarding or
      service-to-forwarding aggregation at the
      ingress node or relay node. When the
      operation type is 'initiation', the
      service sub-layer encapsulates the DetNet
      Control Word (d-CW) and S-Label, which are for
      individual DetNet flows when the incoming type
      is 'app-flow' and for an aggregated DetNet flow
      when the incoming type is 'service' or
      'forwarding'. The service sub-layer swaps the
      service label when the operation type is
      'relay'.";
    reference
      "RFC 8964: Deterministic Networking (DetNet)
      Data Plane: MPLS, Sections 4.2.1 and 4.2.2";
    list service-outgoing {
      key "index";
      description
        "List of outgoing service sub-layers aggregated
        in the forwarding sub-layer.";
      leaf index {
        type uint8;
        description
          "This index allows a list of multiple outgoing
          forwarding sub-layers.";
      }
      uses detnet-header;
      uses forwarding-sub-layer-group;
    }
  }
  container service-sub-layer {
    description
      "This service sub-layer is sending to the
      service sub-layer of the lower layer for
      service-to-service aggregation at the
      ingress node or relay node. The service
      sub-layer encapsulates the d-CW and S-Label when
      the operation type is 'initiation' and
      swaps the S-Label when the operation type is
      'relay'.";
    reference
      "RFC 8964: Deterministic Networking (DetNet)
      Data Plane: MPLS, Sections 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

```



```
key "name";
description
  "List of one or more DetNet service/forwarding
  types.";
leaf name {
  type string;
  description
    "The name of the DetNet forwarding sub-layer.";
}
leaf traffic-profile {
  type traffic-profile-ref;
  description
    "The traffic profile for this group.";
}
leaf operation {
  type mpls-fwd-operation;
  description
    "The forwarding operation types
    'impose-and-forward', 'pop-and-forward',
    'pop-impose-and-forward', 'forward', and
    'pop-and-lookup'.";
}
container incoming {
  description
    "The DetNet forwarding sub-layer incoming
    configuration.";
  choice incoming {
    description
      "Choices of incoming types.";
    container service-sub-layer {
      description
        "This forwarding sub-layer is related to the
        service sub-layer of the upper layer and
        provides DetNet forwarding or
        service-to-forwarding aggregation at
        the ingress node or relay node.";
      uses service-sub-layer-group;
    }
    container forwarding-aggregation {
      description
        "This forwarding sub-layer is related to the
        forwarding sub-layer of the upper layer and
        provides forwarding-to-forwarding aggregation at
        the ingress node, relay node, or transit node.";
      uses forwarding-sub-layer-group;
    }
  }
  container forwarding-id {
    description
      "This forwarding sub-layer is related to all of
      the lower layers and provides DetNet forwarding
      swap or termination at the transit node,
      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 {
    description
      "A service is connected directly to an
      interface with no forwarding sub-layer.";
    container interface {
      description
        "This forwarding sub-layer is sending to the
        interface, for sending to the next hop at the
        ingress node, relay node, or transit node.";
      uses detnet-forwarding-next-hop-content;
    }
    container service-aggregation {
      description
        "This forwarding sub-layer is sending 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 sending to the
        forwarding sub-layer of the lower layer for
        forwarding-to-forwarding aggregation at the ingress
        node, relay node, or transit node.";
      leaf aggregation-sub-layer {
        type forwarding-sub-layer-ref;
        description
          "This is a reference to the forwarding sub-layer.";
      }
      container forwarding-label {
        description
          "This is the forwarding label for forwarding
          sub-layer aggregation.";
        uses rt-types:mpls-label-stack;
      }
    }
    container service-sub-layer {
      description
        "This forwarding sub-layer is sending to the
```


10. Security Considerations

Security considerations for DetNet are covered in "[Deterministic Networking Architecture](#)" [RFC8655] and "[Deterministic Networking \(DetNet\) Security Considerations](#)" [RFC9055].

The YANG module specified in this document defines 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 this YANG 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 have a negative effect on network operations. Unauthorized write operations (e.g., edit-config) to any elements of this module can break or incorrectly connect DetNet flows. Since DetNet is a configured data plane, any changes that are not coordinated with all devices along the path will result in a denial of service. In addition, arbitrary write operations could enable an attacker to modify a network path to enable select traffic to avoid inspection or treatment by security controls or to route traffic in such a way that the traffic would be subject to inspection/modification by an adversary node.

Some of the readable data nodes in this YANG module 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 nodes and their sensitivity/vulnerability:

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

/detnet/traffic-profile/member-app-flow: This links traffic profiles to applications, service sub-layers, and/or forwarding sub-layers, so this could also be considered more 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.

The above nodes can reveal identifiable characteristics of the application flows.

/detnet/service/sub-layer: This defines the service and forwarding operations.

/detnet/forwarding/sub-layer: This defines the forwarding operations.

The above nodes can reveal some aspects of the network topology in the case of unauthorized access to this configuration.

11. References

11.1. 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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
 - [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
 - [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>>.

11.2. Informative References

- [IEEE8021Q-2022] IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks", DOI 10.1109/IEEESTD.2022.10004498, IEEE Std 802.1Q-2022, December 2022, <<https://ieeexplore.ieee.org/document/10004498>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [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>>.
- [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>>.

Appendix A. DetNet Configuration YANG Tree

This is the full YANG tree per the guidelines provided 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? decimal64
      | | +--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-flow* app-flow-ref
      | +--ro member-svc-sublayer* 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 (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 (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
+--:(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 next-hop-address?
|   |   inet:ip-address

