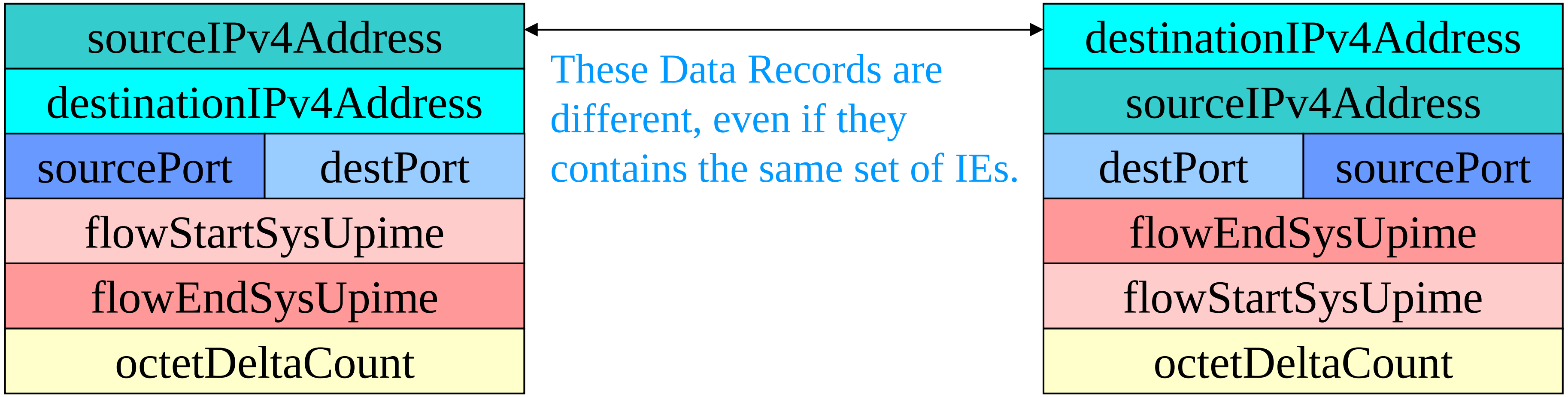
Guideline for the order of Information Elements
draft-irino-ipfix-ie-order-04.txt

NTT Networks Service Lab.

Hitoshi Irino

About this draft

- Motivation
  - Templates can define data structures of Data Records freely.
  - This feature allows various Templates even if they contain the same set of Information Elements (IEs).



sourceIPv4Address

destinationIPv4Address

sourcePort destPort

flowStartSysUpime

flowEndSysUpime

octetDeltaCount

These Data Records are different, even if they contains the same set of IEs.

