Telemetry Report Format Specification

Version 2.0

The P4.org Applications Working Group
Contributions from *CableLabs, Cisco Systems, Intel, VMware, Xilinx*

2018-06-11

Contents

1. 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. Further information regarding the motivation and usage of detailed telemetry information can be found in the IETF draft for In-situ OAM [1].

Specifications are being defined for embedding telemetry metadata within data packets, such as INT [2] and IOAM [3]. This allows for telemetry metadata to be collected as packets traverse a

[1] Requirements for In-situ OAM, draft-brockners-inband-oam-requirements-03, March 2017.
[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.
[3] Data Fields for In-situ OAM, draft-ietf-ippm-ioam-data-05, March 2020.

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, routers, NICs) 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.

1.1. 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 IOAM [3] and INT [2], when using INT-MD mode. This specification is also applicable when each network device directly generates telemetry reports, including transit network devices in the middle of the network, such as in INT-XD (where data packet formats between successive network devices are not affected) and in INT-MX (where only INT instructions are embedded in data packets).

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-MD and IOAM. The embedded telemetry metadata is included as is in telemetry reports, so the packet formats defined in INT-MD and IOAM also define some aspects of the telemetry report format. See Section 3.4 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 [4] and SAI TAM 2.0 [5].
- 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 [6].

2. Key Concepts

2.1. Telemetry Report Definition

We define a *telemetry report* as a message that a network device sends to the monitoring system. A *telemetry report* typically 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 an in-band

[3] Data Fields for In-situ OAM, draft-ietf-ippm-ioam-data-05, March 2020.
[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.
[4] SAI Data Plane Telemetry API Proposal, December 2017.
[5] SAI Telemetry and Monitoring (TAM) 2.0, July 2019.
[6] Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC 7011, September 2013.

mechanism like INT-MD 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 network devices that generate telemetry reports are referred to as *nodes* in the rest of this specification. Depending on deployment scenarios, examples of such *nodes* may include network devices such as switches, routers, and NICs.

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

2.2. 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 *watchlist* in this specification) determines which data packets to monitor by matching packet header fields and optionally identification of the ingress interface. The action in the matched entry in the *watchlist* may specify monitoring of this flow, triggering generation of telemetry reports based on these packets. (Note that the telemetry specific watchlist is not performing any access control. It only makes decisions related to monitoring actions.) 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 dropped packets matching a telemetry specific access control list (called a *watchlist* in this specification), when the action in the matched entry specifies monitoring of dropped packets. 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 Section 3.3. New categories (and corresponding association bits) may be added to future versions of this specification.

Nodes 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 [4] for one API proposal to enable data plane telemetry capabilities in nodes across all three categories.

[4] SAI Data Plane Telemetry API Proposal, December 2017.

2.3. Telemetry Report Events

Telemetry reports are typically triggered by packet processing at a node. 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. Nodes may apply filters to determine when significant events occur that should be reported. This is called event detection in this specification. For example, a node 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.

2.4. Telemetry Reporting Modes

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

2.4.1. Per Hop Reports in INT-XD/MX modes

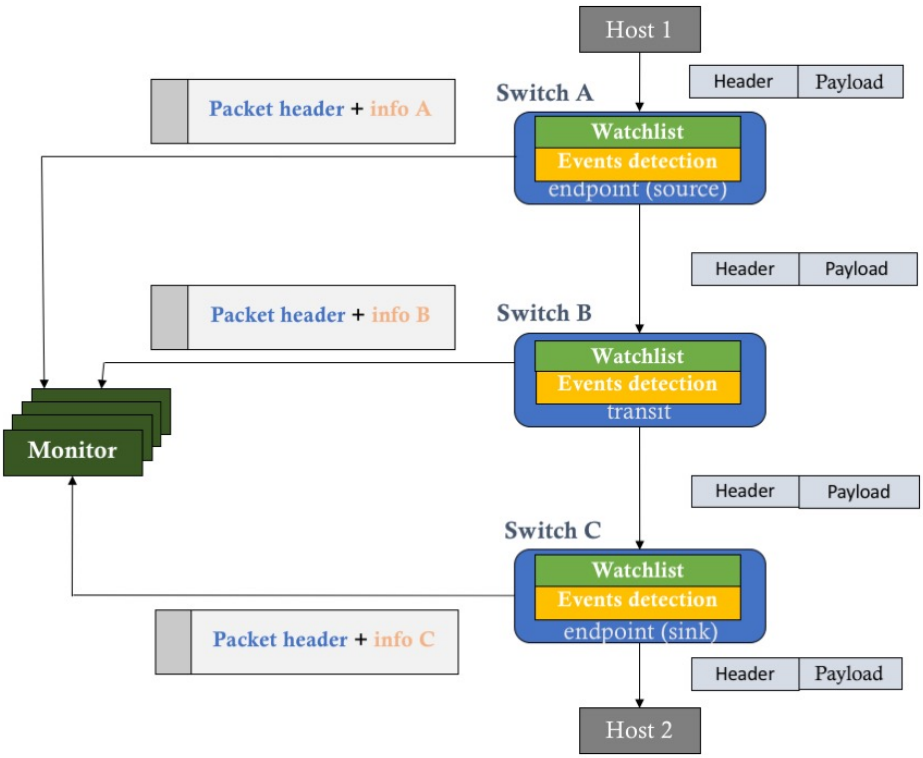


Figure 1. INT-XD mode - Telemetry Architecture with per hop reports generated by each node

In the *INT-XD (eXport Data)* mode, as defined in the INT specification [2], each node generates

[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.

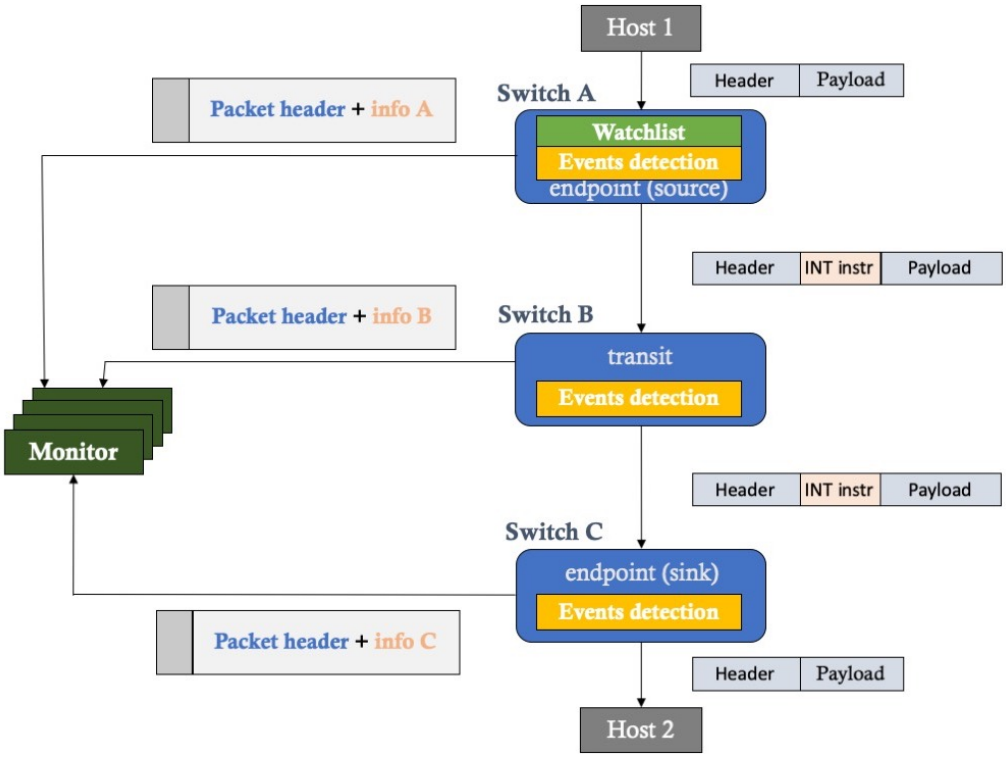


Figure 2. INT-MX mode - Telemetry Architecture with per hop reports generated by each node

its own telemetry reports (Figure 1). The distributed telemetry monitoring system will receive reports from different nodes, each describing the telemetry metadata (such as node IDs, interface IDs, latency) for one hop. Within the per hop telemetry reports, the telemetry metadata precedes the details of the original packet header. There is no change to data packets traversing the network. This mode was known as "Postcard" mode in previous versions of this Telemetry Report specification.