```

```

|                                     +---:(mpls)
|                                     +---rw mpls-label-stack
|                                     +---rw entry* [id]
|                                     |---rw id
|                                     |   uint8
|                                     +---rw label?
|                                     |   rt-types:mpls-
|                                     |   label
|                                     +---rw ttl?
|                                     |   uint8
|                                     +---rw traffic-class?
|                                     |   uint8
+---rw service
|   +---rw sub-layer* [name]
|   |   +---rw name                string
|   |   +---rw service-rank?       uint8
|   |   +---rw traffic-profile?    traffic-profile-ref
|   |   +---rw service-protection
|   |   |   +---rw protection?      service-protection
|   |   |   +---rw sequence-number-length? sequence-number-field
|   |   +---rw operation?          operation
|   |   +---rw incoming
|   |   |   +---rw (incoming)?
|   |   |   |   +---:(app-flow)
|   |   |   |   |   +---rw app-flow
|   |   |   |   |   |   +---rw flow* app-flow-ref
|   |   |   |   +---:(service-aggregation)
|   |   |   |   |   +---rw service-aggregation
|   |   |   |   |   |   +---rw sub-layer* service-sub-layer-ref
|   |   |   |   +---:(forwarding-aggregation)
|   |   |   |   |   +---rw forwarding-aggregation
|   |   |   |   |   |   +---rw sub-layer* forwarding-sub-layer-ref
|   |   |   |   +---:(service-id)
|   |   |   |   |   +---rw service-id
|   |   |   |   |   |   +---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 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?          mpls-fwd-operation
    +---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 next-hop-address?
| | | | | inet:ip-address
| | | | +--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 next-hop-address?
|       |   inet:ip-address
|   +--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

This section provides several examples. These examples were tested with the "yanglint" program and use operational output to exercise both "config true" and "config false" objects. Note that IPv4 and IPv6 addresses are supported, but for clarity, IPv4 is used, with the exception of Example A-1 ([Appendix B.1](#)). The IP types are imported from [\[RFC6991\]](#); these types support both IPv4 and IPv6.

The following conventions are used in the diagrams.

- In the diagrams found in the PDF and HTML copies of this document, replication and elimination points are shown as "R" and "E" in circles, respectively.
- Packet headers, including a DetNet aggregation label (A-Label), service label (S-Label), and forwarding label (F-Label), are illustrated at each hop as defined in [\[RFC8964\]](#).
- Aggregation/disaggregation nodes are indicated by dashed-line boxes.
- Since the model augments IETF interfaces, minimal interface YANG data is provided to validate the interface data as well. This shows up as a named value, such as "eth0", that is referenced by the configuration.

Below are examples of aggregation and disaggregation at various points in DetNet. Where indicated, figures are provided in the PDF and HTML copies of this document.

B.1. Example A-1: Application Flow Aggregation

This example illustrates multiple App-flows with the same source, destination, and traffic specification aggregated into a single DetNet flow service sub-layer. Ingress node 1 aggregates App-flows 0 and 1 into a service sub-layer of DetNet flow 1. Two ways to illustrate this are provided in [Figures 1](#) and [2](#); the JSON operational data model [\[RFC8259\]](#) corresponding to the diagrams is then shown in [Figure 3](#). The address format used in this example is IPv6.

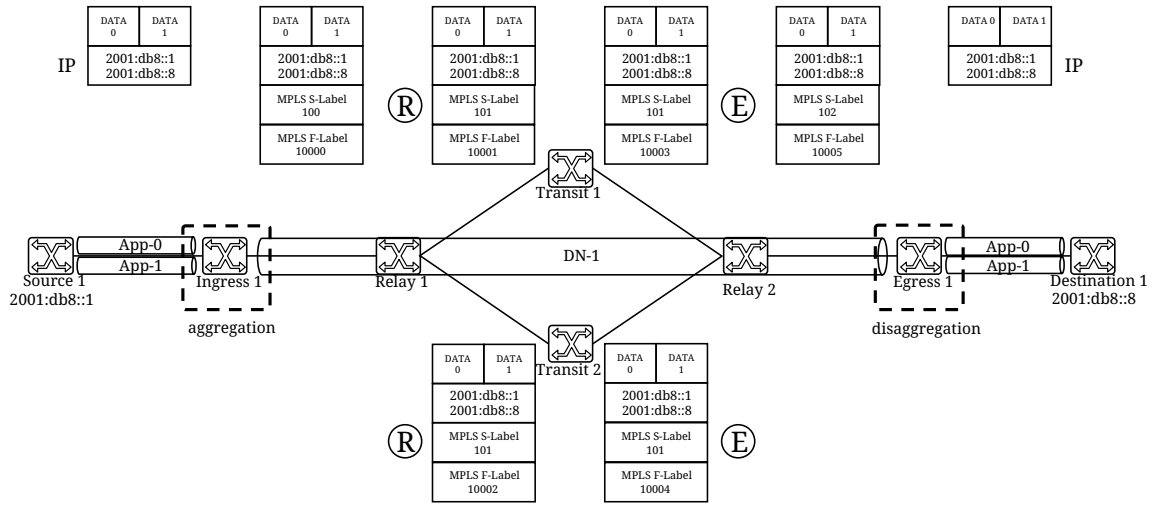


Figure 1: Case A-1: Application Flow Aggregation

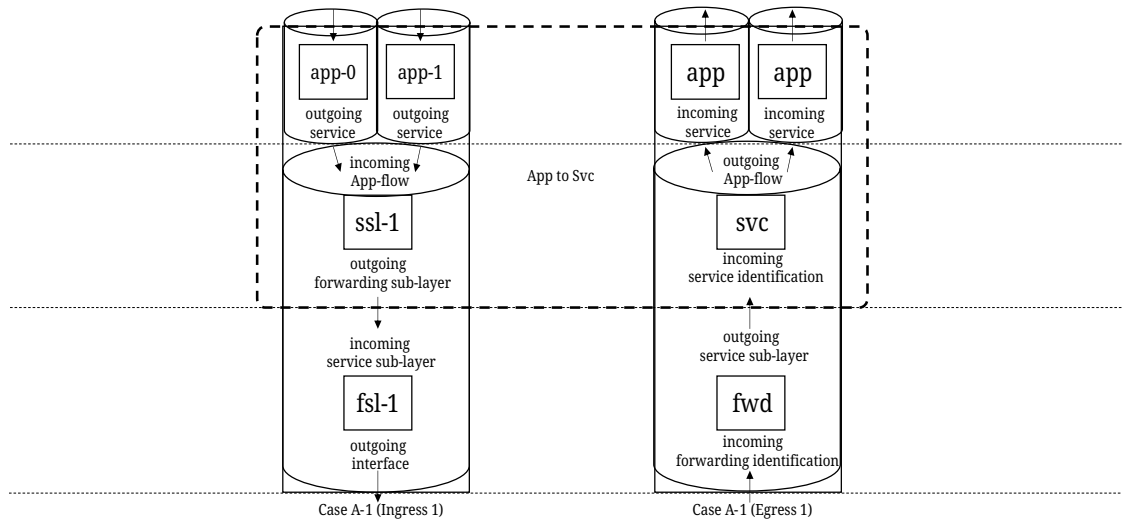


Figure 2: Case A-1: Stack Details for Application Flow Aggregation

Figure 3 contains the operational JSON configuration for the ingress aggregation node illustrated in Figures 1 and 2. "app-0" and "app-1" are aggregated into service sub-layer ssl-1.

