

Export of Encapsulation Layer Information in IPFIX

draft-liu-opsawg-ipfix-muti-layer

Yao Liu ZTE (presenter)

Taoran Zhou ZTE

OPSAWG

IETF#125

Mar, 2026

Quick Recap

- **Application Scenarios:** Packets with multi-layer network encapsulation become more and more common in the network, typical scenarios include IP-in-IP, even IP-in-IP-in-IP
- **Potential Requirements for IPFIX**
 - Req-a: Collecting the same fields from both the outer and inner headers
 - Req-b: Collecting only the fields from the inner packet header
 - Req-c: Collecting only the fields from the outer packet header
 - Req-d: Collecting different fields from the outer header (e.g, outer DA) and inner header(e.g.inner SA)
- **Gaps**
 - Gaps for Req-b/c/d: When receiving a IPFIX message with a certain IE(e.g, sourceIPv6Address) , the collector is not able to tell which encapsulation layer this IE belongs to.

Option 1: Dedicated Encapsulation Layer Indication IEs for Each Layer

New IEs as Encapsulation Layer Indicator

- encapLayerTop: A 16-bit identifier. Fixed value 0xFFFF
 - The IEs follows immediately after it till the next Encapsulation Layer IE belong to the outmost network encapsulation layer (e.g, from the outmost Ethernet header to the first IP header).
- encapLayer2: A 16-bit identifier. Fixed value 0xFFFF
 - The IEs follows immediately after it till the next Encapsulation Layer IE belong to the second network encapsulation layer.
- encapLayer3:

Pros:

- Straightforward, seems workable

Cons:

- The semantic of one IPFIX Information Element (e.g, destIPv6address) relies on the content of another IE(e.g, encapLayerTop, encapLayer2)
- More IEs may be needed if there're packets with more than three layers of encapsulation