In the *INT-MX (eMbed instruct(X)ions)* mode, as defined in the INT specification [2], the source node embeds instructions in the INT-MX header. Upon receipt of a packet with this INT type, each node in the path generates its own telemetry reports, as shown in Figure 2. The distributed telemetry monitoring system will receive reports from different nodes, each describing the telemetry metadata (such as node IDs, interface IDs, latency) for one hop. The only change to data packets is the source node embedding the INT-MX Header with instructions. The sink node removes the INT-MX Header from such packets. When using INT-MX mode, the telemetry metadata precedes the details of the original packet headers within the telemetry report.

2.4.2. Stacked Reports in INT-MD mode

In the *INT-MD (eMbed Data)* mode, telemetry metadata is embedded in between the original headers of data packets as they traverse the network, as shown in Figure 3. 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 node may insert a telemetry instruction header, thereby instructing downstream

[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.

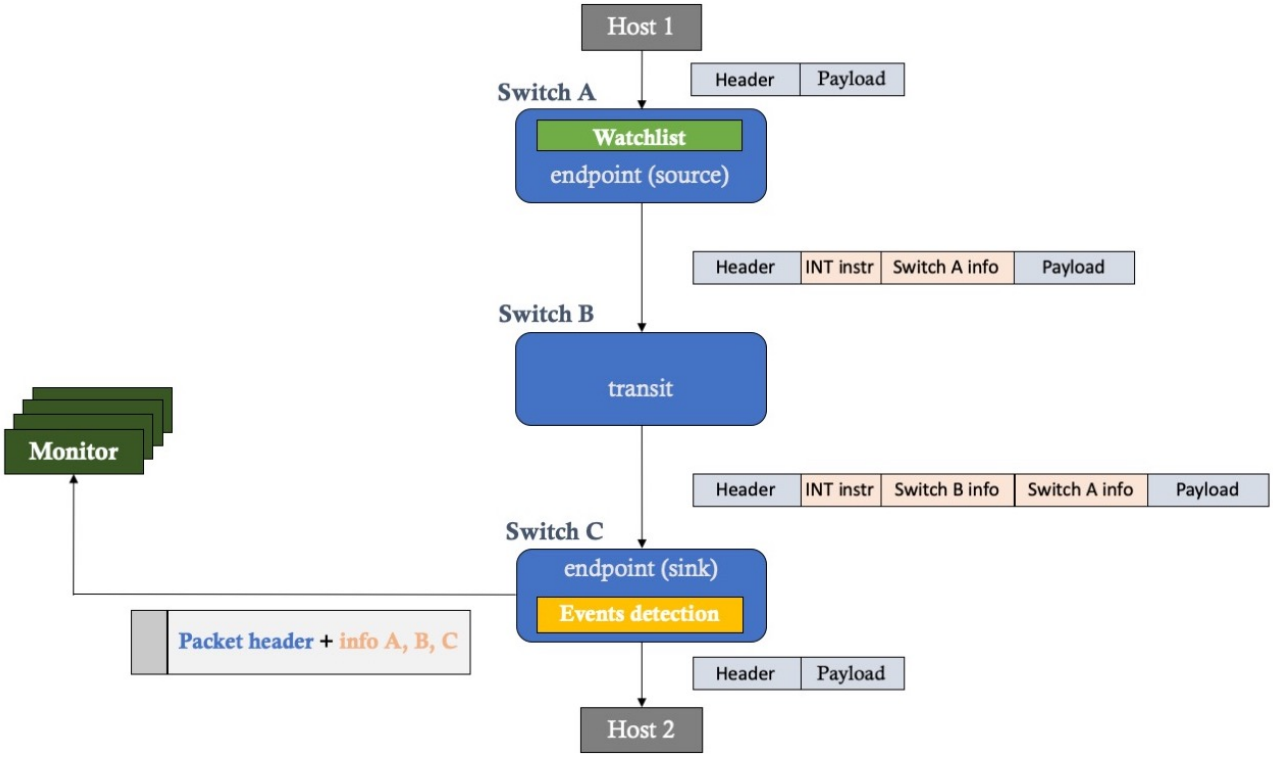


Figure 3. INT-MD mode - Telemetry Architecture with stacked reports generated by sink nodes

nodes to add the desired telemetry metadata. At each hop, the transit node inserts its telemetry metadata at the top of the stack. The sink node extracts the telemetry instruction header before progressing the original packet. Depending on the result of event detection, the sink node may generate a telemetry report containing the stacked telemetry metadata from all hops across the network.
In order to reduce complexity at the sink node, some telemetry reports may include embedded telemetry metadata intermingled with the details of 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 node 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.

2.4.3. Using Different Telemetry Modes for Different Telemetry Categories

Even when stacked reports are generated for the category of tracked flows using INT-MD mode, it is possible to generate per hop reports for other categories such as dropped packets and congested queues. The latter categories are often monitored as per node, per port, or per queue local events, suggesting that telemetry reports should be generated directly from the affected node(s).

2.5. Correlation of Telemetry Reports

Telemetry reports for a specific application flow matching a watchlist may be received from multiple nodes. In case of INT-XD and INT-MX modes, each hop will generate a separate report. Even when stacked 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 nodes in case of path change or in case of dropped packets.

The distributed telemetry monitoring system may want to correlate these telemetry reports on a per flow basis (see Section 2.6 for details on flow identification). 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.

2.6. Flow Identification

There is no explicit metadata defined for flow identification. The expectation is that either:

- a truncated packet fragment including the original 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, or
- domain specific flow identification metadata will be included.

Tunneled packets such as VXLAN packets raise the question whether flow identification should be based on outer or inner headers. The answer may vary depending on the goals of tracked flow monitoring, deployment aspects, operational issues, and the capabilities of the distributed telemetry monitoring system. Note that it is possible to identify flows based on inner packet headers even when using an INT encapsulation based on outer headers such as INT over TCP/UDP.

When using INT-MD and flow identification based on inner headers is desired, the distributed telemetry monitoring system should parse the truncated packet fragment all the way down past any embedded telemetry metadata (if present), even when the Individual Report includes optional metadata such as drop reason. 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 node where packets are being dropped.

