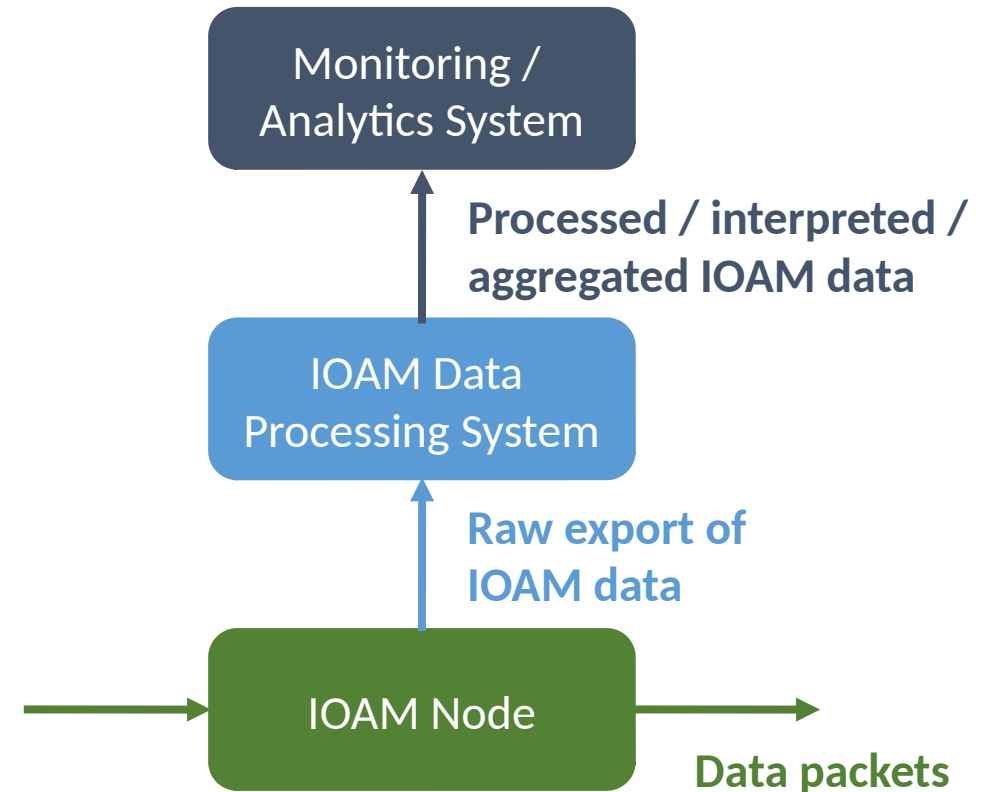


# IOAM Raw Data Export with IPFIX

[draft-spiegel-ippm-ioam-rawexport-02](#)

# Raw Export of IOAM Data

- IOAM records operational and telemetry information in a packet while the packet traverses a path between two points in the network
- *Raw export* refers to a mode of operation where a node exports the IOAM data as it is received in the packet, typically at the edge of a network domain
- The exporting node neither interprets, aggregates nor reformats the IOAM data before it is exported
  - Off-load from the node which performs data plane operations
  - Implementable at billions of packets per second



# Using IPFIX for Raw Export of IOAM

- IPFIX is defined as a generic export protocol that can export any Information Element(s) as long as they are described in the information model
- IPFIX protocol is well suited for and is defined as the protocol for exporting packet samples in [RFC5476]
  - IPFIX/PSAMP already define many of the information elements needed for exporting raw sections of packets, needed for deriving context
- Many network devices may be quite constrained in the IPFIX formats that they can support for IOAM export at high speeds
  - Small number of information elements supported
  - Alignment along 4 octet boundaries
    - Alignment constraint varies depending on the encoding used for packet snippets:
      - Fixed length :  $4n$
      - Variable length < 255 :  $4n + 3$
      - Variable length < 65536 :  $4n + 1$
  - Possibly a limited number of fixed templates