destinationIPv4Address

sourceIPv4Address

destPort sourcePort

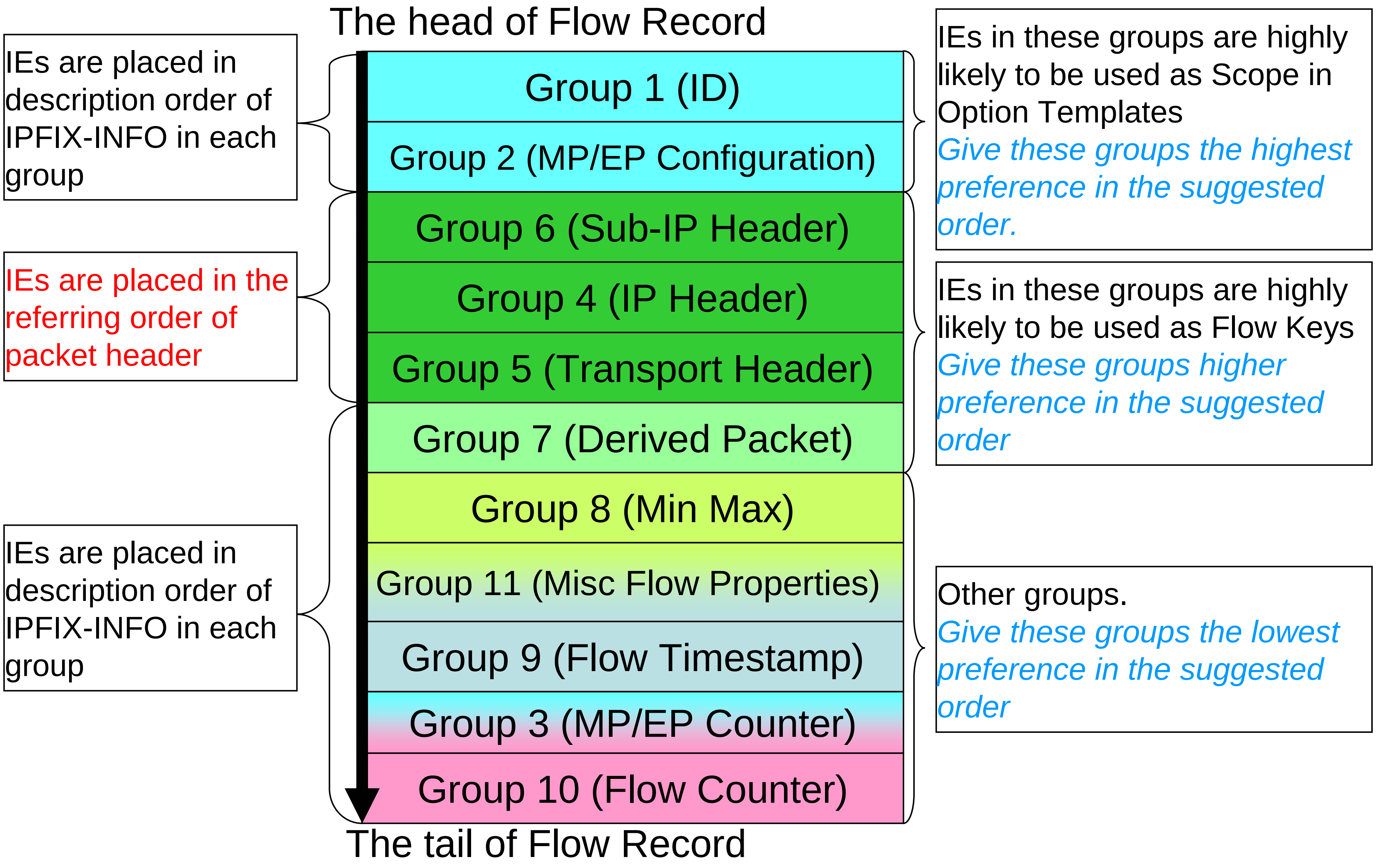
flowEndSysUpime

flowStartSysUpime

octetDeltaCount

- It is not efficient to allow various templates containing the same IEs.
  - e.g., Template management
- This draft provides a fundamental rule about the order of IEs
  - to prevent multiple Templates and Data Records being made from the same set of IEs.
  - without losing the flexibility of the Template mechanism.

The proposed order of Information Elements in the draft



The head of Flow Record

IEs are placed in description order of IPFIX-INFO in each group

IEs are placed in the referring order of packet header

IEs are placed in description order of IPFIX-INFO in each group

Group 1 (ID)

Group 2 (MP/EP Configuration)

Group 6 (Sub-IP Header)

Group 4 (IP Header)

Group 5 (Transport Header)

Group 7 (Derived Packet)

Group 8 (Min Max)

Group 11 (Misc Flow Properties)

Group 9 (Flow Timestamp)

Group 3 (MP/EP Counter)

Group 10 (Flow Counter)

The tail of Flow Record

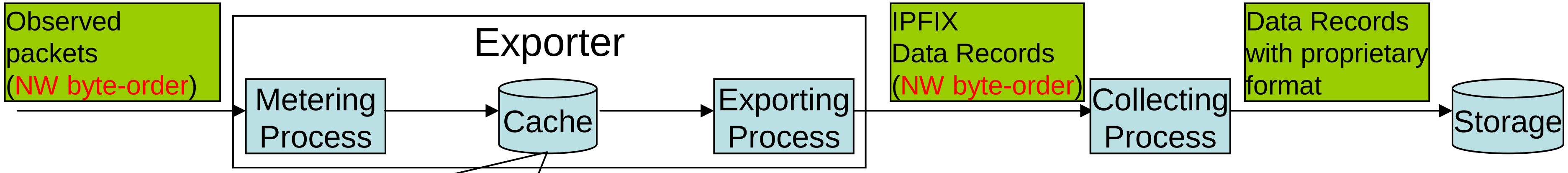
IEs in these groups are highly likely to be used as Scope in Option Templates
Give these groups the highest preference in the suggested order.