2.7. Coalescing A Group of Telemetry Reports In A Single Packet

Starting with Version 2.0, the telemetry report format allows for a group of individual reports, each corresponding to one data plane packet or one flow, to be coalesced into the same telemetry report packet. This can help reduce the packet processing overhead associated with telemetry reports. The only restrictions are that all of the individual reports in the group must be generated by the same node (or hardware subsystem within the node), and they are all addressed to the same destination within the distributed telemetry monitoring system. Beyond those restrictions, implementations are free to come up with their own methods for deciding which individual reports to group together.

Support for coalescing a group of telemetry reports in a single packet is optional.

3. Telemetry Report Format

This section specifies the packet format for telemetry reports.

3.1. Outer Encapsulation

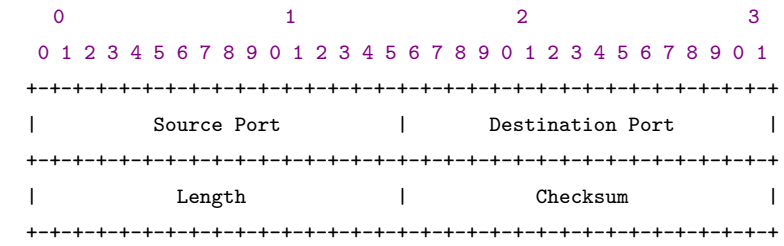
Telemetry reports are defined using a UDP-based encapsulation. Various outer encapsulations may be used to transport the UDP packets. Typically this would simply be an Ethernet header, followed by an IPv4 or IPv6 header, followed by the UDP header. This specification does not preclude the use of different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

The source IP address identifies the node 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.

3.1.1. UDP header (8 octets)

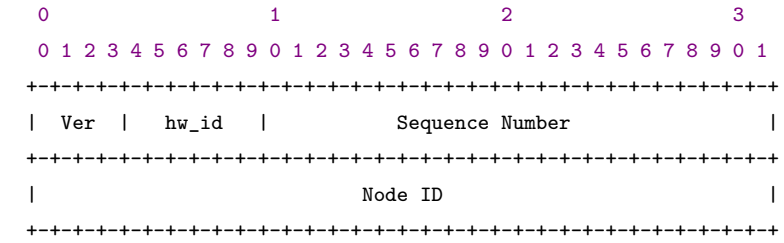


The Source Port may optionally be used to carry flow entropy, for example based on a hash of the inner 5-tuple. Otherwise, it should be user configurable.

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.

3.2. Telemetry Report Group Header (Ver 2.0) (8 octets)

The Telemetry Report Group Header immediately follows the UDP header whose destination port identifies the contents as a telemetry report. This header contains the common fields in a telemetry report that optionally contains multiple coalesced individual reports, each corresponding to one data plane packet. There is at most one instance of the Telemetry Report Group Header in a packet.



Ver (4b): Version
This specification defines **version 2**.

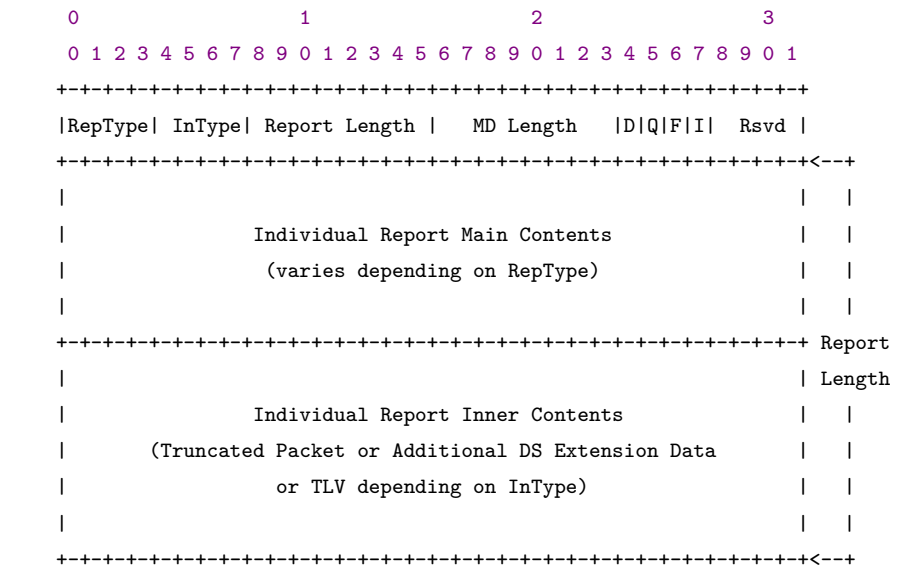
hw_id (6b): Hardware ID
Identifies the hardware subsystem within the node that generated this report. For example, in a chassis with multiple linecards this could identify a specific linecard, or a subsystem within a linecard. The hw_id is unique within the scope of a Node ID.

Sequence Number (22b): Sequence Number
Reflects the sequence of reports from a specific combination of (Node ID, hw_id) to a particular telemetry report destination. This can be used to detect loss of telemetry reports before they reach their intended destination.

Node ID (32b): Node ID
The unique ID of a node. This is generally administratively assigned. Node IDs must be unique within a management domain.

3.3. Individual Report Header (Ver 2.0) (4+ octets)

Each telemetry report packet contains one or more individual reports immediately following the Telemetry Report Group Header. Each report within the packet starts with the Individual Report Header. The presence of multiple reports corresponding to multiple data plane packets, possibly from multiple flows, can be determined by comparing the *Report Length* in the Individual Report Header with the length in the UDP header.



RepType (4b): Report Type
Type of the individual report:

- **0**: Inner Only
- **1**: INT
- **2**: IOAM
- **3 – 15**: Reserved

InType (4b): Inner Type

Type of data embedded after the *Individual Report Main Contents*:

- 0: None
- 1: TLV
- 2: Domain Specific Extension Data
- 3: Ethernet
- 4: IPv4
- 5: IPv6
- 6-15: Reserved.

Report Length (8b): Report Length

Indicates the length of the Individual Report Header in a multiple of 4-byte words, including the *Individual Report Main Contents* and *Individual Report Inner Contents*, but excluding the length of the first 4-byte word (*RepType, InType, Report Length, MD Length, D, Q, F, I, Rsvd*).

For *RepType* codepoint 1 *INT*, this includes the length of *RepMdBits, Domain Specific ID, DSMdBits, DSMdstatus, Variable Optional Baseline Metadata,* and *Variable Optional Domain Specific Metadata* (see Section 3.3.1).

MD Length (8b): Metadata Length

Indicates the length of metadata included in this report in a multiple of 4-byte words. This may help the telemetry monitoring system determine where the *Individual Report Inner Contents* begins. Note that this does not include the length of the fixed portion of the *Individual Report Main Contents*.

For *RepType* codepoint 1 *INT*, this includes the length of the *Variable Optional Baseline Metadata* and *Variable Optional Domain Specific Metadata* in 4-byte words (see Section 3.3.1).

D (1b): Dropped

Indicates that at least one packet matching a watchlist was dropped.

Q (1b): Congested Queue Association

Indicates the presence of congestion on a monitored queue.

F (1b): Tracked Flow Association

Indicates that this telemetry report is for a tracked flow, i.e. the packet matched a watchlist somewhere (in case of INT-MD, INT-MX or IOAM) or locally (in case of INT-XD). The report might include INT-MD or IOAM metadata in the truncated packet. Other telemetry reports are likely to be received for the same tracked flow, from the same node and (in case of drop reports, INT-MX, INT-XD or path changes) from other nodes.

I (1b): Intermediate Report

Indicates that a transit node sent this intermediate report for INT-MD.