# Key IPFIX IEs Leveraged for IOAM Raw Data Export

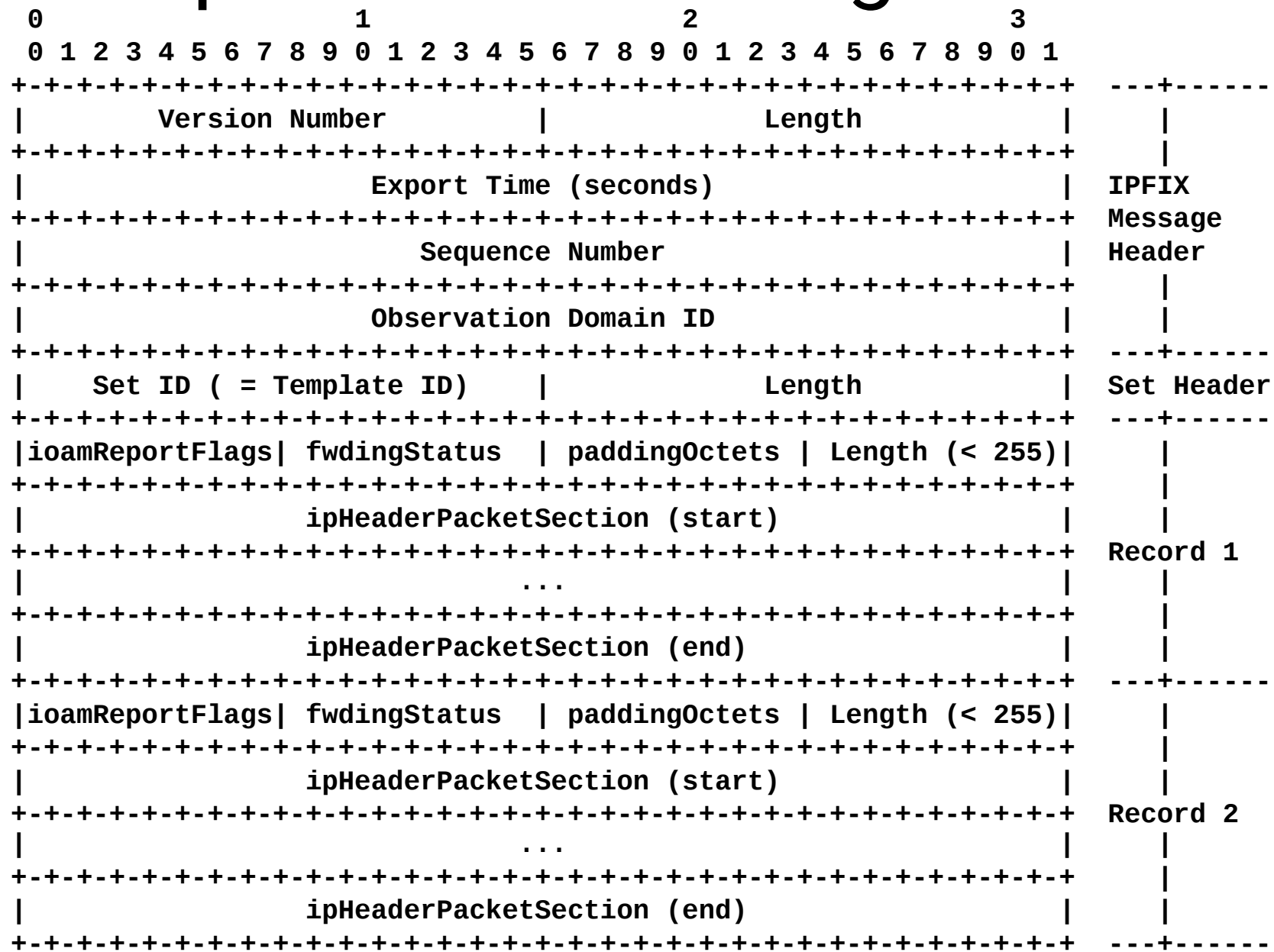
313	ipHeaderPacketSection	octetArray	default	This Information Element carries a series of n octets from the IP header of a sampled packet ... With sufficient length, this element also reports octets from the IP payload. ...
315	dataLinkFrameSection	octetArray	default	This Information Element carries n octets from the data link frame of a selected frame ... Further Information Elements, i.e., dataLinkFrameType and dataLinkFrameSize, are needed to specify the data link type and the size of the data link frame of this Information Element. ...
408	dataLinkFrameType	unsigned16	flags	This Information Element specifies the type of the selected data link frame. The following data link types are defined here: - 0x01 IEEE802.3 ETHERNET [IEEE802.3] - 0x02 IEEE802.11 MAC Frame format [IEEE802.11]
410	sectionExportedOctets	unsigned16	quantity	This Information Element specifies the observed length of the packet section ... when padding is used. ...
89	forwardingStatus	unsigned8	identifier	This Information Element describes the forwarding status of the flow and any attached reasons. ...

