# Telemetry Report Format Specification (Draft)

November 10, 2017

Mickey Spiegel, Jeongkeun Lee: *Barefoot Networks*
Gordon Brebner: *Xilinx*
Mukesh Hira: *VMware*

# Introduction

Traditional network monitoring has relied on statistics and probe packets such as ICMP echo requests/replies. Recent innovations provide greater insight into network behavior by generating detailed reports of telemetry metadata such as paths, queue occupancy, latency experienced by data packets, and timestamps that can be used to determine hop-by-hop and end-to-end delay. Generation of telemetry reports can be triggered by various events in categories such as flow monitoring, queue congestion, and packet drops. For further information regarding the motivation and usage of detailed telemetry information, see [iOAM_reqs].

Specifications are being defined for embedding telemetry metadata within data packets, such as [INT] and [iOAM]. This allows for telemetry metadata to be collected as packets traverse a network. When the packets reach the edge of the network, the telemetry metadata is removed and telemetry reports are generated.

This specification defines packet formats for telemetry reports from data plane network devices (e.g. switches) to a distributed telemetry monitoring system. The packet formats use headers that describe the contents of telemetry reports, along with existing (non-telemetry specific) packet headers that can be used to categorize flows.

## Scope

The scope of this specification is interoperability between network devices that generate telemetry reports based on what they see in the data plane, and the initial preprocessors within distributed telemetry monitoring systems that receive the telemetry reports. This specification is applicable when telemetry reports are generated by network devices at the edges of a network, with source and transit network devices embedding telemetry metadata in data packets according to specifications such as [INT] and [iOAM]. This specification is also applicable when each network device directly generates telemetry reports (including transit network devices in the middle of the network), without affecting data packet formats between successive network devices.

Telemetry report encapsulation formats are defined that allow for the inclusion of additional telemetry metadata, beyond the (optional) telemetry metadata embedded between other packet headers as defined in [INT] and [iOAM]. The embedded telemetry metadata is included as is in telemetry reports, so the packet formats defined in [INT] and [IOAM] also define some aspects of the telemetry report format. See Embedded Telemetry Metadata for further discussion.

This specification does not address any of the following, which are considered out of scope:

- Configuration of network devices so that they can determine when to generate telemetry reports, and what information to include in those reports, such as [SAI_DTel].
- Events that trigger generation of telemetry reports.
- Selection of particular destinations within distributed telemetry monitoring systems, to which telemetry reports will be sent.
- Export format for flow statistics or summarized flow records such as [IPFIX].

# Key Concepts

## Telemetry Report Definition

We define a telemetry report as a message that a network device sends to the monitoring system. A telemetry report carries a snapshot of the original data packet (mostly the inner + outer headers), which triggered the reporting, together with additional telemetry metadata collected from the reporting network device, and possibly from its upstream network devices (in case of in-band mechanism like INT or IOAM). The report message is encapsulated by IP+UDP, hence it can be forwarded from the reporting network device through the data network, and to the destination monitoring system.

The following sections will cover the details on the report generation, report format and encapsulation.

## Telemetry Report Associations

There are many reasons why users may want telemetry reports to be generated. This specification currently considers three categories for telemetry report generation:

- **Tracked Flows**: Telemetry reports are generated matching certain flow definitions. A telemetry specific access control list (called a *flow watchlist* in this specification) determines which data packets to monitor by matching packet header fields and optionally identification of the ingress interface. (Note that the telemetry specific watchlist is not performing any access control. It only makes decisions related to monitoring actions.) The expectation is that telemetry reports can be generated for those packets that match the flow watchlist. The telemetry reports include information about the path

that packets traverse as well as other telemetry metadata such as hop latency and queue occupancy.
- **Dropped Packets**: Telemetry reports are generated for all dropped packets matching a telemetry specific access control list (called a *drop watchlist* in this specification). This provides visibility into the impact of packet drops on user traffic.
- **Congested Queues**: Telemetry reports are generated for traffic entering a specific queue during a period of queue congestion. This provides visibility into the traffic causing and prolonging queue congestion, for example a few large elephant flows that overwhelm a queue, as well as the victim traffic (mice flows) getting hurt by the congestion. This also enables the detection and “re-play” of a short microburst, caused by a large number of mice flows arriving at the queue at the same time.

Each telemetry report may be associated with one or more of these categories. This is indicated in the telemetry report by defining association bits, one for each category, as will be shown in the report format section. New categories (and corresponding association bits) may be added to future versions of this specification.

Network devices will need to be configured so that they can determine when to generate telemetry reports, and what information to include in those reports. Such configuration is considered to be beyond the scope of this specification. See [SAI_DTel] for one API proposal to enable data plane telemetry capabilities in network devices across all three categories.