IEs in these groups are highly likely to be used as Flow Keys
Give these groups higher preference in the suggested order

Other groups.
Give these groups the lowest preference in the suggested order

Update in the newest draft

- The newest draft adds description about effective usage and ideas for increasing performance.
- In the environment where IPFIX devices use the same order of IEs:
  1. Copy method for multiple adjacent IEs is enabled
     - Conditions for the copy method:
       - – Data lengths of IEs are the same in copy source and destination.
         - » Most IEs, whose values are derived from observed packet header fields, have a fixed length.
       - – Byte-order of values of IEs are the same in copy source and destination.
         - » Exporters can satisfy this condition for values derived from observed packet header fields.


Observed packets (NW byte-order)
Exporter
Metering Process
Cache
Exporting Process
IPFIX Data Records (NW byte-order)
Collecting Process
Data Records with proprietary format
Storage

When data in the cache for characteristic properties of flows
- uses the same data structure of the packet headers.
- uses Network byte-order.

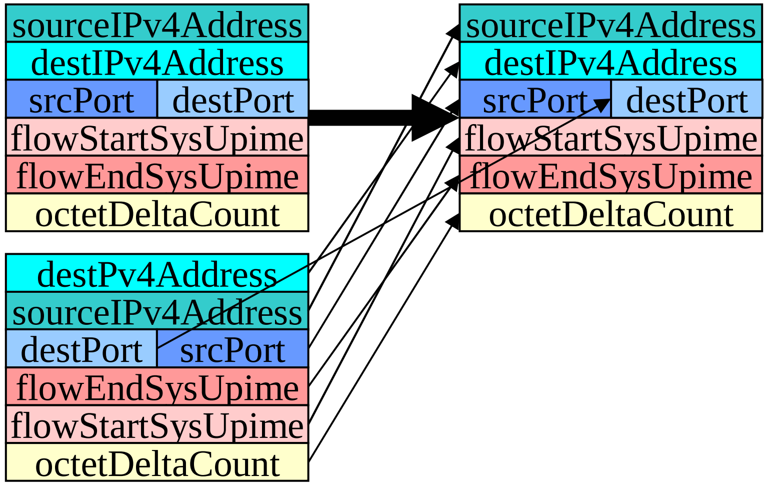
⇩

- The number of copies can be minimized from MP to cache and from cache to EP.
- (Moreover MP can compare observed packets and Data Records in cache efficiently.)

Update in the newest draft (cont.)

2. Merging Data Records based on multiple Templates to a data structure.
    - When a collector stores collected Data Records based on multiple Templates into their proprietary format.
    - When an Aggregation Process of Mediators aggregates collected Data Records based on multiple Templates to Data Records based on a Template for exporting.

Merging Data Records based on multiple Templates that use a different order

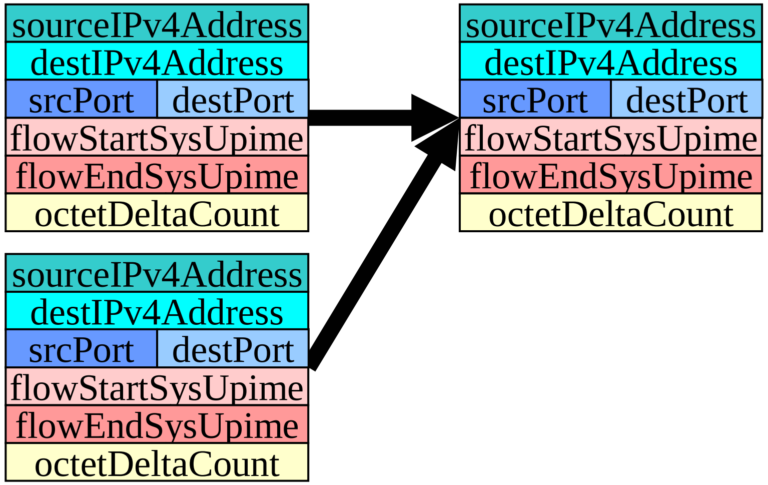


sourceIPv4Address
destIPv4Address
srcPort destPort
flowStartSysUpime
flowEndSysUpime
octetDeltaCount

destPv4Address
sourceIPv4Address
destPort srcPort
flowEndSysUpime
flowStartSysUpime
octetDeltaCount

sourceIPv4Address
destIPv4Address
srcPort destPort
flowStartSysUpime
flowEndSysUpime
octetDeltaCount

Merging Data Records based on multiple Templates that use the same order
(ideal situation)



sourceIPv4Address
destIPv4Address
srcPort destPort
flowStartSysUpime
flowEndSysUpime
octetDeltaCount

sourceIPv4Address
destIPv4Address
srcPort destPort
flowStartSysUpime
flowEndSysUpime
octetDeltaCount

sourceIPv4Address
destIPv4Address
srcPort destPort
flowStartSysUpime
flowEndSysUpime
octetDeltaCount

Summary and next step

- The proposed order can be effective for increasing the performance.
  - It was designed by referring to the order of the fields in the packet header.
- The newest draft adds descriptions about effective usage and ideas for increasing performance.