# Proposed new IPFIX Information Elements

ioamReportFlags	unsigned8	flags	This IE describes properties associated with an IOAM report. bit   description 0 Dropped Assn – Dropped packet of interest 1 Congested Queue Association – Indicates the presence of congestion on a monitored queue 2 Tracked Flow Association – Matched a flow of interest
ioamEncapsulationType	unsigned8	identifier	0 None: [ID.ietf-ippm-ioam-data] 1 GRE: [ID.weis-ippm-ioam-eth] 2 IPv6: [ID.ioametal-ippm-6man-ioam-ipv6-options] 3 VXLAN-GPE: [ID.brockners-ippm-ioam-vxlan-gpe] 4 GENEVE Option: [ID.brockners-ippm-ioam-geneve] 5 GENEVE Next Protocol: [ID.weis-ippm-ioam-eth] 6 NSH: [ID.ietf-sfc-ioam-nsh]
ioamPreallocatedTraceData	octetArray	default	This IE carries n octets of IOAM Preallocated Trace data. *
ioamIncrementalTraceData	octetArray	default	This IE carries n octets of IOAM Incremental Trace data. *
ioamE2EData	octetArray	default	This IE carries n octets of IOAM E2E data. *
ioamPOTData	octetArray	default	This IE carries n octets of IOAM POT data. *

\* When ioamEncapsulationType is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload

# Example: Variable Length < 255 IP Packet

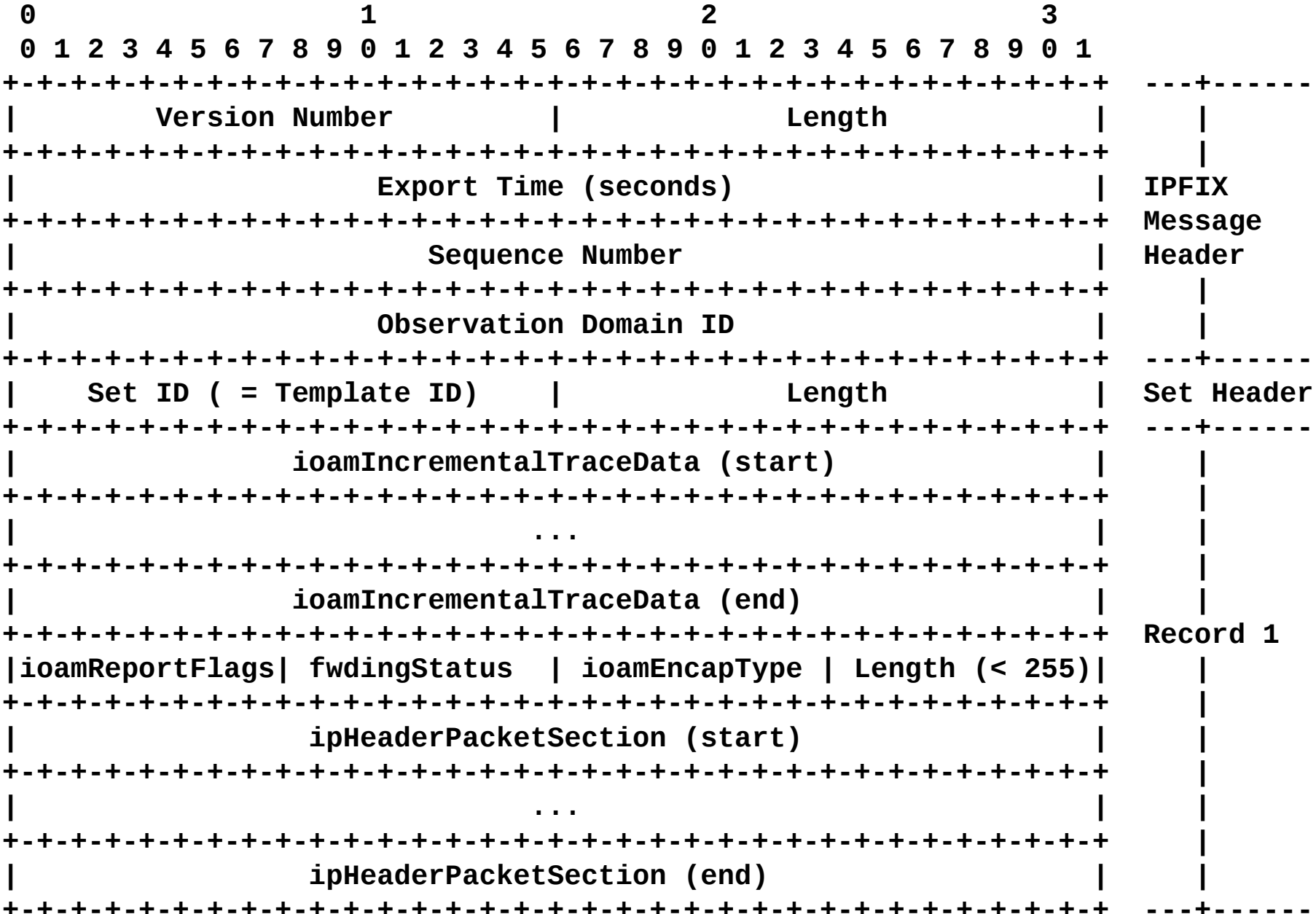


IPFIX  
Message  
Header  
Set Header

Record 1  
  
Record 2

IOAM data is part of  
ipHeaderPacketSection

# Example: Variable Length IP Packet w/ Outer Incremental Trace



# New IEs to optimize IP / Ethernet packet sections

- Two new information elements that optimize use of IP packet section and Ethernet frame section
  - Templates can use existing *ipHeaderPacketSection* or new *ipHeaderPacketSectionWithPadding*
    - First example uses *ipHeaderPacketSection*
    - Other examples use *ipHeaderPacketSectionWithPadding*
  - Templates can use existing *dataLinkFrameSection* or new *ethernetFrameSection*
    - Example changed from *dataLinkFrameSection* to *ethernetFrameSection*



# ipHeaderPacketSectionWithPadding Differences

sectionExportedOctets exists?	Fixed vs variable length	ipHeaderPacketSection: Padding allowed?	ipHeaderPacketSectionWithPadding: Padding allowed?
Yes	Fixed length	Yes	Yes
Yes	Variable length	?	Up to 3 octets of padding *
No	Fixed length	No	Yes, if the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding
No	Variable length	No	Up to 3 octets of padding *, if the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding

\* used to preserve 4-octet alignment of subsequent IEs or subsequent records within the same set

# Next Steps

- Should this effort progress in OPSAWG or IPPM working group?
- Soliciting technical input/feedback

Backup Slides

# ipHeaderPacketSectionWithPadding Text

- First 3 paragraphs (IP header contents, use of sectionOffset, payload and security considerations) are identical to *ipHeaderPacketSection*
- Use of padding with fixed or variable length IE is different:
  - When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.
  - When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.
  - In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

# ethernetFrameSection

- Allows padding similar to *ipHeaderPacketSectionWithPadding*
  - Last 3 paragraphs are the same except for *IP packet*  $\square$  *Ethernet frame*
- Eliminates need to include *dataLinkFrameType* and *dataLinkFrameSize*
- First paragraph text:

This Information Element carries a series of n octets from the IEEE 802.3 Ethernet frame of a sampled packet, starting after the preamble and start frame delimiter (SFD), plus sectionOffset octets into the frame if there is a sectionOffset field corresponding to this Information Element.
- Second paragraph (payload and security considerations) is the same as *dataLinkFrameSection* and *ipHeaderPacketSection*