```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 20000000,
          "max-loss": "0.0000001",
          "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-flow": [
          "app-0",
          "app-1"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 20000000,
          "max-loss": "0.000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 20,
          "max-payload-size": 1500,
          "min-payload-size": 100,
          "min-pkts-per-interval": 1
        },
        "member-svc-sublayer": [
          "ssl-1"
        ]
      },
      {
        "name": "pf-3",
        "traffic-spec": {
          "interval": 5,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500
        },
        "member-fwd-sublayers": [
          "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": "2024-02-21T18:59:00-05:00"
    }
  },
  {
    "name": "eth2",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2024-02-21T18:59:00-05:00"
    }
  }
]
}
```

Figure 3: Example A-1: DetNet Configuration Application Flow Aggregation

B.2. Example B-1: Aggregation Using a Forwarding Sub-layer

As illustrated in [Figure 4](#), DetNet service sub-layer flows 1 and 2 are aggregated into a single forwarding sub-layer. For the same destination, multiple DetNet flows use a single forwarding path, and service protection is performed by the corresponding service sub-layer of each flow. The corresponding XML operational data for node "Ingress 1" follows.

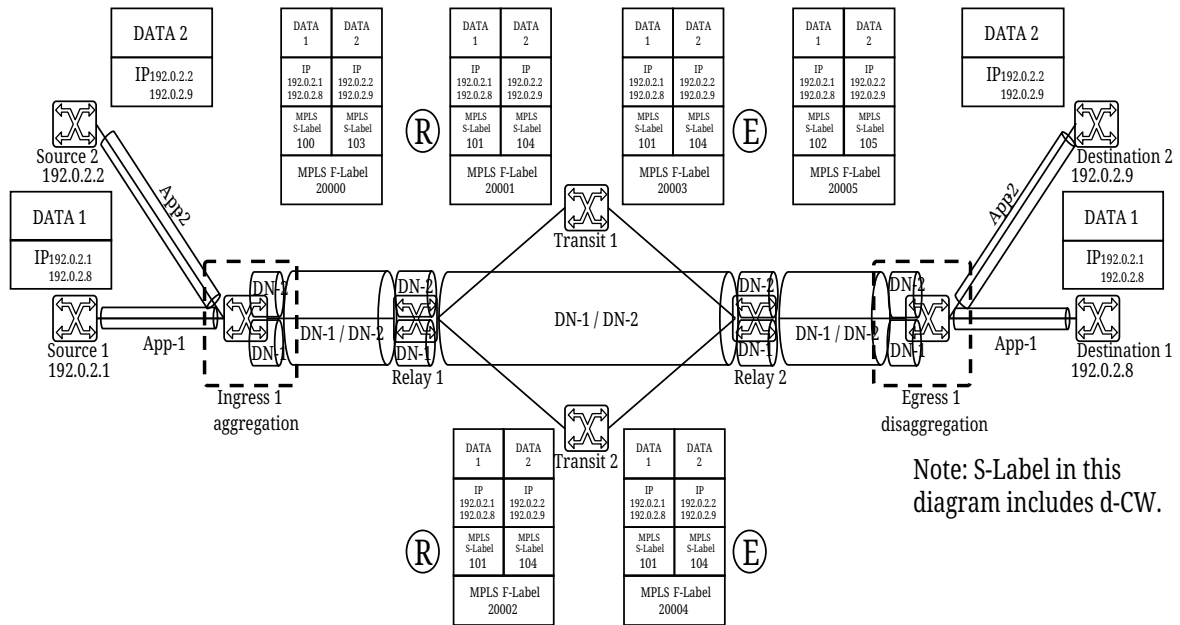


Figure 4: Case B-1: Aggregation Using a Forwarding Sub-layer

Figure 5 contains the operational XML configuration for the ingress aggregation node illustrated in Figure 4. In this example, "app-0" and "app-1" are in separate service sub-layers with MPLS labels, and the aggregation happens at forwarding sub-layer afl-1, using MPLS labels.