## Telemetry Report Events

Telemetry reports are typically triggered by packet processing at a network device. However, even when processed packets match a watchlist for a telemetry report category, it is not necessary for each inspected packet to trigger generation of a telemetry report. Network devices may apply filters to determine when significant events occur that should be reported. This is called *event detection* in this specification. For example, a network device may trigger telemetry report generation whenever a packet matching a tracked application flow is received or transmitted on a different path than previous packets, or if a significant change in latency is experienced at one particular hop.

Determination of which packets trigger reports, in other words the specific conditions and logic to determine the events of interest, is left open for implementations to differentiate themselves, and is considered to be beyond the scope of this specification.

## Telemetry Modes

There are two different modes which differ with regard to the locations from which telemetry reports are generated.

## Postcard mode

In the *postcard* mode, each network device generates its own telemetry reports, as shown in Figure 1. The distributed telemetry monitoring system will receive reports from different network devices, each describing the telemetry metadata (such as switch IDs, port IDs, latency) for one hop. There is no change to data packets traversing the network. When using postcard mode, the telemetry metadata precedes the original packet headers within the telemetry report.

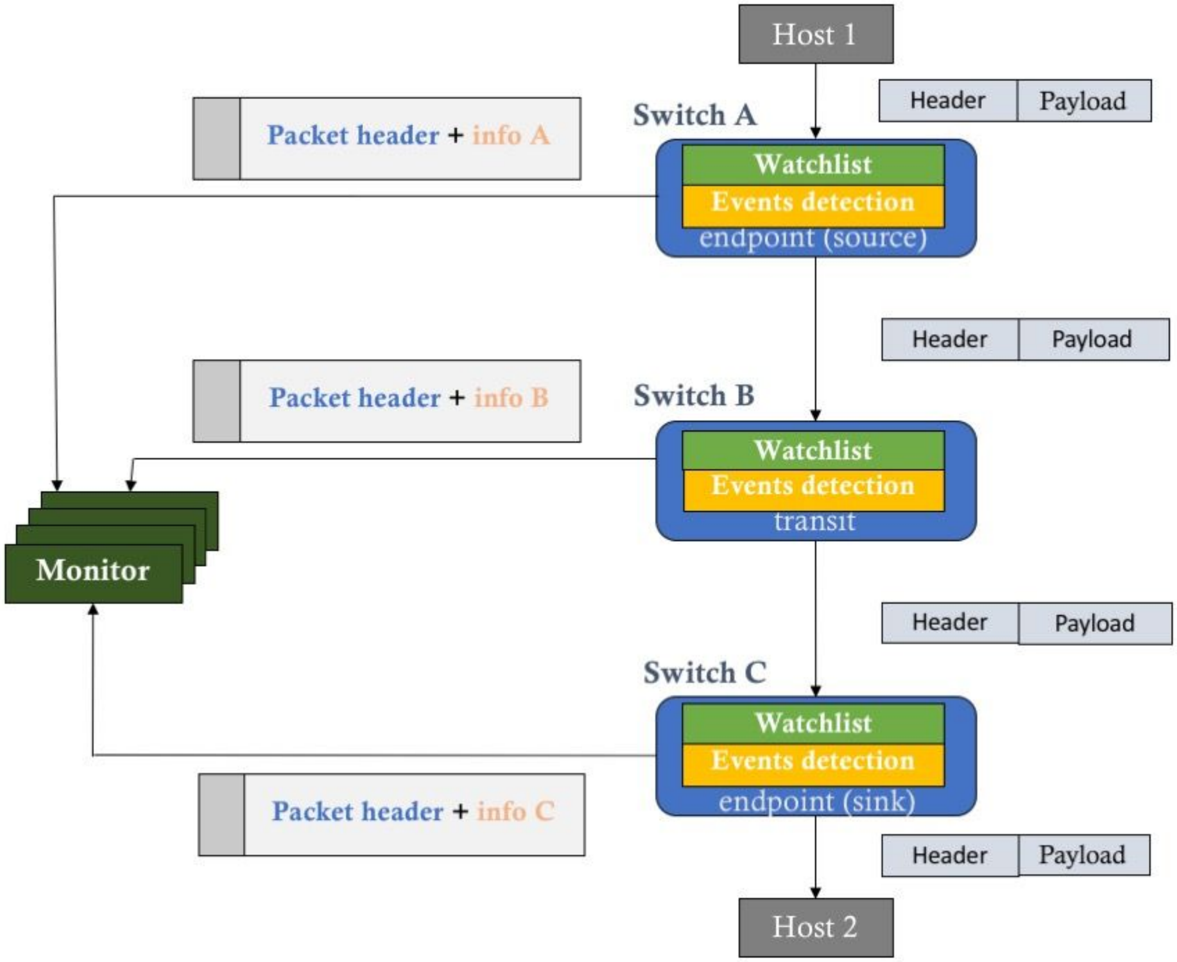

Figure 1: Telemetry Architecture with reports generated by all switches, aka *postcard*