Rsvd (4b): Reserved

Should be set to zero upon transmission, and ignored upon reception

Individual Report Main Contents

The metadata that comprises this report, along with associated fields that assist in processing the metadata. The format varies depending on *RepType*.

When the *RepType* value is *Inner Only*, then the Individual Report Main Contents is empty. *MD Length* should be set to zero upon transmission, and ignored upon reception.

The *INT* Individual Report Main Contents format (see Section 3.3.1) was derived with INT 2.0/2.1 in mind, but it may be used with other INT versions as well. It is possible that other *RepType* codepoints and corresponding Individual Report Main Contents formats may be

defined for future versions of INT.

The *IOAM* Individual Report Main Contents format will be defined in a future version of this specification.

Truncated Packet

L2/L3/ESP/L4 of the packet for flow details. Presence of this field is indicated by *InType* codepoint 3, 4, or 5, which identifies the type of header at the beginning of the truncated packet. The length of the truncated packet can be determined as *Report Length* - ((fixed length of *Individual Report Main Contents*) + *MD Length*).

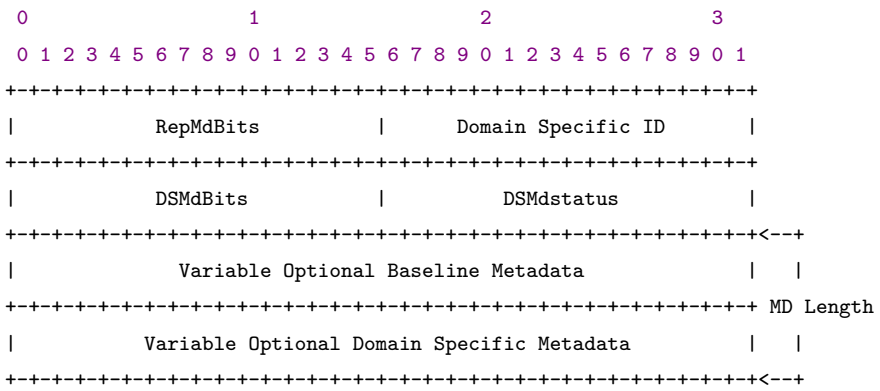
Additional DS Extension Data

Additional Domain Specific Extension Data, whose format can be determined from the *Domain Specific ID* specified in the *Individual Report Main Contents*. For *RepType* codepoint 1 *INT*, this is additional domain specific data that is not associated with *DSMdBits*. Presence of this field is indicated by *InType* codepoint 2.

TLV

Type Length Value format. Multiple TLV formatted data (see Section 3.3.2). Presence of this field is indicated by *InType* codepoint 1.

3.3.1. Individual Report Main Contents for *RepType* 1 (*INT*) (8+ octets)



RepMdBits (16b): Report Metadata Bits

Bitmap that indicates which optional baseline metadata is present in the telemetry report header. Each bit represents 4 octets of optional metadata, except for bits 4, 5 & 6 which represents 8 octets of optional metadata.

- bit 0 (MSB): Reserved
- bit 1: Level 1 Ingress Interface ID (16 bits) & Egress Interface ID (16 bits)
- bit 2: Hop Latency
- bit 3: Queue ID (8 bits) + Queue Occupancy (24 bits)
- bit 4: Ingress Timestamp (64 bits)
- bit 5: Egress Timestamp (64 bits)
- bit 6: Level 2 Ingress Interface ID (32 bits) + Egress Interface ID (32 bits)
- bit 7: Egress Port TX Utilization
- bit 8: Buffer ID (8 bits) + Buffer Occupancy (24 bits)
- bit 9-14: Reserved.
- bit 15: Queue ID (8 bits) + Drop Reason (8 bits) + Padding (16 bits)

This specification defines the following metadata:

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

See the INT specification [2] for definitions of the remaining metadata.

Domain Specific ID (16b)
The unique ID of the INT Domain.

The *Domain Specific ID* value 0x0000 is the default, known to all nodes. For this value, all *DSMdBits* are treated as reserved. Operators can assign values in the range 0x0001 to 0xFFFF.

DSMdBits (16b): Domain Specific Md Bits
Bitmap that indicates which optional domain specific metadata is present in the the telemetry report header. Each bit represents 4 octets or a multiple of 4 octets of domain specific optional metadata.

When using INT-MD or INT-MX, if the *Domain Specific ID* does not match any Domain ID known to this node, then the node may either:

- Set the Telemetry Report *DSMdBits* field to zero and rederive the Telemetry Report *MD Length* from *RepMdBits*, or
- Not send any of its own metadata to the monitoring systems, doing any of the following:
  - Not generate any Telemetry Report, or
  - Clear *RepMdBits* and *MD Length* as well as *DSMdBits* (this only makes sense for INT-MD), or
  - Use *RepType* value *Inner Only* (this only makes sense for INT-MD).

DSMdstatus (16b): Domain Specific Md Status
Indicates the domain specific metadata status.

Variable Optional Baseline Metadata
The metadata corresponding to *RepMdBits*, 4 octets for each bit, except 8 octets for bits 4, 5 & 6.

If a node receives an INT-MX or INT-MD packet with an *Instruction Bitmap* that requests one or more metadata values that are not available or reserved, then the node must ensure that the corresponding bit(s) in the Telemetry Report *RepMdBits* that specify the unavailable metadata are not set. The Telemetry Report *MD Length* must be derived based on the adjusted *RepMdBits* (and *DSMdBits*) values.

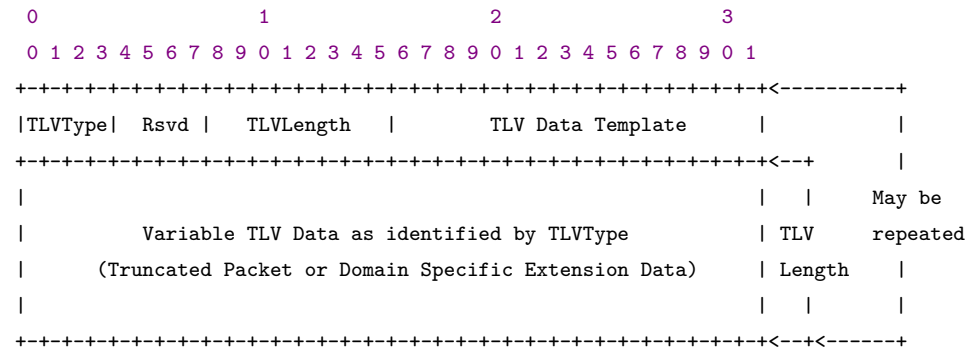
Variable Optional Domain Specific Metadata
The metadata corresponding to *DSMdBits*, 4 octets or a multiple of 4 octets for each bit.

If a node receives an INT-MX or INT-MD packet with a *DS Instruction* that requests one or more metadata values that are not available or reserved, then the node must ensure that the corresponding bit(s) in the Telemetry Report *DSMdBits* that specify the unavailable metadata are not set. The Telemetry Report *MD Length* must be derived based on the adjusted *DSMdBits* (and *RepMdBits*) values.

[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.

3.3.2. Individual Report Inner Contents for *InType* 1 (*TLV*) (4+ octets)

One or more TLVs, each following the format defined in this section. The presence of multiple TLVs can be determined by comparing the *TLVLength* in the first TLV with the *Report Length* in the Individual Report Header.