```
<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>2024-02-21T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth1</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time>2024-02-21T23:59:00Z</discontinuity-time>
    </statistics>
  </interface>
  <interface>
    <name>eth2</name>
    <type>ia:ethernetCsmacd</type>
    <oper-status>up</oper-status>
    <statistics>
      <discontinuity-time>2024-02-21T23:59:00Z</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>
      <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>
        </ip-app-flow>
      </ingress>
    </app-flow>
  </app-flows>
</detnet>
```

```

        <dscp>7</dscp>
      </ip-app-flow>
    </ingress>
  </app-flow>
</app-flows>
<traffic-profile>
  <name>1</name>
  <traffic-requirements>
    <min-bandwidth>100000000</min-bandwidth>
    <max-latency>100000000</max-latency>
    <max-latency-variation>20000000</max-latency-variation>
    <max-loss>0.0000001</max-loss>
    <max-consecutive-loss-tolerance>5
      </max-consecutive-loss-tolerance>
    <max-misordering>0</max-misordering>
  </traffic-requirements>
  <traffic-spec>
    <interval>5</interval>
    <max-pkts-per-interval>10</max-pkts-per-interval>
    <max-payload-size>1500</max-payload-size>
  </traffic-spec>
  <member-app-flow>app-1</member-app-flow>
  <member-app-flow>app-2</member-app-flow>
</traffic-profile>
<traffic-profile>
  <name>2</name>
  <traffic-requirements>
    <min-bandwidth>100000000</min-bandwidth>
    <max-latency>100000000</max-latency>
    <max-latency-variation>20000000</max-latency-variation>
    <max-loss>0.0000001</max-loss>
    <max-consecutive-loss-tolerance>5
      </max-consecutive-loss-tolerance>
    <max-misordering>0</max-misordering>
  </traffic-requirements>
  <member-svc-sublayer>ssl-1</member-svc-sublayer>
  <member-svc-sublayer>ssl-2</member-svc-sublayer>
</traffic-profile>
<traffic-profile>
  <name>3</name>
  <traffic-spec>
    <interval>5</interval>
    <max-pkts-per-interval>20</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>
        <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>
</sub-layer>
</forwarding>
</sub-layer>
</service>
</incoming>
</app-flow>
</outgoing>
</sub-layer>
</app-flow>
</incoming>
```

```
    </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 5: Example B-1: DetNet Configuration Forwarding Sub-layer Aggregation

B.3. Example B-2: Service Aggregation

As illustrated in [Figure 6](#), DetNet service sub-layer flows 1 and 2 are aggregated into a service sub-layer of an aggregated flow. Multiple DetNet flows with the same requirements for the same destination are aggregated into a single aggregated DetNet flow, and service protection and resource allocation are performed by an aggregated DetNet flow service sub-layer and forwarding sub-layer. The corresponding JSON operational data for node "Ingress 1" follows.

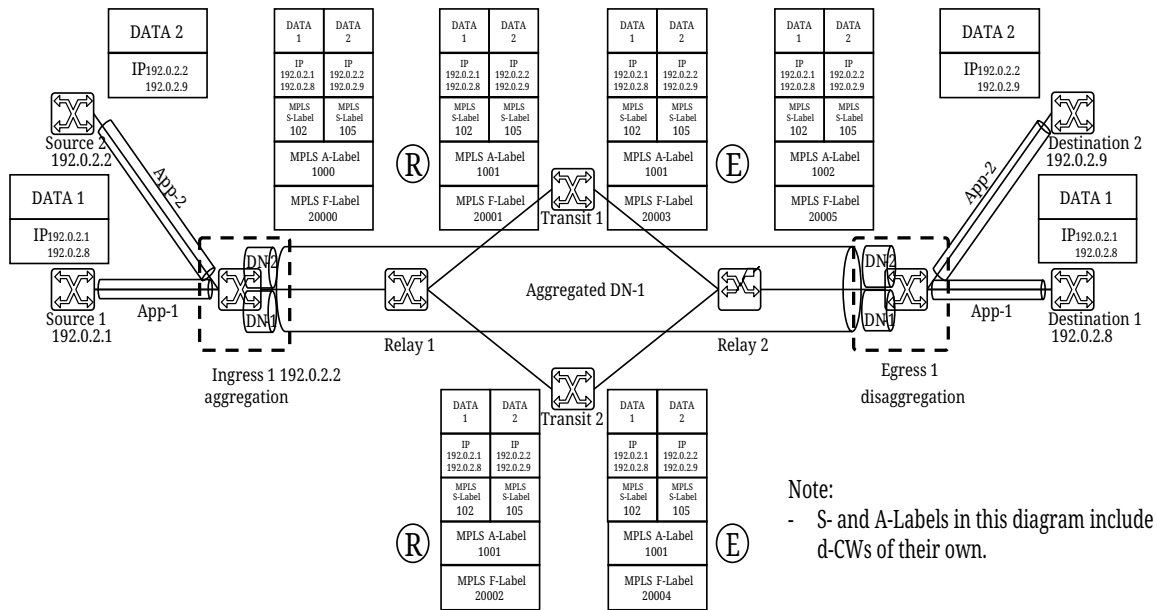


Figure 6: Case B-2: Example Service Aggregation

Figure 7 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 6. In this example, service sub-layer ssl-1 for DetNet flow DN-1 and ssl-2 for DetNet flow DN-2 aggregate at service sub-layer DetNet flow asl-1. In this example, an aggregation service sub-layer, asl-1, is created to aggregate ssl-1 and ssl2, and that label is encapsulated in a separate forwarding sub-layer, afl-1, with MPLS labels.

```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 20000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 10,
          "max-payload-size": 1500
        },
        "member-app-flow": [
          "app-1",
          "app-2"
        ]
      },
      {
        "name": "2",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 20000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "3",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 20000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "asl-1"
        ]
      },
      {
        "name": "4",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 20,

```

```

    "max-payload-size": 1500
  },
  "member-fwd-sublayer": [
    "af1-1"
  ]
}
],
"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": "3",
    "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": "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
                            }
                        ]
                    }
                }
            }
        }
    ]
}