1. Introduction
   - 1.1. Problem statement
   - 1.2. Purposes of this draft
2. Terminology
3. Approach to the ordering of Information Elements
   - 3.1. Order of Information Element groups
   - 3.2. Padding
   - 3.3. Enterprise-specific Information Elements
4. Recommended order of Information Elements
5. Applications of order of Information Elements
   - 5.1. Copy method for multiple adjacent Information Elements
   - 5.2. Merging with two or more Templates
6. Security considerations
7. IANA considerations
8. References

- Next Step:
  - The order have to be more detailed.
  - I’m looking for any person who implements these idea.
  - I hope it is discussed in WG when charter will be changed in future…

Supplementary materials

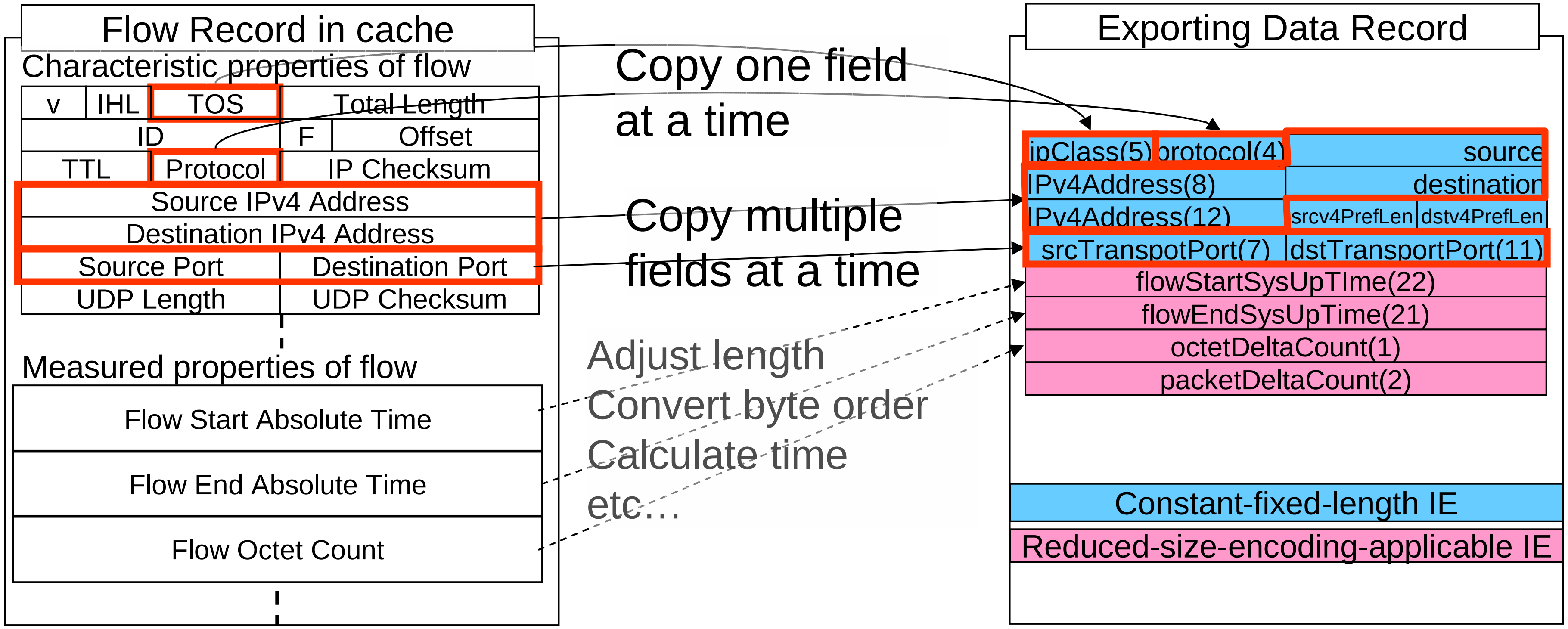
Efficiency of the proposed order

- The reason for proposing an unified order
  - It is efficient for data copies between 2 processes.
  - The situation that 2 processes use same order is about 60% faster than another situation that 2 processes use different order.
    (a similar evaluation is introduced in IETF69:
    http://www.ietf.org/proceedings/07jul/slides/ipfix-10.pdf)