TLVType (4b): TLV data type

- **0**: Domain Specific Extension Data
- **1**: Ethernet
- **2**: IPv4
- **3**: IPv6
- **4-15**: Reserved.

Rsvd (4b) – Reserved

Should be set to zero upon transmission and ignored upon reception.

TLVLength (8b)

Indicates the length, in 4-byte words, of *Variable TLV Data* as identified by *TLVType*. Note that this does not include the length of the first 4-byte word (*TLVType, Rsvd, TLVLength, TLV Data Template*).

TLV Data Template (16b)

Specifies the format of the *Variable TLV Data*. A non-zero *TLV Data Template* value specifies the template for *TLVType* codepoint of *Domain Specific Extension Data*. For *TLVType* codepoints *Ethernet*, *IPv4*, and *IPv6*, the *TLV Data Template* value should be zero upon transmission and ignored upon reception.

Variable TLV Data

Variable length data based upon *TLVType*. The following two fields are defined in this version.

Truncated Packet

L2/L3/ESP/L4 of the packet for flow details. Presence of this field is indicated by *TLVType* codepoint 1, 2, or 3, which identifies the type of header at the beginning of the truncated packet.

Domain Specific Extension Data

Domain Specific Extension Data, whose format can be determined from the *Domain Specific ID* specified in the *Individual Report Main Contents* and the *TLV Data Template*. Presence of this field is indicated by *TLVType* codepoint 0.

For *RepType* codepoint 1 *INT*, this is additional domain specific data that is not associated with *DSMdBits*.

3.4. Embedded Telemetry Metadata In Stacked Reports

There may still be further telemetry metadata embedded within a truncated packet fragment. For example, this is typically the case when there is stacked telemetry metadata from hops prior to the node generating the report. The telemetry metadata will typically be encoded using a defined data plane format such as INT-MD or IOAM.

A node generating a telemetry report with stacked telemetry metadata may include its local telemetry metadata in any of the following:

- the embedded telemetry metadata in a truncated packet fragment,
- the stacked telemetry metadata in domain specific extension data,
- the *Individual Report Main Contents* in the same Individual Report Header that contains the stacked telemetry metadata from previous hops, in either a truncated packet fragment or in domain specific extension data, or
- the *Individual Report Main Contents* in a separate report from the stacked telemetry metadata from previous hops. Note that in this case the ingress timestamp (if present) will be the same in both reports.

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

If the *Tracked Flow Association (F)* bit is set to 1, there may or may not be any embedded telemetry metadata in the report.

4. Examples of Telemetry Reports

This section shows examples of Telemetry Reports. These examples are not intended to be complete or exclusive.

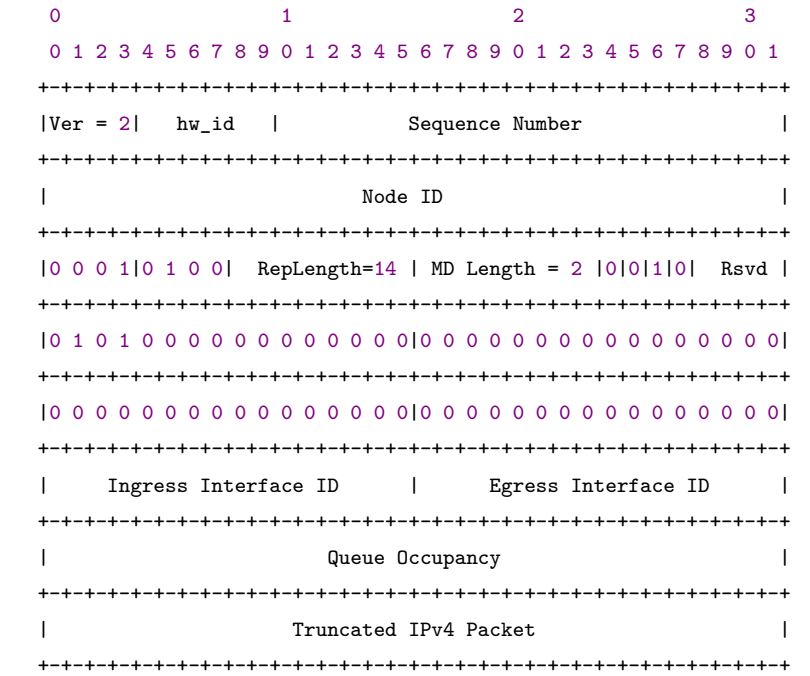
4.1. Example with Baseline Metadata and Truncated IPv4

This example shows a telemetry report with baseline metadata consisting of level 1 interface IDs and queue occupancy, and a truncated IPv4 packet.

The values of Telemetry Report header fields in this example are as follows:

- *Ver* = 2
- *RepType* = 1 (*INT*)
- *InType* = 4 (*IPv4*)
- *MD Length* = 2 for level 1 interface IDs and queue occupancy
- *Report Length* = 2 (individual report fixed size) + 2 (*MD Length*) + 5 (original IP header) + 5 (original TCP)
  = 14 (assuming truncated packet fragment ends after the original TCP header)
- *D* = 0 since the packet is not dropped
- *Q* = 0 since the packet does not experience congestion at Switch3
- *F* = 1
- *I* = 0
- *Domain Specific ID*, *DsMdBits* and *DSMdstatus* are all 0

Below is the telemetry report packet starting from the Telemetry Report Group Header.

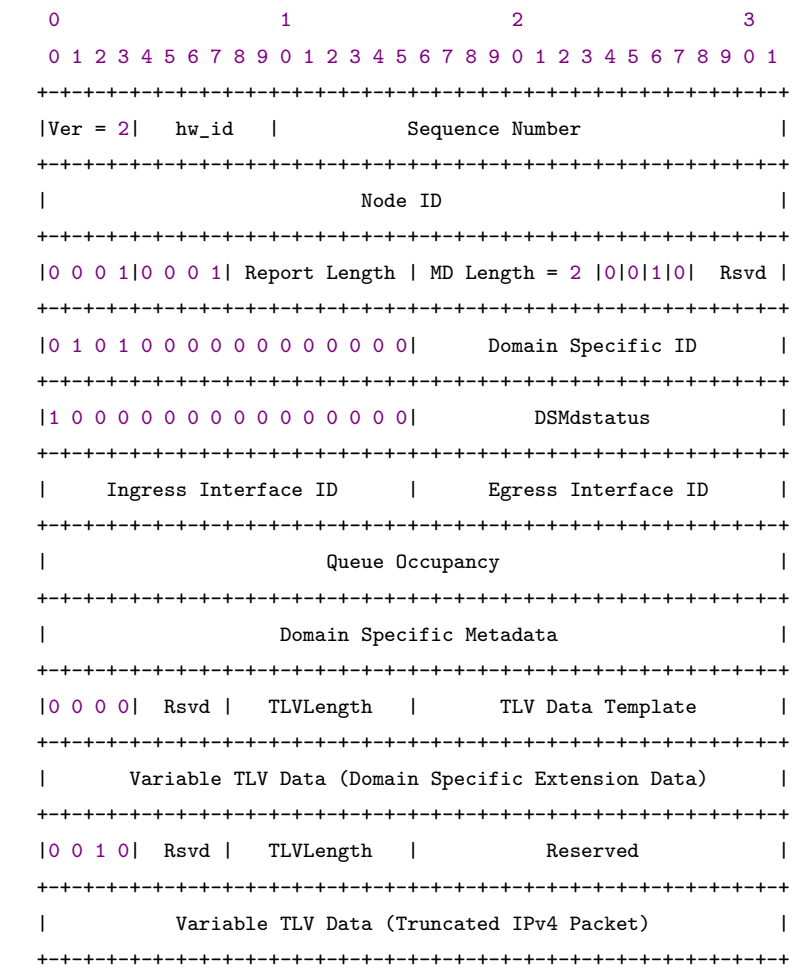


4.2. Example with Baseline Metadata, Domain Specific Metadata, DS Extension Data and Truncated IPv4

This example shows a telemetry report with baseline metadata consisting of level 1 interface IDs and queue occupancy, one domain specific metadata that is 4 bytes long, domain specific extension data and a truncated IPv4 packet.