```

```
    }
  }
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 7: Example B-2: DetNet Service Aggregation

B.4. Example C-1: DetNet Relay Service Sub-layer

Figure 8 illustrates the DetNet relay node's forwarding sub-layer flows 1 and 2 aggregated into a single forwarding sub-layer. Service protection and resource allocation are performed by the corresponding service sub-layer and forwarding sub-layer of each flow. Figure 8 illustrates both aggregation and disaggregation, and the corresponding JSON operational data follows.

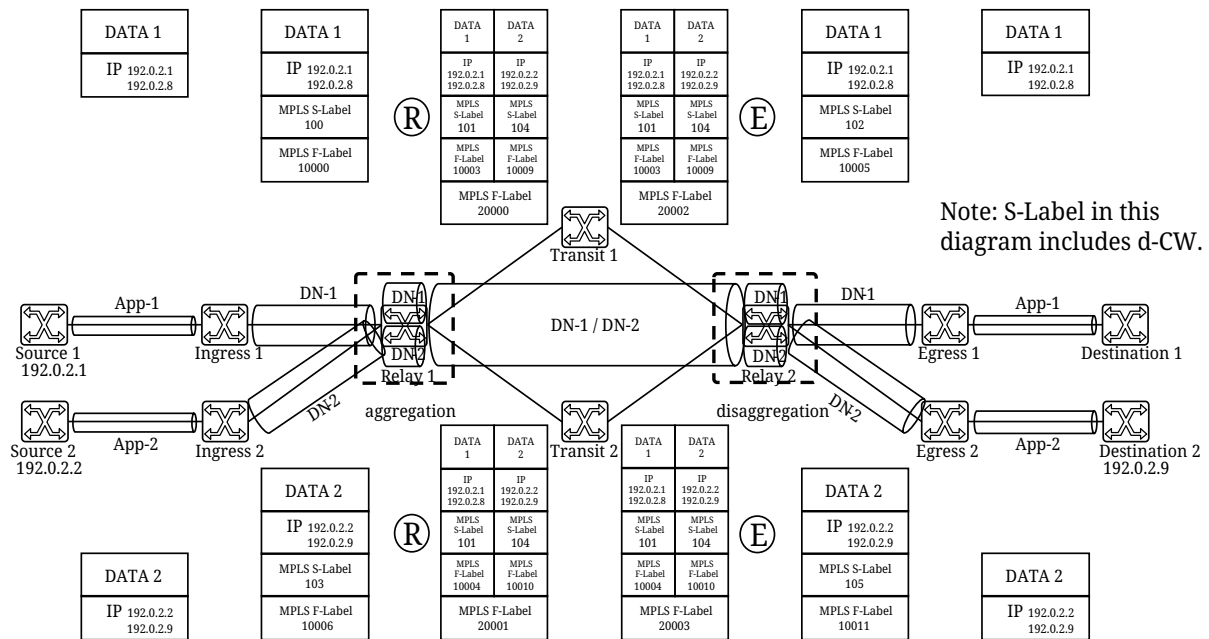


Figure 8: Case C-1: Example Service Aggregation/Disaggregation

Figure 9 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 8. In this example, a relay performing aggregation at the forwarding sub-layer is illustrated. Two DetNet flows -- DN-1 and DN-2 -- are replicated at each service sub-layer. The two forwarding sub-layers for the upper path are aggregated at the forwarding sub-layer with label 20000, and the two forwarding sub-layers for the lower path are aggregated at the forwarding sub-layer with label 20001. Figure 10 contains the operational JSON configuration for the egress disaggregation node illustrated in Figure 8.

```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "af1-1",
          "af1-2"
        ]
      },
      {
        "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": "af1-1",
      "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": "af1-2",
      "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": "af1-1",
      "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": "af1-2",
      "forwarding-label": {
        "mpls-label-stack": {
          "entry": [
            {
              "id": 0,

```