## In-band (In-situ) Telemetry mode

In the other telemetry mode, telemetry metadata is embedded in between the original headers of data packets as they traverse the network, as shown in Figure 2. This may be done using any of the telemetry data plane specifications such as [INT] or [iOAM]. When a packet enters the network, the source switch may insert a telemetry instruction header, thereby instructing downstream switches to add the desired telemetry metadata. At each hop, the transit switch inserts its telemetry metadata. The sink switch extracts the telemetry instruction header before progressing the original packet. Depending on the result of event detection, the sink switch may generate a telemetry report containing all of the telemetry metadata from all hops across the network.

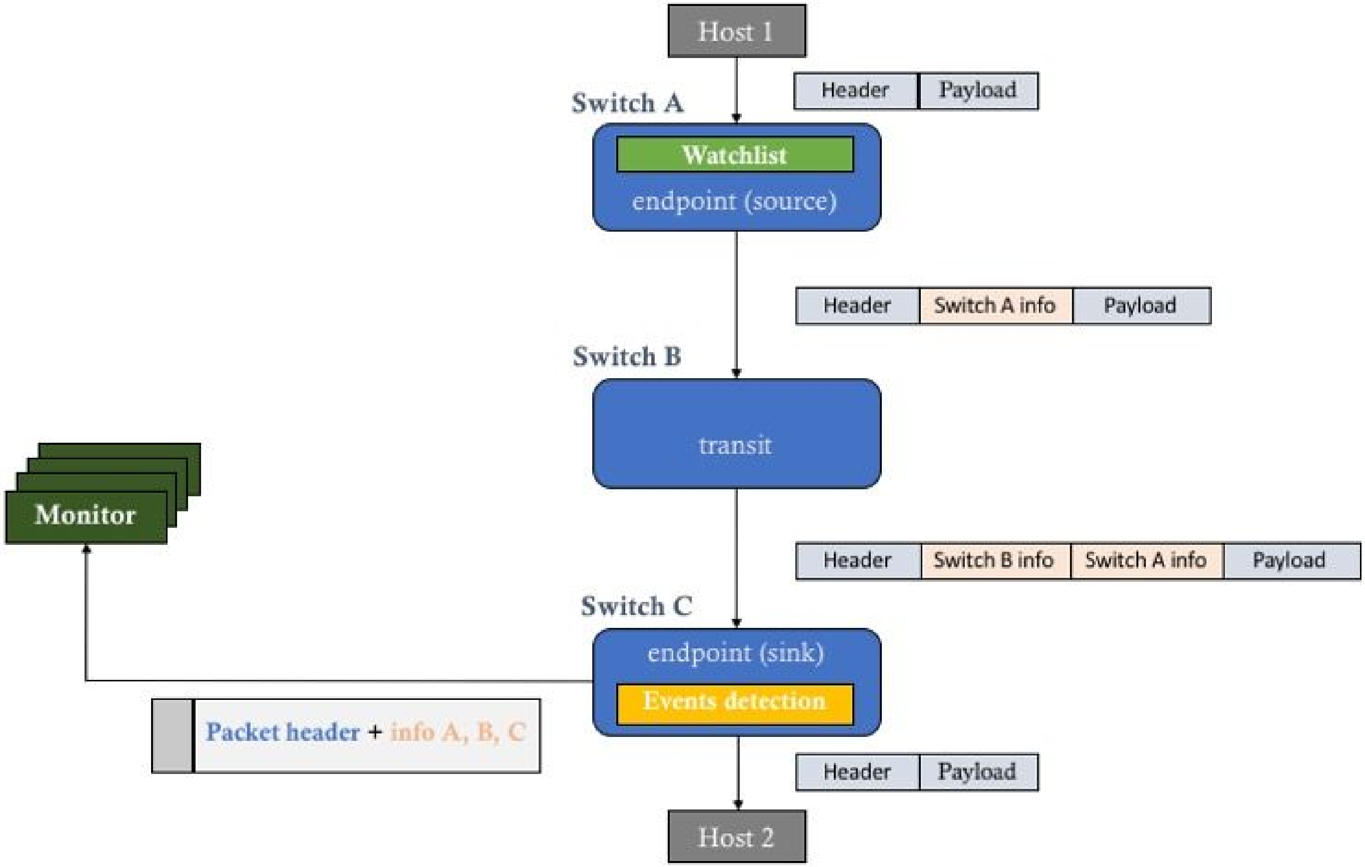

Figure 2: Telemetry Architecture with reports generated by sink switches

In order to reduce complexity at the sink switch, some telemetry reports may include embedded telemetry metadata intermingled with the original packet headers. This simplifies generation of telemetry reports due to receipt of data packets with embedded telemetry metadata. The telemetry data plane specification such as [INT] or [iOAM] specifies the format for this portion of the telemetry metadata. This approach reduces data plane complexity, allowing for all telemetry report processing and generation to be done in the data plane itself without any need to punt to the control plane for further processing.

The sink switch has the option to add its local telemetry metadata either in the telemetry report headers defined in this specification, or in the embedded telemetry metadata intermingled with the original packet headers.

### Using Different Telemetry Modes for Different Telemetry Categories

Even when in-band (in-situ) telemetry mode is used for the category of tracked flows, it is possible to use the postcard telemetry mode for other categories such as dropped packets and congested queues. The latter categories are often monitored as per switch, per port, or per queue local events, suggesting that telemetry reports should be generated directly from the affected switch(es).

## Correlation of Telemetry Reports

Telemetry reports for a specific application flow matching a flow watchlist may be received from multiple network devices. In case of postcard mode, each hop will generate a separate telemetry report. Even when telemetry metadata is embedded in the data plane according to a specification such as [INT] or [iOAM], telemetry reports for one flow may still be generated by multiple network devices in case of path change or in case of dropped packets.

The distributed telemetry monitoring system may want to correlate these telemetry reports, based on the original packet header fields included in each telemetry report. The telemetry reports include one association bit for each telemetry report category, providing hints to the distributed telemetry monitoring system that it can use to assist with telemetry report correlation. In particular, the distributed telemetry monitoring system may want to apply certain types of telemetry report correlation only when the corresponding bits are set.