The values of Telemetry Report header fields in this example are as follows:

- $Ver = 2$
- $RepType = 1$ (INT)
- $InType = 1$ (TLV)
- $MD\ Length = 2$ for level 1 interface IDs and queue occupancy
- $D = 0$ since the packet is not dropped
- $Q = 0$ since the packet does not experience congestion at Switch3
- $F = 1$
- $I = 0$
- The domain specific metadata that is included is represented by the first bit (MSB) of *DsMdBits*
- The first TLV uses $TLVType = 0$ (*Domain Specific Extension Data*)
- The second TLV uses $TLVType = 2$ (*IPv4*)



4.3. Example with Embedded INT-MD in a TCP Packet

This example shows one possibility for the telemetry report corresponding to the INT packet in Section 6.3 of the INT 2.1 specification [2].

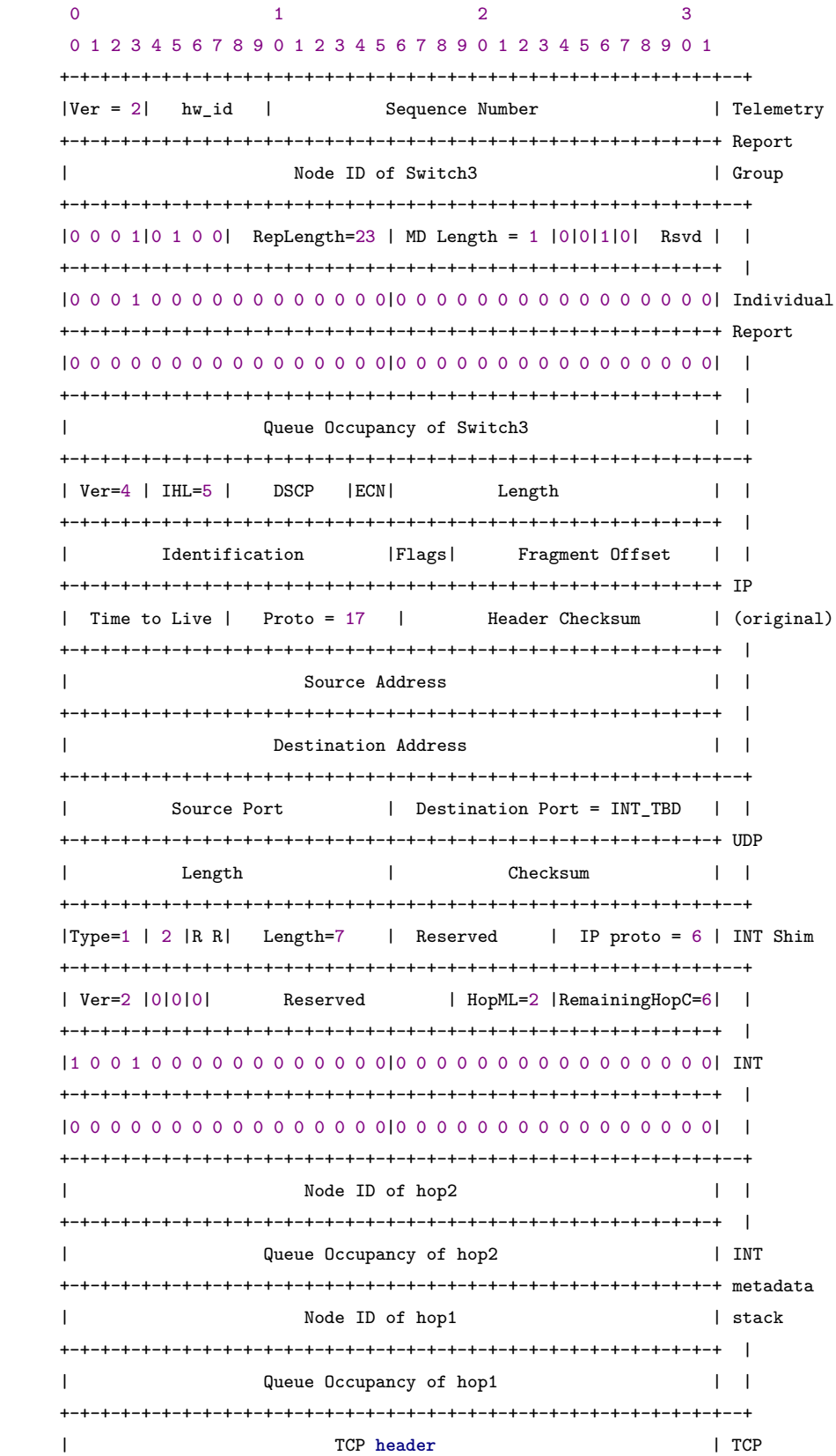
Two hosts (Host1 and Host2) communicate over a network path composed of three network switches (Switch1, Switch2 and Switch3) as shown below.

```
                ==> packet P travels from Host1 to Host2 ==>
Host1 --------> Switch1 ---------> Switch2 ---------> Switch3 --------> Host2
```

- The ToR switch of host1 (Switch1) acts as the INT source. It adds a new UDP header, INT-MD headers and its own metadata in the packet. It requests each INT hop to insert node ID and queue occupancy (For the sake of illustration we only consider node ID and queue occupancy being inserted at each hop. Queue IDs are typically defined per port, hence in a real use-case queue occupancy is likely to be collected along with egress interface ID.)
- The values of INT metadata header fields in this example are as follows:
  - *Ver* = 2
  - $D = 0$ (Packet is not a clone/copy, hence the Sink must not Discard)
  - $E = 0$ (Max Hop Count not exceeded)
  - $M = 0$ (MTU not exceeded at any node)
  - Per-hop Metadata Length = 2 (for node id & queue occupancy)
  - *Remaining Hop Count* starts at 8, decremented by 1 at each hop that inserts INT metadata
- Switch2 prepends its node ID and queue occupancy into the metadata stack.
- The ToR switch of host2 (Switch3) acts as the INT sink, removing the UDP and INT-MD headers before forwarding the packet to host2. It generates a Telemetry Report packet with a single individual report. It inserts its node ID and queue occupancy into the Individual Report Header rather than the embedded INT stack in the truncated packet.
- The values of Telemetry Report header fields in this example are as follows:
  - *Ver* = 2
  - *RepType* = 1 (*INT*)
  - *InType* = 4 (*IPv4*)
  - *MD Length* = 1 for queue occupancy
  - *Report Length* = 2 (individual report fixed size) + 1 (*MD Length*) + 5 (original IP header) + 2 (UDP header for embedded INT) + 1 (INT shim) + 3 (INT fixed) + 4 (INT metadata stack) + 5 (original TCP)
    = 23 (assuming truncated packet fragment ends after the original TCP header)
  - $D = 0$ since the packet is not dropped
  - $Q = 0$ since the packet does not experience congestion at Switch3
  - $F = 1$
  - $I = 0$ since the report includes the metadata for all hops
  - *Domain Specific ID*, *DsMdBits* and *DSMdstatus* are all 0

Below is the telemetry report packet generated by sink Switch3, starting from the Telemetry Report Group Header.