```
        "label": 10010
      }
    ]
  }
}
},
{
  "name": "af1-1",
  "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": "af1-2",
  "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": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 9: Example C-1: DetNet 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": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-spec": {
          "interval": 125,
          "max-pkts-per-interval": 2,
          "max-payload-size": 1518
        },
        "member-fwd-sublayer": [
          "af1-1",
          "af1-2"
        ]
      },
      {
        "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": "af1-1",
            "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": "af1-2",
        "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": "2024-02-21T18:59:00-05:00"
    }
  },
  {
    "name": "eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2024-02-21T18:59:00-05:00"
    }
  },
  {
    "name": "eth2",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2024-02-21T18:59:00-05:00"
    }
  },
  {
    "name": "eth3",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2024-02-21T18:59:00-05:00"
    }
  }
]
}
```

Figure 10: Example C-1: DetNet Relay Service Disaggregation

B.5. Example C-2: DetNet Relay Service Sub-layer Aggregation/Disaggregation

Figure 11 illustrates the DetNet relay node's service sub-layer flows 1 and 2 aggregated into a single forwarding sub-layer. Service protection is performed by the corresponding service sub-layer of each flow, and resource allocation is performed by an aggregated forwarding sub-layer for all aggregated flows. Figure 11 illustrates both aggregation and disaggregation, and the corresponding JSON operational data follows.

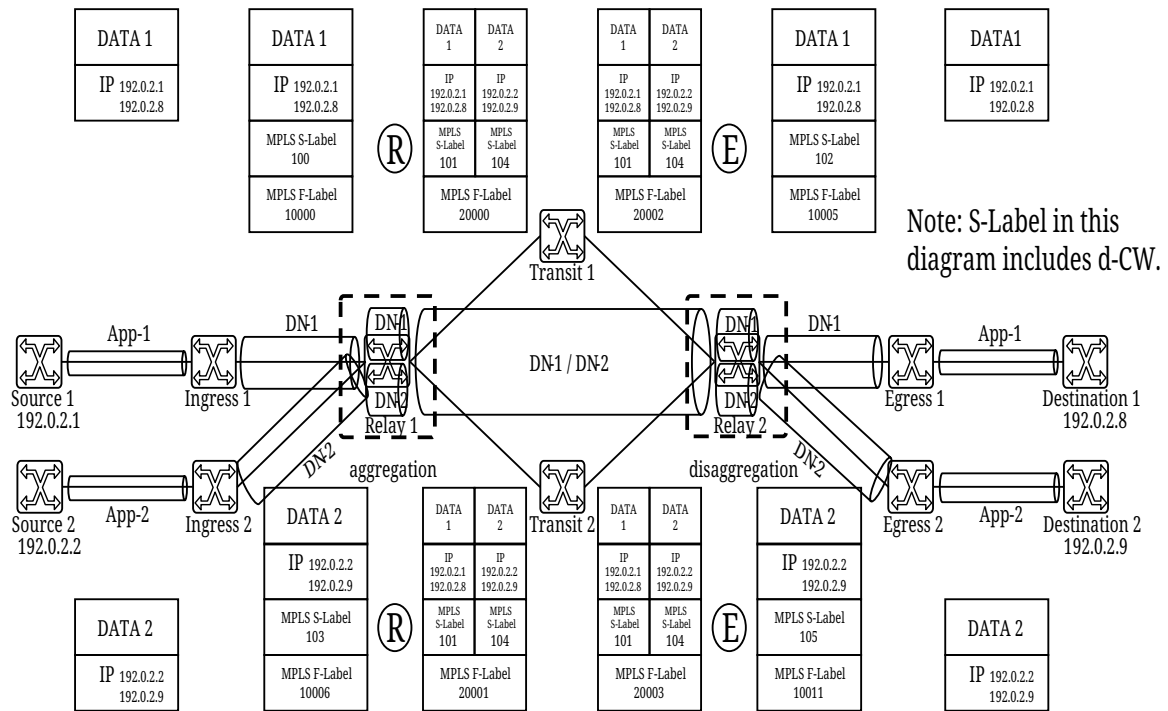


Figure 11: Case C-2: Example Service Aggregation/Disaggregation

Figure 12 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 11. In this example, a relay performing aggregation at the forwarding sub-layer is illustrated. Two DetNet flows -- DN-1 and DN-2 -- are replicated at each service sub-layer. Each replicated flow for the service sub-layer for the upper path is aggregated at the single forwarding sub-layer with MPLS label 20000, and each replicated flow for the service sub-layer for the lower path is aggregated at the forwarding sub-layer with MPLS label 20001. Figure 13 contains the operational JSON configuration for the egress disaggregation node illustrated in Figure 11.


```

{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": "af1-1",
  "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": "af1-2",
  "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-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}

```

Figure 12: Example C-2: DetNet Relay Aggregation Service Sub-layer

```

{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": "af1-1",
            "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": "af1-2",
        "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": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}
}

```

Figure 13: Example C-2: DetNet Relay Disaggregation Service Sub-layer

B.6. Example C-3: DetNet Relay Service Sub-layer Aggregation/Disaggregation

Figure 14 illustrates the DetNet relay node's service sub-layer flows 1 and 2 aggregated into a service sub-layer flow. Multiple DetNet flows with the same requirements that can use the same path are aggregated into a single aggregated DetNet flow, and service protection and resource

allocation are performed by the service sub-layer and forwarding sub-layer of the aggregated DetNet flow. Figure 14 illustrates both aggregation and disaggregation, and the corresponding JSON operational data follows.

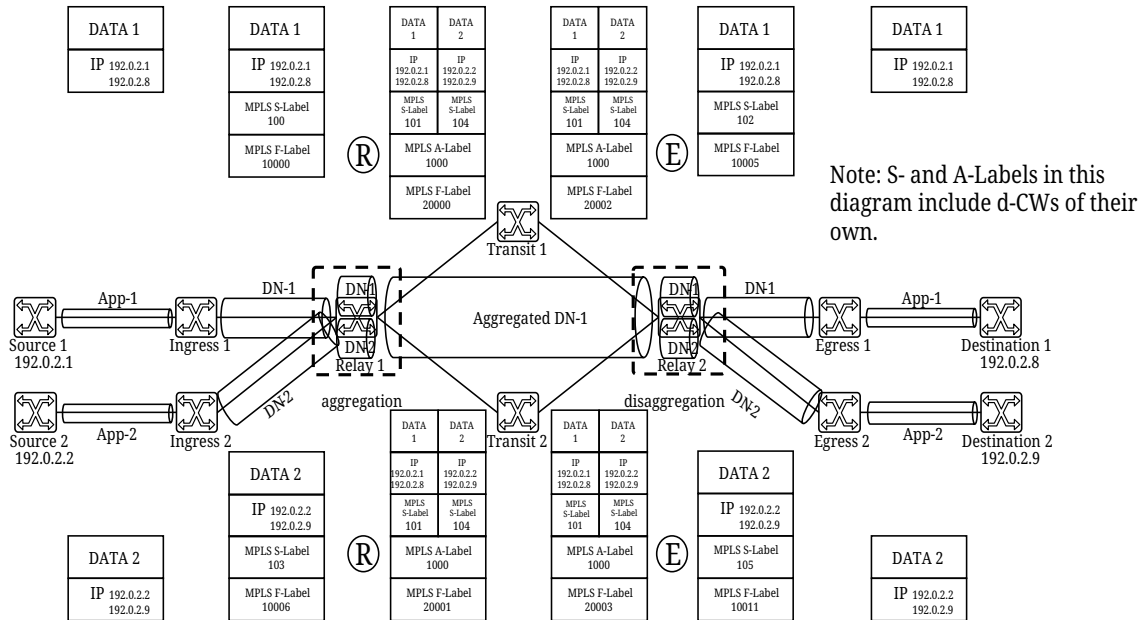


Figure 14: Case C-3: Example Service Aggregation/Disaggregation

Figure 15 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 14. In this example, a relay performing aggregation at the service sub-layer is illustrated. Two DetNet flows -- DN-1 and DN-2 -- are relayed at each service sub-layer with MPLS labels 101 and 104, respectively, and each service sub-layer is aggregated at a single service sub-layer flow and replicated. Figure 16 contains the operational JSON configuration for the egress disaggregation node illustrated in Figure 14.