The mechanisms for correlation are left to each implementation, and are considered to be beyond the scope of this specification.

# Telemetry Report Formats

This section specifies the packet formats for telemetry reports.

## Outer Encapsulation

Telemetry reports are defined using a UDP-based encapsulation. Various outer encapsulations may be used to transport the UDP packets. Typically this would simply be an Ethernet header, followed by an IPv4 or IPv6 header, followed by the UDP header. This specification does not preclude the use of different transport encapsulations.

The source IP address identifies the network device that generates the telemetry report.
The Destination IP address identifies a location in the distributed telemetry monitoring system that will receive the telemetry report.

In case of IPv4, as is the case for any other IP packet, either the Don’t Fragment (DF) bit must be set, or the IPv4 ID field must be set so that the value does not repeat within the maximum datagram lifetime for a given source address/destination address/protocol tuple.

### UDP header (8 octets)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Source Port may optionally be used to carry flow entropy, for example based on a hash of the inner 5-tuple. Otherwise, it should be set to 0.

The Destination Port is user configurable. The expectation is that the same Destination Port value will be used for all telemetry reports in a particular deployment.

## Telemetry Report Headers

### Telemetry Report Fixed Header (12 octets)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ver  |NProto |D|Q|F|         Reserved           |   hw_id    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Ingress Timestamp                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NProt: Next Protocol {

0 Ethernet

1 Telemetry Drop header, followed by Ethernet

2 Telemetry Switch Local header, followed by Ethernet

}

D: Dropped - Indicates that at least one packet matching a drop watchlist was dropped.

Q: Congested Queue Association - Indicates the presence of congestion on a monitored queue.

F: Tracked Flow Association - Indicates that this telemetry report is for a tracked flow, i.e. the packet matched a flow watchlist somewhere (in case of INT or iOAM) or locally (in case of postcard). The report might include INT or iOAM metadata beyond the inner ethernet header. Other telemetry reports are likely to be received for the same tracked flow, from the same network device and (in case of drop reports, postcard or path changes) from other network devices.

hw_id: Identifies the hardware subsystem within the source that generated this report. For example, in a chassis with multiple linecards this could identify a specific linecard, or a subsystem within a linecard.

Sequence Number: Reflects the sequence of reports from a specific hw_id to a particular telemetry report destination. This can be used to detect loss of telemetry reports before they reach their intended destination.

Ingress Timestamp: The device local time when the packet was first received on the ingress physical or logical port, in nanoseconds.