- The reason for proposed order referring order of fields in observed packet header
  - It is not only above copy issue but also efficient for observed packets and stored flows in MP.
    - This draft is proposed on the assumption that MP stores packets headers in cache.
      - When Exporters supports all IEs of RFC5102, MP have to be able to store the all fields in packet headers at least.
    - On the assumption, performance is increased about 40% in maximum.
      (This issue is introduced in Flocon2008:
      http://www.cert.org/flocon/2008/presentations/flocon2008-irino-katayama.pdf)

Example of copy method for multiple fields

- Conditions for copying multiple fields
  - Flow Record in cache and Exporting Data Record must use the same order.
  - IEs must have a constant fixed length.
    - Almost all IE characterizing properties of flow are constant fixed length.
  - Byte-orders must be the same.
    - Observed packet and Exporting Data Records use network byte order.
  - IEs for copying multiple fields must be adjacent.



Flow Record in cache

Characteristic properties of flow

v | IHL | TOS | Total Length

ID | F | Offset

TTL | Protocol | IP Checksum

Source IPv4 Address

Destination IPv4 Address

Source Port | Destination Port

UDP Length | UDP Checksum

Measured properties of flow

Flow Start Absolute Time

Flow End Absolute Time

Flow Octet Count

Copy one field at a time

Copy multiple fields at a time

Adjust length
Convert byte order
Calculate time
etc…

Exporting Data Record

ipClass(5) | protocol(4) | source

IPv4Address(8) | destination

IPv4Address(12) | srcv4PrefLen | dstv4PrefLen

srcTranspotPort(7) | dstTransportPort(11)

flowStartSysUpTIme(22)

flowEndSysUpTime(21)

octetDeltaCount(1)

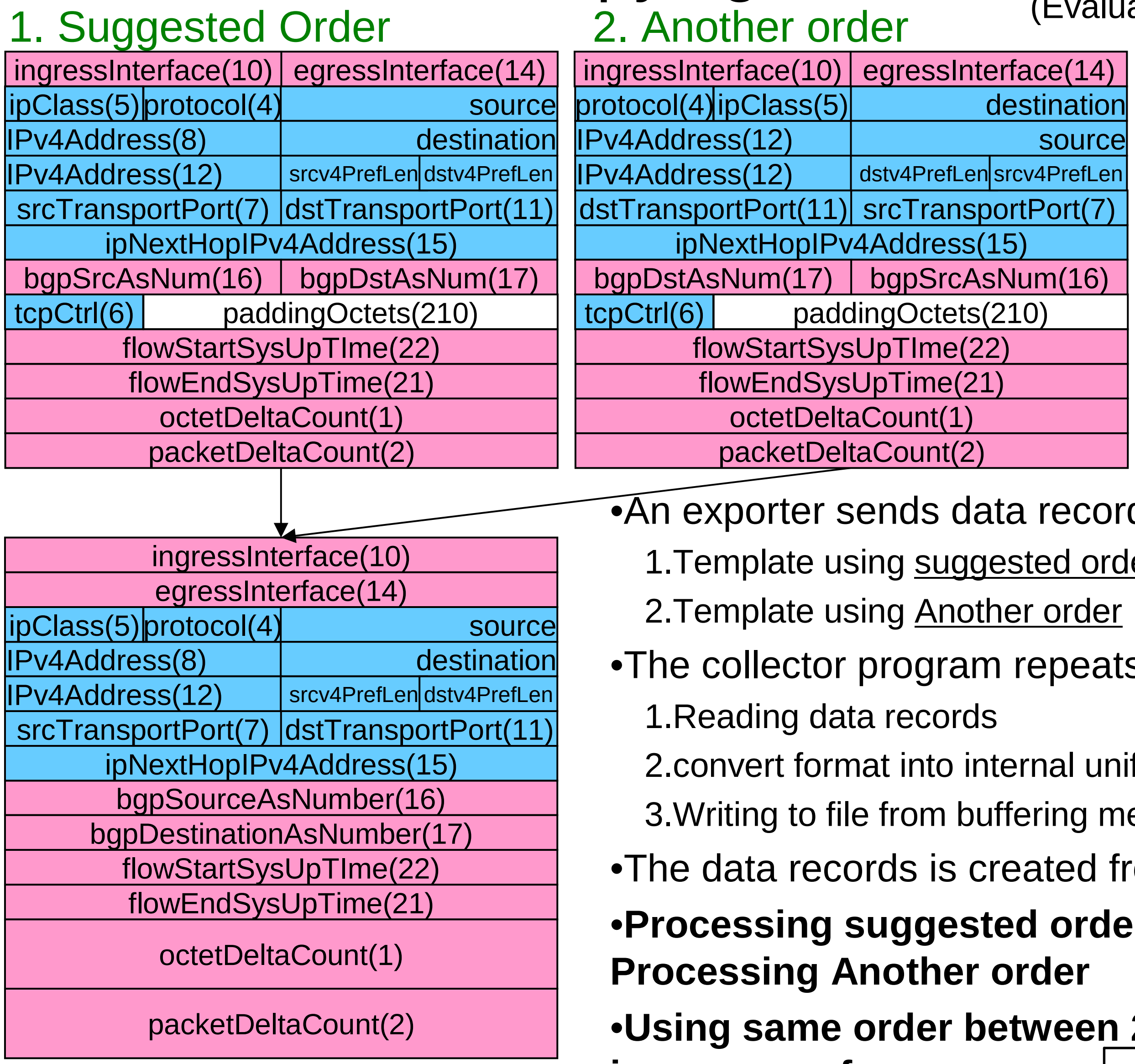
packetDeltaCount(2)