```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
},

```



```
{
  "name": "eth1",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
}
]
}
```

Figure 15: Example C-3: DetNet 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": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 100000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}

```

```
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
```

Figure 16: Example C-3: DetNet Relay Service Sub-layer Disaggregation

B.7. Example C-4: DetNet Relay Service Sub-layer Aggregation/Disaggregation

Figure 17 illustrates the DetNet relay node's forwarding sub-layer flows 1 and 2 aggregated into a service sub-layer DetNet flow. Multiple DetNet flows with the same requirements that can use the same path are aggregated into a single aggregated DetNet flow. Service protection is performed by the service sub-layer of the aggregated DetNet flow, and resource allocation is performed by the forwarding sub-layer of each aggregated DetNet flow. Figure 17 illustrates both aggregation and disaggregation, and the corresponding JSON operational data follows.

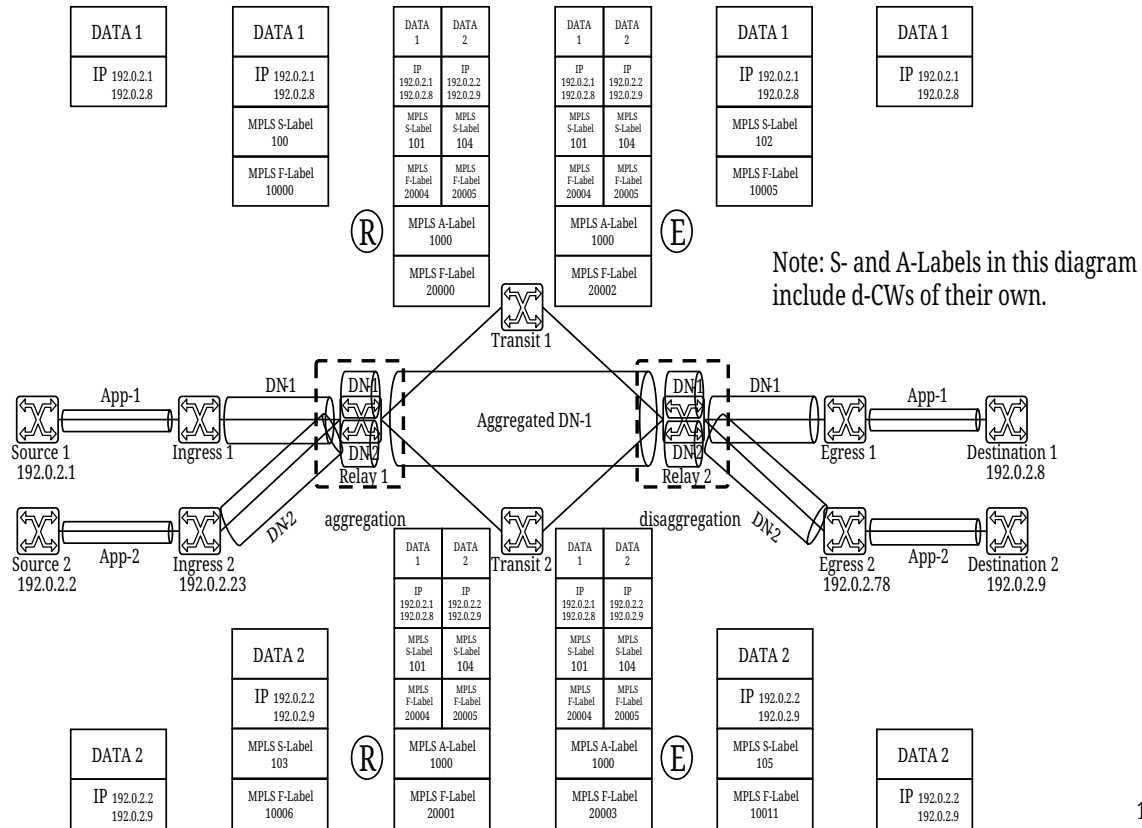


Figure 17: Case C-4: Example Service Aggregation/Disaggregation

Figure 18 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 17. In this example, a relay performing aggregation at the service sub-layer is illustrated. Two DetNet flows -- DN-1 and DN-2 -- are relayed at each service sub-layer. The two DetNet forwarding sub-layer flows with MPLS labels 20004 and 20005 are aggregated at the single service sub-layer DetNet flow and then replicated. Figure 19 contains the operational JSON configuration for the egress disaggregation node illustrated in Figure 17.


```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth1",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
}
]
}
```

Figure 18: Example C-4: DetNet Relay Service Sub-layer Aggregation

```
{
  "ietf-detnet:detnet": {
    "traffic-profile": [
      {
        "name": "pf-1",
        "traffic-requirements": {
          "min-bandwidth": "100000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "ssl-1",
          "ssl-2"
        ]
      },
      {
        "name": "pf-2",
        "traffic-requirements": {
          "min-bandwidth": "200000000",
          "max-latency": 100000000,
          "max-latency-variation": 10000000,
          "max-loss": "0.0000001",
          "max-consecutive-loss-tolerance": 5,
          "max-misordering": 0
        },
        "member-svc-sublayer": [
          "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": {