## Telemetry Drop Report Header (12 octets)

This header is used to report telemetry metadata from a network device that dropped a packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Switch id                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Ingress port id        |         Egress port id        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Queue id    |  Drop reason  |              Pad              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following metadata are defined in [INT]:

- Switch id
- Ingress port id
- Egress port id

In addition, the Telemetry Report Header defines the following metadata:

- Queue id
  - The ID of the queue via which a packet was sent out, or where a packet was dropped. This ID is unique within the scope of the egress port.
- Drop reason

    - An enumeration that indicates the reason why a packet was dropped, for example as defined in github.com/p4lang/switch.

### Telemetry Switch Local Report Header (16 octets)

This header may be used to include telemetry metadata from the network device generating the report.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Switch id                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Ingress port id        |        Egress port id         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Queue id    |                Queue occupancy                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Egress timestamp                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following metadata are defined in [INT]:

- Switch id
- Ingress port id
- Egress port id
- Queue id
- Queue occupancy

In addition, the Telemetry Switch Local Report Header defines the following metadata:

- Egress timestamp
    - The device local time when the packet was sent out on the ***egress*** physical or logical port, in nanoseconds.

Note that the ingress timestamp is already included in the Telemetry Report Fixed Header, so there is no need to include it in the Telemetry Switch Local Report Header.

## Flow Identification

There is no explicit metadata defined for flow identification. The expectation is that packet headers will be included in the telemetry report, allowing the distributed telemetry monitoring system to categorize and identify flows in any manner that it desires.

# Embedded Telemetry Metadata

There may still be further telemetry metadata embedded within the payload after the Telemetry Report headers. For example, this is typically the case when there is telemetry metadata from hops prior to the network device generating the report. The telemetry metadata will typically be encoded using a defined data plane format such as [INT] or [iOAM].

A network device generating a telemetry report may include its local telemetry metadata in any of the following:

- the embedded telemetry metadata,
- the Telemetry Switch Local Report header or Telemetry Drop Report header in the same telemetry report as the embedded telemetry metadata from previous hops, or
- the Telemetry Switch Local Report header or Telemetry Drop Report header in a separate telemetry report from the embedded telemetry metadata from previous hops. Note that in this case the ingress timestamp will be the same in the Telemetry Report Fixed Header in both telemetry reports.

If the Tracked Flow Association bit is set to 0, then there will not be any embedded telemetry metadata in the report.

If the Tracked Flow Association bit is set to 1, there may or may not be any embedded telemetry metadata in the report. See the next section for parsing considerations.

## Parsing Considerations

When a telemetry report is received by the distributed telemetry monitoring system, it must parse the packet to retrieve the telemetry metadata and to identify the flow. Figure 3 shows which headers will be present at the beginning of the packet, assuming a simple Ethernet/IP transport of the telemetry report packet.

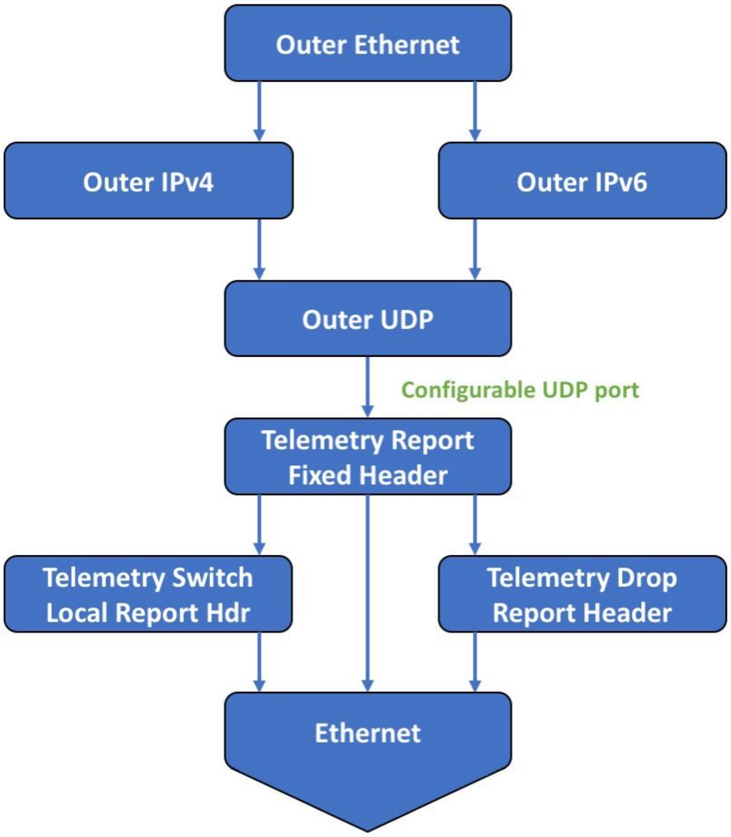
Outer Ethernet
Outer IPv4
Outer IPv6
Outer UDP
Configurable UDP port
Telemetry Report Fixed Header
Telemetry Switch Local Report Hdr
Telemetry Drop Report Header
Ethernet

Figure 3: Telemetry Report Outer Encapsulation Format

The packet format after this point can vary depending on the format of the original packet, and whether embedded telemetry metadata is present. The following figures show a few examples of the remaining packet format. These examples are not intended to be complete or exclusive.