[2] In-band Network Telemetry (INT) Dataplane Specification Version 2.1, May 2020.


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 = 2| hw_id | Sequence Number | Telemetry
Node ID of Switch3 | Group
|0 0 0 1|0 1 0 0| RepLength=23 | MD Length = 1 |0|0|1|0| Rsvd | |
|0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0| Individual
|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0| Report
Queue Occupancy of Switch3 | |
| Ver=4 | IHL=5 | DSCP |ECN| Length | |
Identification |Flags| Fragment Offset | |
| Time to Live | Proto = 17 | Header Checksum | (original)
Source Address | |
Destination Address | |
Source Port | Destination Port = INT_TBD | |
Length | Checksum | | UDP
|Type=1 | 2 |R R| Length=7 | Reserved | IP proto = 6 | INT Shim
| Ver=2 |0|0|0| Reserved | HopML=2 |RemainingHopC=6| |
|1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0| INT
|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0| |
Node ID of hop2 | |
Queue Occupancy of hop2 | INT
Node ID of hop1 | metadata stack
Queue Occupancy of hop1 | |
TCP header | TCP

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4. Example with Embedded INT-MD over UDP in a VXLAN Packet

Two hosts (Host1 and Host2) communicate over a network path composed of three network switches (Switch1, Switch2 and Switch3) as shown below.

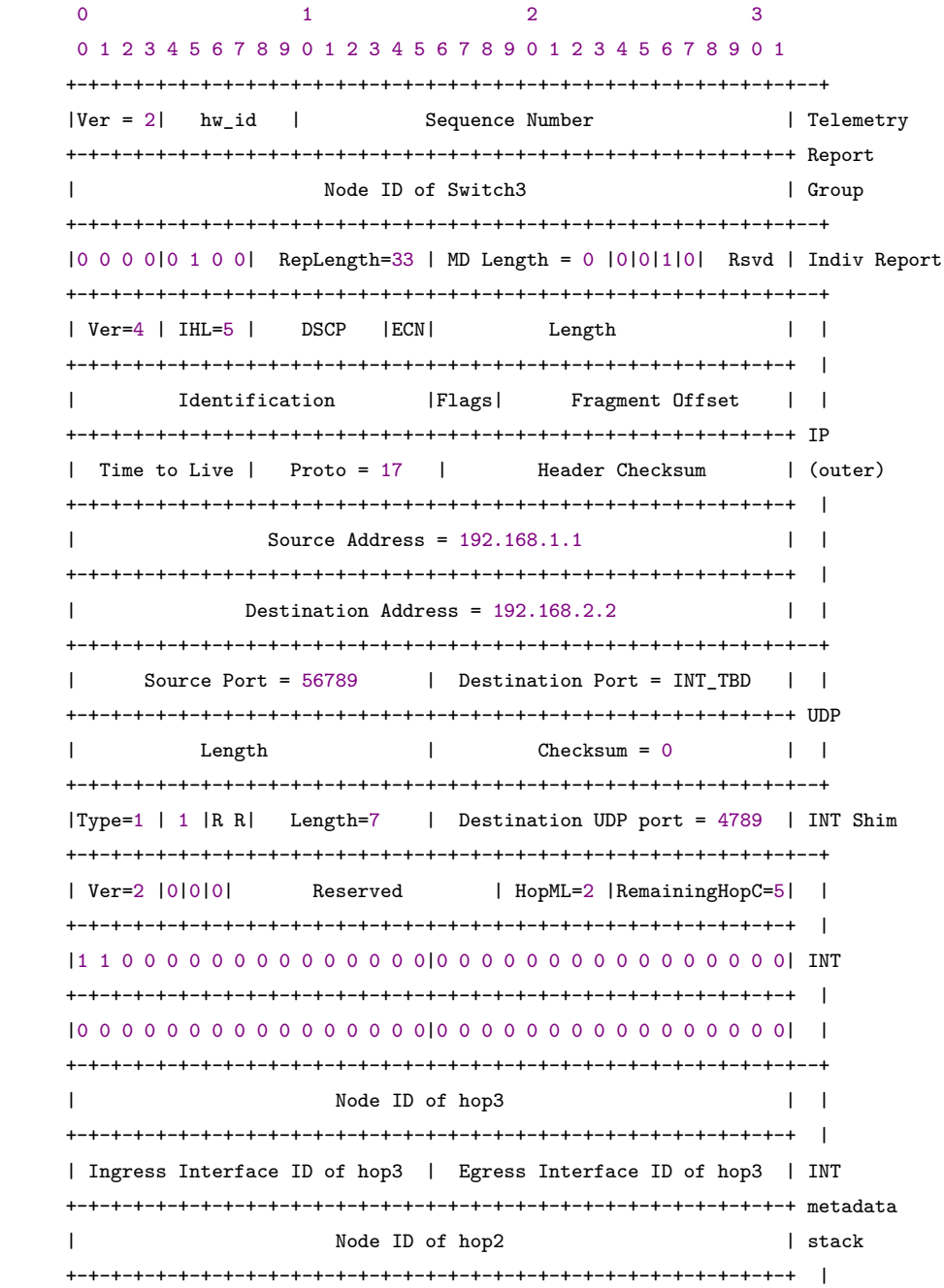
```
                ==> packet P travels from Host1 to Host2 ==>
Host1 --------> Switch1 ---------> Switch2 ---------> Switch3 --------> Host2
```