Constant-fixed-length IE

Reduced-size-encoding-applicable IE

Performance of copying IEs between an EP and a CP

(Evaluation with suggested order of -04 draft.)



1. Suggested Order

ingressInterface(10) | egressInterface(14)
ipClass(5) | protocol(4) | source
IPv4Address(8) | destination
IPv4Address(12) | srcv4PrefLen | dstv4PrefLen
srcTransportPort(7) | dstTransportPort(11)
ipNextHopIPv4Address(15)
bgpSrcAsNum(16) | bgpDstAsNum(17)
tcpCtrl(6) | paddingOctets(210)
flowStartSysUpTIme(22)
flowEndSysUpTime(21)
octetDeltaCount(1)
packetDeltaCount(2)

2. Another order

ingressInterface(10) | egressInterface(14)
protocol(4) | ipClass(5) | destination
IPv4Address(12) | source
IPv4Address(12) | dstv4PrefLen | srcv4PrefLen
dstTransportPort(11) | srcTransportPort(7)
ipNextHopIPv4Address(15)
bgpDstAsNum(17) | bgpSrcAsNum(16)
tcpCtrl(6) | paddingOctets(210)
flowStartSysUpTIme(22)
flowEndSysUpTime(21)
octetDeltaCount(1)
packetDeltaCount(2)

ingressInterface(10)
egressInterface(14)
ipClass(5) | protocol(4) | source
IPv4Address(8) | destination
IPv4Address(12) | srcv4PrefLen | dstv4PrefLen
srcTransportPort(7) | dstTransportPort(11)
ipNextHopIPv4Address(15)
bgpSourceAsNumber(16)
bgpDestinationAsNumber(17)
flowStartSysUpTIme(22)
flowEndSysUpTime(21)
octetDeltaCount(1)
packetDeltaCount(2)

🟦 Constant Fixed Length IE

🟥 Reduced Size Encoding applicable IE

- An exporter sends data records using 2 templates
  1. Template using suggested order
  2. Template using Another order
- The collector program repeats as follows:
  1. Reading data records
  2. convert format into internal unified format and buffering
  3. Writing to file from buffering memory
- The data records is created from about 7million packets
- **Processing suggested order is about 64 % faster than Processing Another order**
- **Using same order between 2 processes is important to increase performance.**

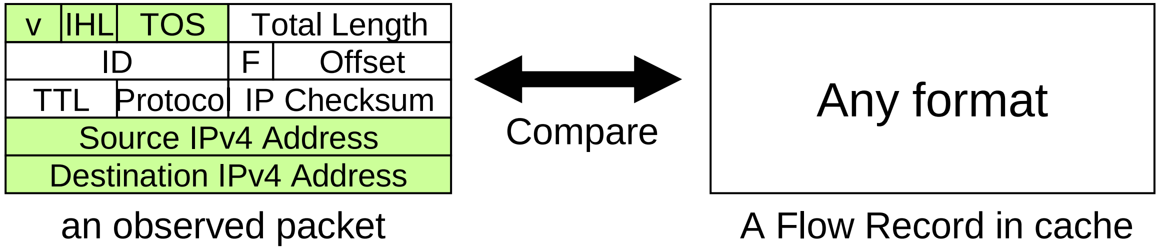
Collector's internal unified format uses suggested order