Figure 4 shows the remaining packet format when the original packet is a simple flat packet and there is no embedded telemetry metadata.

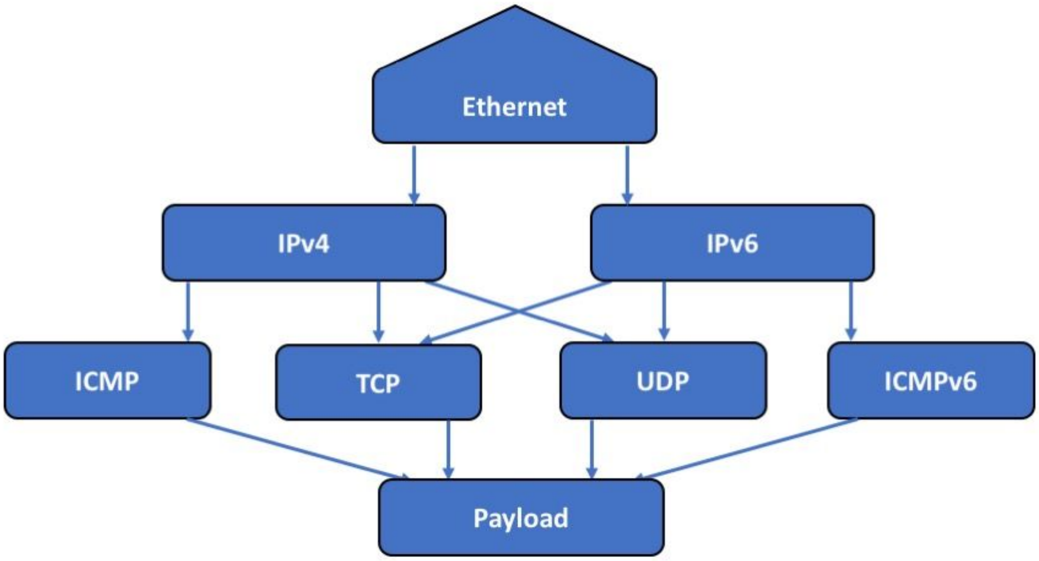

Figure 4: Remaining Packet Format - Flat Packet

Figure 5 shows the remaining packet format when the original packet is a simple flat packet and there is embedded INT over TCP/UDP/ICMP telemetry metadata.