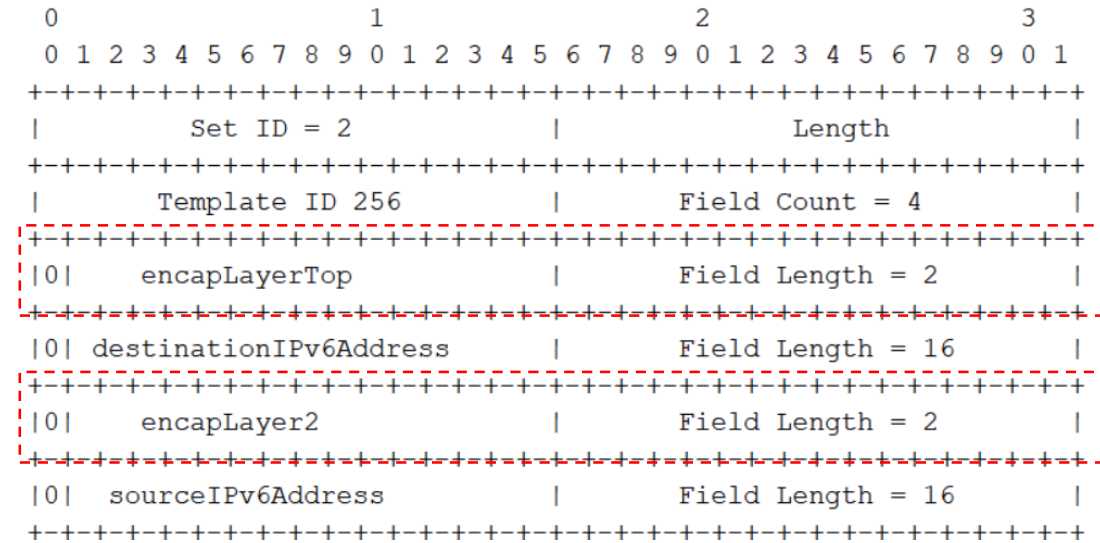


Fig 1 Template for Flow Record

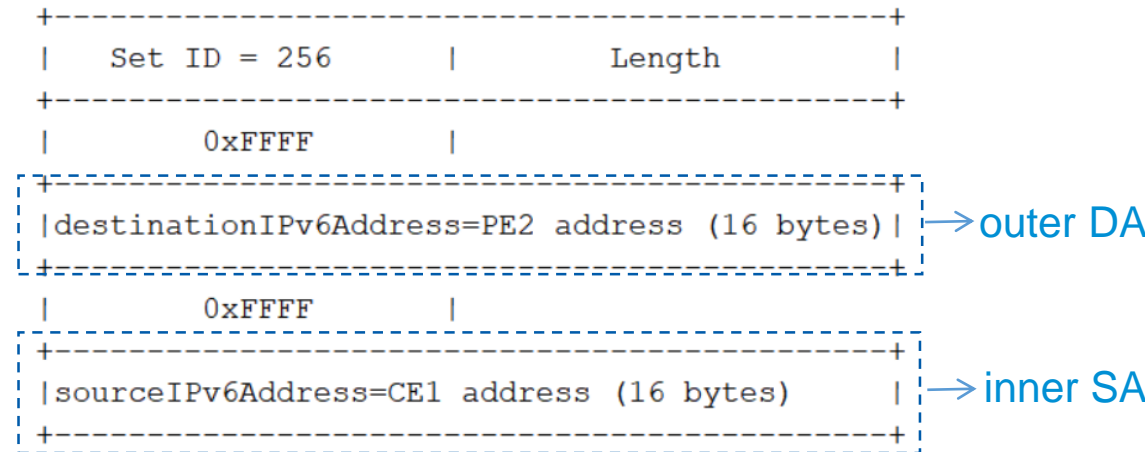


Fig 1 Data Record

Option 2: Uniformed Structured Data Encoding

One New IE

- Name: multiLayerEcapFields
- Description:
 - The IE indicates that the header fields of different encapsulation layers will be exported. Each top-level element in a subTemplateMultiList Information Element carries a Template ID, Length, and zero or more Data Records corresponding to the Template ID.
 - The template IDs from top to bottom carried in the IE correspond to encapsulation layers of the packet flow, starting from the outermost layer
 - If not all of the encapsulation layers are required to be exported, the template ID in the data record and the corresponding length field are set to 0 to indicate the absence of the information of this layer.
- Abstract Data Type: **subTemplateList**
- Data Type Semantics: list

Pros:

- Generic, standard

Cons:

- There's no wide implementation of the type subTemplateList

```

+++++
|          Set ID = 2          |          Length = 12          |
+++++
|          Template ID = 300   |          Field Count = 1       |
+++++
|0| IE = multiLayerEcapFields |          Field Length=65535    |
+++++
    
```

Fig 1 Encoding multiLayerEcapFields, Template for Flow Record

```

+++++
|          Set ID = 3          |          Length = 12          |
+++++
|          Template ID = 301   |          Field Count = 1       |
+++++
|0|IE=destinationIPv6Address(28)|          Field Length = 16    |
+++++
    
```

Fig 2 Template for the outmost IPv6 header

```

+++++
|          Set ID = 4          |          Length = 12          |
+++++
|          Template ID = 302   |          Field Count = 1       |
+++++
|0| IE = sourceIPv6Address(27) |          Field Length = 16    |
+++++
    
```

Fig 3 Template for the third IPv6 header

```

+++++
|          Set ID = 300        |          Length = N          |
+++++
|          255                |          List Length= 44      |          semantic            |
+++++
| Top  templateId=301         |          Length = 16         |
+++++
~                               IPv6 DA                               ~
+++++
| Second templateId=0         |          Length = 0          |
+++++
| Third templateId=302        |          Length = 16         |
+++++
~                               IPv6 SA                               ~
+++++
    
```

Fig 4 Encoding multiple-layer headers, Data Set

Option 3: Always Export the Whole Encapsulation

- [RFC7011]: If an Information Element is required more than once in a Template, the different occurrences of this Information Element **SHOULD** follow the logical order of their treatments by the Metering Process...For example, when exporting the two source IP addresses of an IPv4-in-IPv4 packet, the first sourceIPv4Address Information Element occurrence should be the IPv4 address of the outer header, while the second occurrence should be the address of the inner header.
- Using this option, for a packet with three IPv6 headers encapsulated, even if the monitor only wants to collect the DA of the innermost IPv6 header, the exporter needs to export all the three DAs of each layer in order.

Pros:

- No New IEs required
- Easy to implement technically

Cons:

- Redundancy in the IPFIX messages (requirement b/c/d can't be fulfilled)
- For this option, should the "SHOULD" in RFC7011 be updated to "MUST" ? ==> But RFC6313 says "some encoding optimizations are based on the permutation of Information Element order".

Next Steps

- Welcome feedback, comments and cooperation.
 - Which option is better ?
 - Any other options ?

Thank You !

Backup

➤ An example of IP-in-IP-in-IP flows