	Suggested	Another
Ave.	0.4533(sec)	0.7452 (sec)
ratio	1	1.64

Comparison method for multiple fields in MP

Example: Flow Key: Version, IHL, TOS, source Address, destination Address

All fields are compared every time (general approach)



v	IHL	TOS	Total Length	
ID			F	Offset
TTL		Protoco	IP Checksum	
Source IPv4 Address				
Destination IPv4 Address				

an observed packet

Any format

A Flow Record in cache

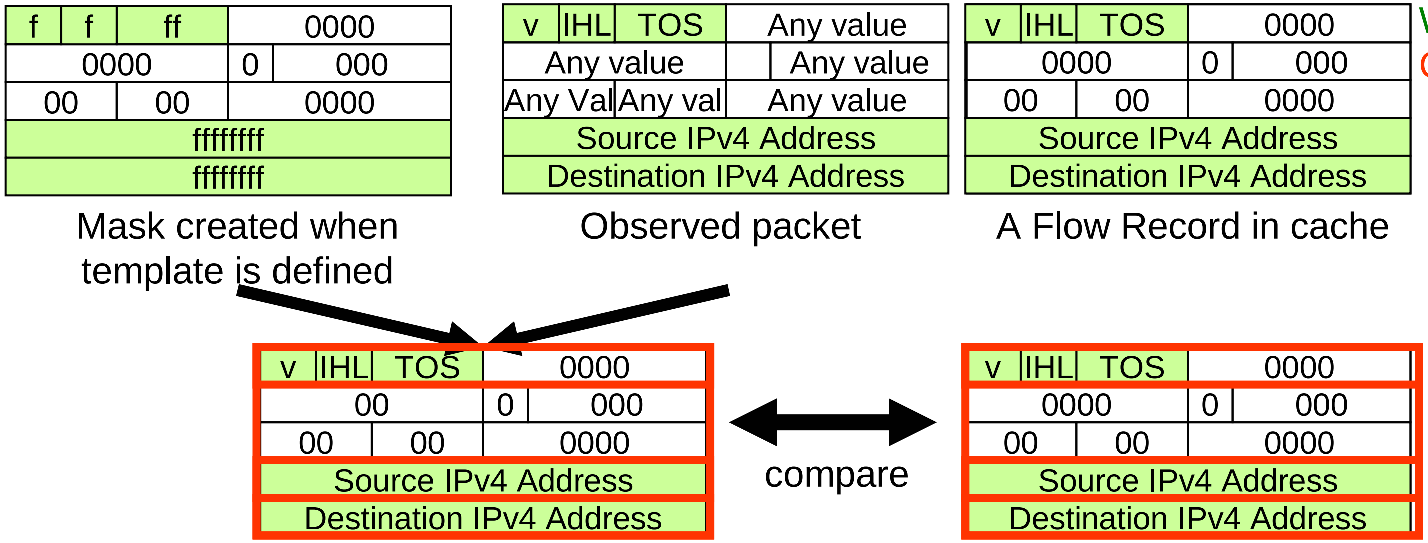
When a packet arrives:

5 comparisons

1. ip version
2. IHL
3. TOS
4. Source Address
5. Destination Address

Multiple field comparison (our approach)

Premise: Fields of Flow Records are placed in the referring order as packet header fields



f	f	ff	0000	
0000			0	000
00	00		0000	
ffffffff				
ffffffff				

Mask created when template is defined

v	IHL	TOS	Any value	
Any value				Any value
Any Val	Any val		Any value	
Source IPv4 Address				
Destination IPv4 Address				

Observed packet

v	IHL	TOS	0000	
0000			0	000
00	00		0000	
Source IPv4 Address				
Destination IPv4 Address				

A Flow Record in cache

v	IHL	TOS	0000	
00			0	000
00	00		0000	
Source IPv4 Address				
Destination IPv4 Address				

Masked observed packet

v	IHL	TOS	0000	
0000			0	000
00	00		0000	
Source IPv4 Address				
Destination IPv4 Address				

A Flow Record in cache

When Template is defined:

Create a Mask

When a packet arrives:

Mask the packet

And

compare these memory areas at the same time

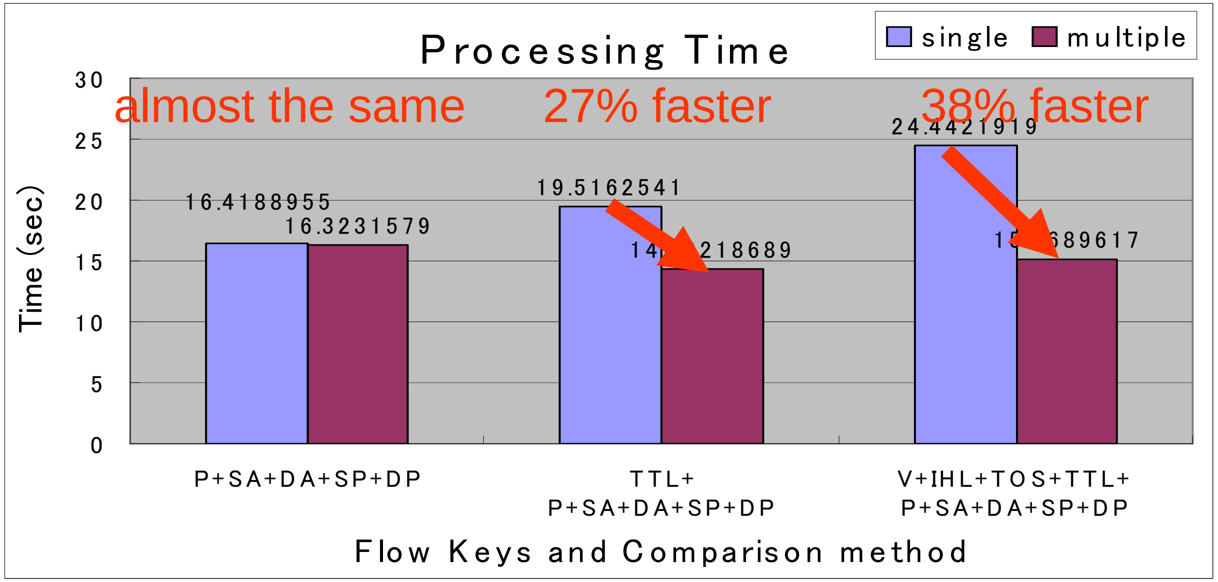
(e.g., memcmp in C language)

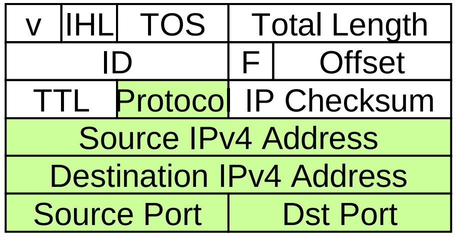
Or

1. v + IHL + TOS
2. Source Address
3. Destination Address

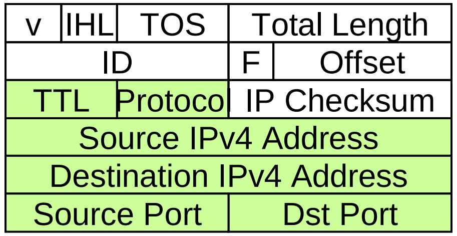
(32-bit architecture)

Evaluation of comparison method for multiple fields

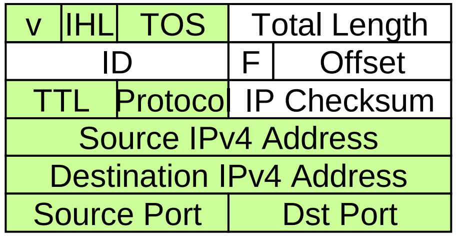



v IHL TOS Total Length
ID F Offset
TTL Protocol IP Checksum
Source IPv4 Address
Destination IPv4 Address
Source Port Dst Port

P+SA+DA+SP+DP


v IHL TOS Total Length
ID F Offset
TTL Protocol IP Checksum
Source IPv4 Address
Destination IPv4 Address
Source Port Dst Port

TTL+
P+SA+DA+SP+DP


v IHL TOS Total Length
ID F Offset
TTL Protocol IP Checksum
Source IPv4 Address
Destination IPv4 Address
Source Port Dst Port

V+IHL+TOS+TTL+
P+SA+DA+SP+DP