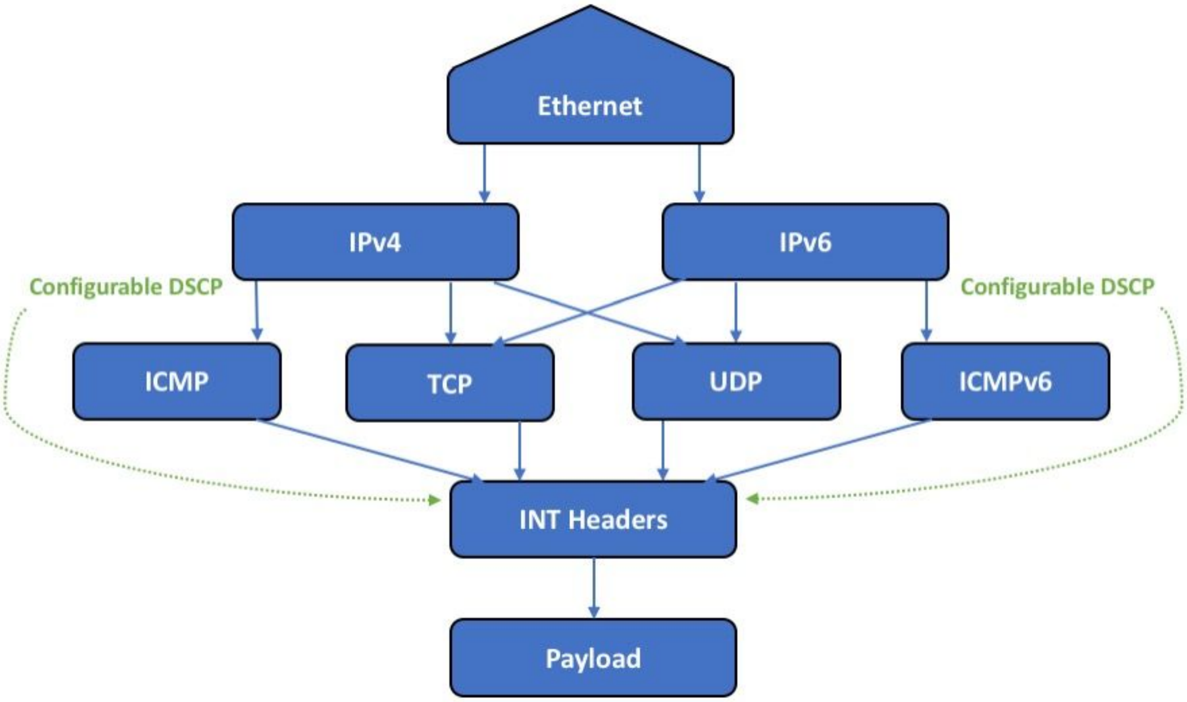

Figure 5: Remaining Packet Format - Flat Packet with INT over TCP/UDP/ICMP

Even when using INT over TCP/UDP/ICMP, the original packet may be an encapsulated packet such as a VXLAN packet. When processing a telemetry report for an encapsulated packet, the distributed telemetry monitoring system may desire to categorize flows based on inner headers. In this case, it should parse the telemetry report all the way down past any embedded telemetry metadata (if present), even when a Telemetry Report Drop Header or Telemetry Report Switch Local Header is present. It may also want to process the embedded telemetry metadata, for example to recognize the case where a path change directs traffic to a congested switch where packets are being dropped.

Figure 6 shows the remaining packet format when the original packet is a VXLAN packet and there is embedded INT over TCP/UDP/ICMP telemetry metadata.