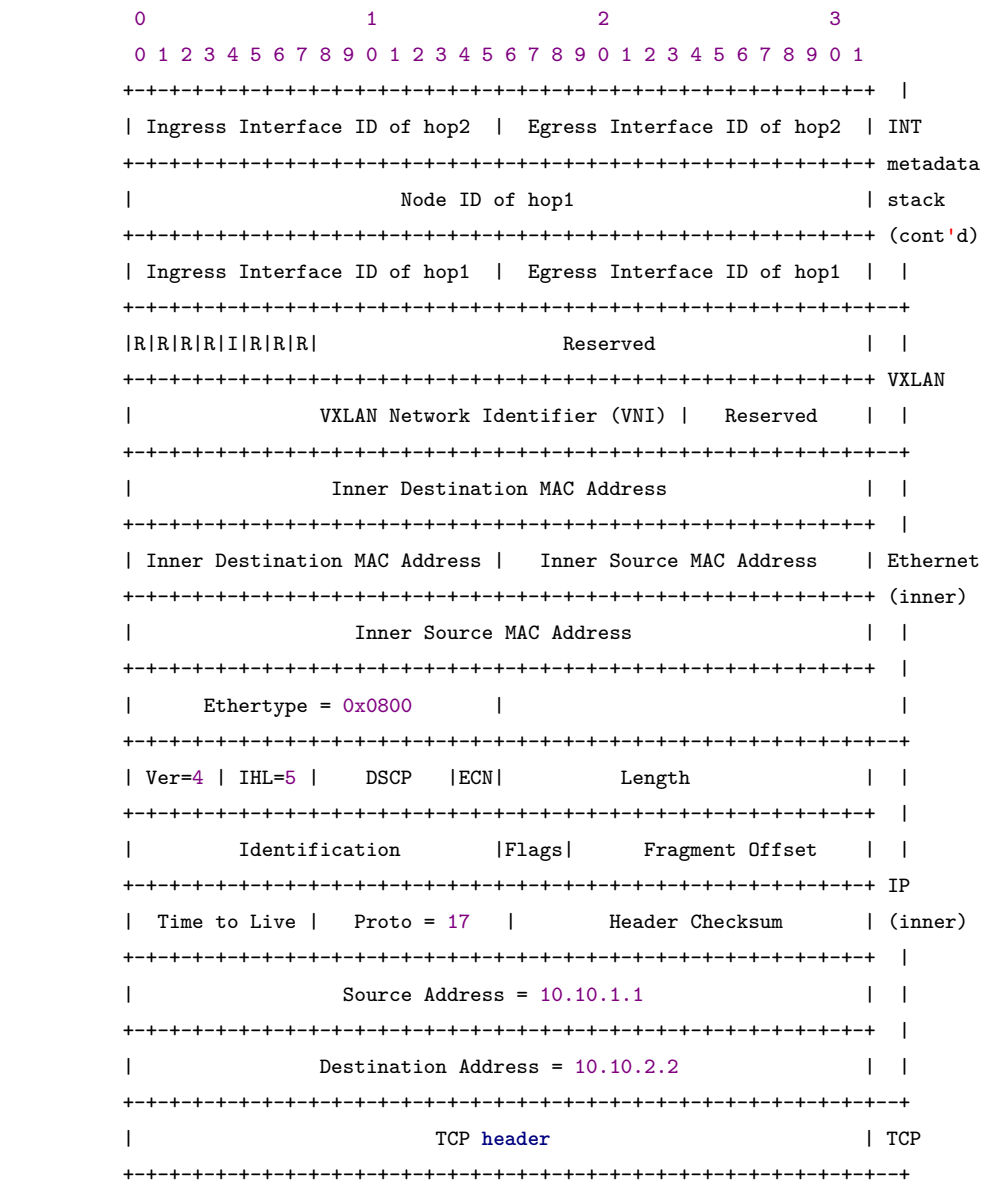
In this example, the two hosts use VXLAN encapsulation. Host1 acts as the VXLAN tunnel endpoint and INT source, inserts VXLAN and INT-MD over UDP headers with instruction bits corresponding to the network state to be reported at intermediate switches. In this example, Host1 and Host2 do not insert any INT metadata. Intermediate switches process the INT-MD header and populate the INT metadata. Host2 acts as INT sink and VXLAN tunnel endpoint, removes the INT-MD and VXLAN headers.

- Host1 sends a VXLAN packet to host2 with inner source IP address 10.10.1.1 (identifying the source workload), inner destination IP address 10.10.2.2 (identifying the destination workload), outer source IP address 192.168.1.1 (identifying host1), outer destination IP address 192.168.2.2 (identifying host2), UDP source port 56789, and UDP checksum 0.
- Host1 acts as the INT source. It inserts INT-MD headers between the outer UDP header and the VXLAN header. It requests each INT hop to insert node ID and level 1 ingress and egress interface IDs.
- The values of INT metadata header fields in this example are as follows:
  - *Ver* = 2
  - *D* = 0 (Packet is not a clone/copy, hence the Sink must not Discard)
  - *E* = 0 (Max Hop Count not exceeded)
  - *M* = 0 (MTU not exceeded at any node)
  - Per-hop Metadata Length = 2 (for node id & level 1 interface IDs)
  - *Remaining Hop Count* starts at 8, decremented by 1 at each hop that inserts INT metadata
- Switch1, Switch2, and Switch3 each prepend their node ID and the relevant ingress and egress interface IDs in the metadata stack.
- Host2 acts as the INT sink and VXLAN tunnel endpoint, removes the INT-MD and VXLAN headers. It generates a Telemetry Report packet with a single individual report. Since it is not adding any of its own metadata, it uses *RepType* value *Inner Only* and *InType* value *IPv4* with the embedded INT stack in the truncated packet.
- The values of Telemetry Report header fields in this example are as follows:
  - *Ver* = 2
  - *RepType* = 0 (*Inner Only*)
  - *InType* = 4 (*IPv4*)
  - *MD Length* = 0
  - *Report Length* = 5 (original outer IP header) + 2 (UDP header) + 1 (INT shim) + 3 (INT fixed) + 6 (INT metadata stack) + 2 (VXLAN) + 4 (inner Ethernet header) + 5 (inner IP header) + 5 (original TCP)
    = 33 (assuming truncated packet fragment ends after the original TCP header)

- $D = 0$ since the packet is not dropped
- $Q = 0$ since the packet does not experience congestion at Switch3
- $F = 1$
- $I = 0$ since the report is being generated by the INT sink, including the metadata for all hops
- *Domain Specific ID*, *DsMdBits* and *DSMdstatus* are all 0

Below is the telemetry report packet generated by Host2, starting from the Telemetry Report Group Header. We use INT_TBD for UDP.Destination_Port to indicate the existence of INT headers.





A. Acknowledgements

We thank the following individuals for their contributions to this specification.

- Gordon Brebner
- Mukesh Hira
- Jeongkeun Lee
- Randy Levensalor
- Ramesh Sivakolundu
- Mickey Spiegel

B. Change log

- 2017-11-10
  - Initial release
- 2017-11-10
  - **Tag v0.5 spec**
- 2018-2-14
  - Promote *Switch id* to fixed portion of the Telemetry Report Header
    - The Switch id is always present.
  - Flexible format allowing for arbitrary combinations of optional telemetry metadata in the Telemetry Report Header
    - Replaces previous Telemetry Drop Header and Telemetry Switch Local Report Header
    - Adds a 4 bit length field indicating the Telemetry Report Header length in multiples of 4 octets
    - Adds a bitmap indicating which optional metadata is present in the Telemetry Report Header
    - Rearranges fields in the first 32 bits of the Telemetry Report Header in order to achieve proper alignment, and to place reserved bits between the report metadata bitmap and the association bits so that either one can expand as necessary
- 2018-04-03
  - Editorial changes for v1.0
- 2018-04-20
  - **Tag v1.0 spec**
- 2019-07-19
  - Added INT modes of operation and nomenclature: INT-XD/MX/MD
- 2020-04-06
  - Revised Telemetry Report header 2.0 format supporting:
    - coalescing of multiple reports in one packet
    - support for domain specific extensions
    - some new RepMdBits codepoints
    - Increased timestamp size to 8 bytes
  - Changed *Switch id* terminology to *Node ID*
- 2020-05-18
  - Separated the Telemetry Report Header description into separate sections for 'Telemetry

Report Group Header' (first 8 octets that appear once in a packet, and 'Individual Report Header' that may be repeated, one per report in the packet
  - Reworded section on Embedded Telemetry Metadata to match v2.0 format, where truncated packet fragments reside within each Individual Report Header
  - Changed 'switch' and 'device' to 'node' where appropriate
  - Reserved Domain Specific ID value 0x0000 as default well known ID
  - Added RepType codepoint for 'Inner Only'

- 2020-05-25
  - Added 2 examples of telemetry reports with embedded telemetry metadata

- 2020-06-11
  - Simplify metadata processing by removing padding option, relying on clearing of bits instead.
  - **Tag v2.0 spec**