```

```

        "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"
                    ]
                }
            ]
        },
        {
            "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": {
                "forwarding-disaggregation": {
                    "sub-layer": [
                        "fsl-3",
                        "fsl-4"
                    ]
                }
            }
        }
    },
    {

```

```

    }
  ]
},
"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": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {

```

```
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
},
{
  "name": "eth3",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "statistics": {
    "discontinuity-time": "2024-02-21T18:59:00-05:00"
  }
}
]
}
```

Figure 19: Example C-4: DetNet Relay Service Sub-layer Disaggregation

B.8. Example D-1: Transit Node Forwarding Sub-layer Aggregation/Disaggregation

As illustrated in [Figure 20](#), at the transit node, forwarding sub-layer flows 1 and 2 are aggregated into a single forwarding sub-layer. Resource allocation is performed by the corresponding forwarding sub-layer for all aggregated flows. [Figure 20](#) illustrates both aggregation and disaggregation, and the corresponding JSON operational data follows.

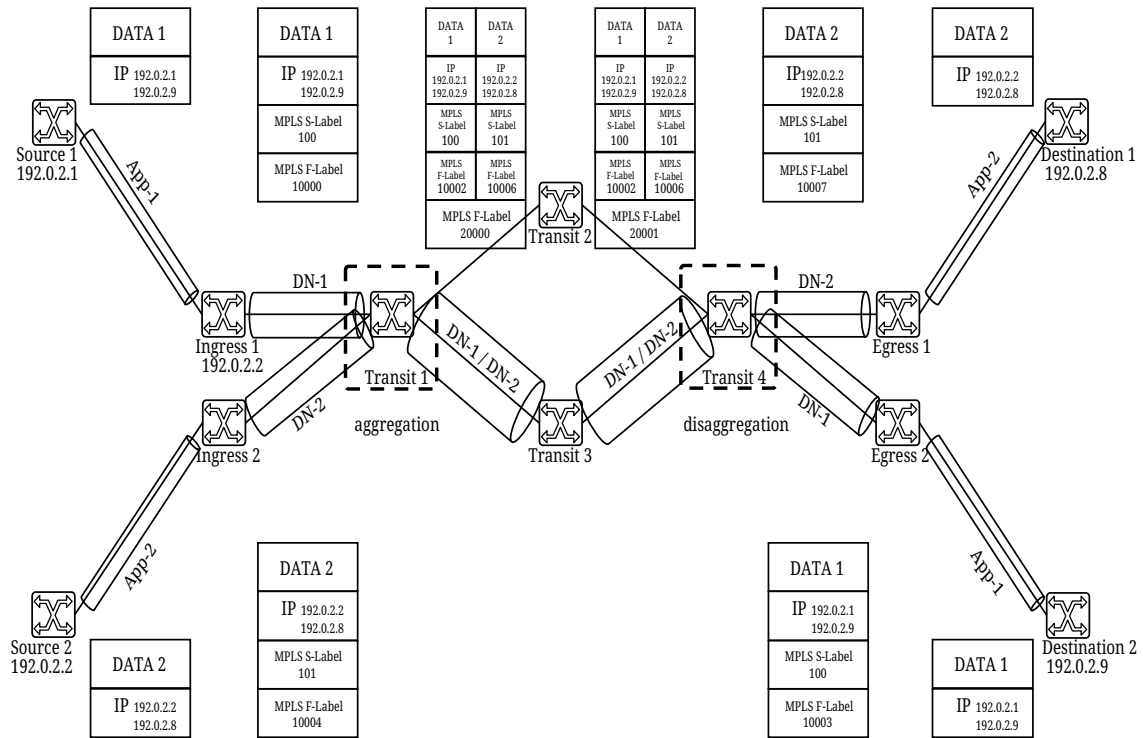


Figure 20: Case D-1: Example Transit Node Forwarding Aggregation/Disaggregation

Figure 21 contains the operational JSON configuration for the ingress aggregation node illustrated in Figure 20. In this example, a transit node performing aggregation at the forwarding sub-layer is illustrated. Two DetNet flows -- DN-1 and DN-2 -- are transmitted at each forwarding sub-layer. The DetNet forwarding sub-layer flows with MPLS labels 10002 and 10006 are aggregated at the single forwarding sub-layer. The resulting aggregated DetNet flow has MPLS label 20000. Figure 22 contains the operational JSON configuration for the egress disaggregation transit node illustrated in Figure 20.

```

{
  "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": [
          "af1-1"
        ]
      }
    ],
    "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": "af1-1",
              "forwarding-label": {
                "mpls-label-stack": {
                  "entry": [
                    {
                      "id": 0,
                      "label": 10002
                    }
                  ]
                }
              }
            }
          }
        ]
      }
    ]
  }
}

```



```

    }
  ]
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}

```

Figure 21: Example D-1: Transit Node Forwarding 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": [
          "af1-1"
        ]
      }
    ],
    "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": "af1-1",
    "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": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth2",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    },
    {
      "name": "eth3",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2024-02-21T18:59:00-05:00"
      }
    }
  ]
}
}
```

Figure 22: Example D-1: Transit Node Forwarding Disaggregation

Acknowledgments

The authors of this document would like to thank Lou Berger, Tom Petch, Xufeng Liu, Julien Meuric, John Scudder, and Florian Kauer for their detailed comments.

Contributors

The authors of this document wish to thank and acknowledge the following individual, who contributed substantially to the content of this document and should be considered a coauthor:

Mach(Guoyi) Chen

Huawei Technologies

Email: mach.chen@huawei.com

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**

Equinix

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

China Mobile

Email: lizhenqiang@chinamobile.com