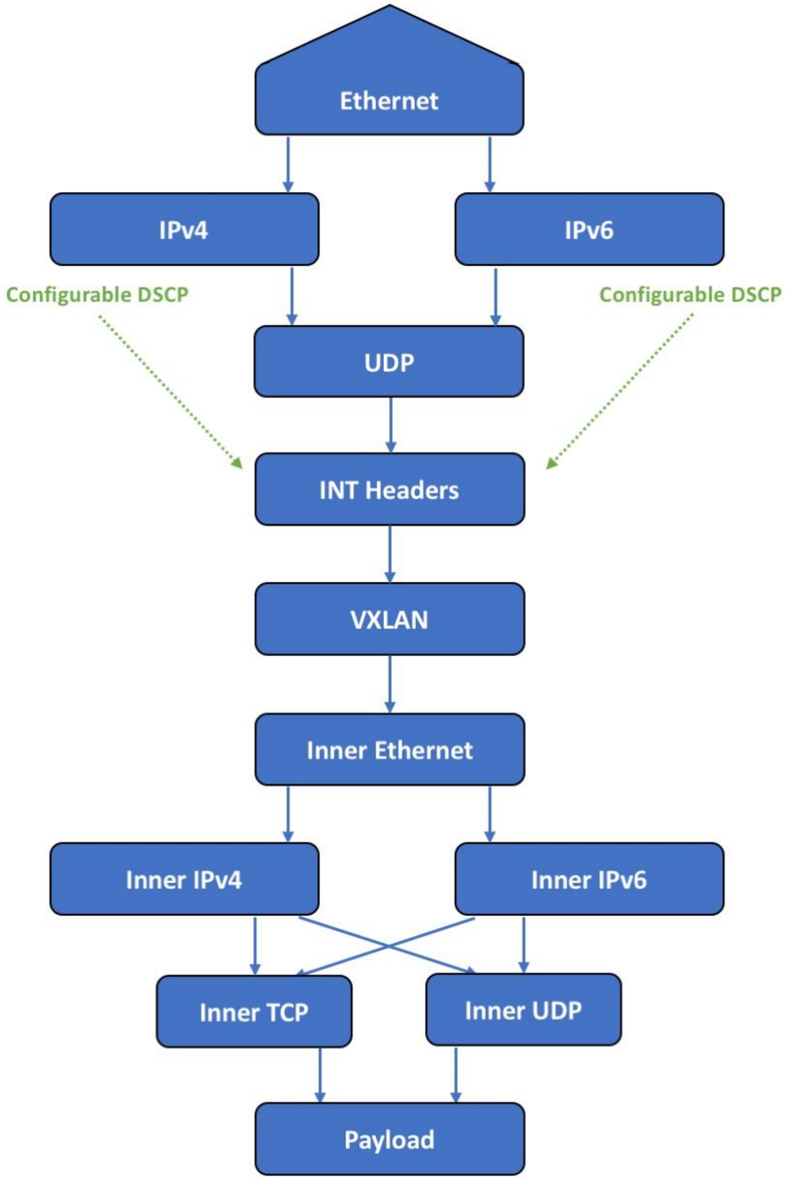

Figure 6: Remaining Packet Format - VXLAN Packet with INT over TCP/UDP

Figure 7 shows the remaining packet format when the original packet is a VXLAN packet and there is embedded iOAM Trace telemetry metadata. See [iOAM] and [iOAM_VXLAN_GPE] for further details.

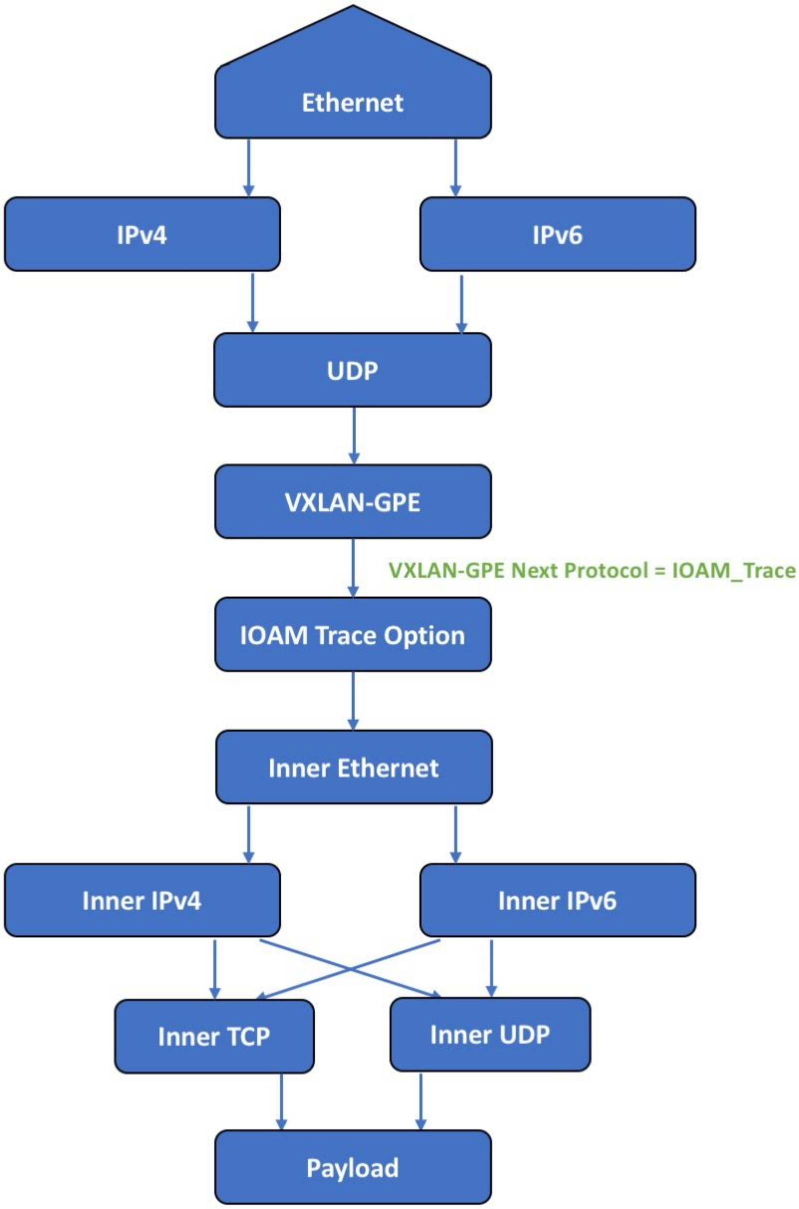

Figure 7: Remaining Packet Format - VXLAN Packet with iOAM Trace

# References

[INT] In-band Network Telemetry (INT), October 2017.

[iOAM] Data Fields for In-situ OAM, draft-ietf-ippm-ioam-data-00, September 2017.

[iOAM_reqs] Requirements for In-situ OAM, draft-brockners-inband-oam-requirements-03, March 2017.

[iOAM_VXLAN_GPE] VXLAN-GPE Encapsulation for In-situ OAM Data, draft-brockners-ioam-vxlan-gpe-00, October 2017.

[IPFIX] Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC 7011, September 2013.

[SAI_DTel] SAI Data Plane Telemetry Proposal, October 2017.