

# Order of Information Elements

draft-irino-ipfix-ie-order-02.txt

NTT Network Service System Laboratories, NTT Corporation

Hitoshi Irino, NTT

## Overview

- Purpose of this work
  - Optimize collectors' performance by using the same order of the IEs in exporters and collectors.
- The draft
  - Defines the order of different Element ID IEs
  - Suggests the order based on the size of IE: Three types of the IE size are used:
    1. Constant Fixed Length IEs
      - Addresses, Absolute Time Stamp, etc
    2. Reduced Size Encoding applicable IEs
      - Counters, etc
    3. Variable Length IEs
      - Strings, octet arrays

## What I have done since the last meeting

- Update from 01 draft
  - Clarification for Reduces-Sized-Encoding IEs
    - Different size multiple IEs with the same Element ID should be placed in area for Reduced-Size-Encoding-applicable IEs.
  
- Performance measurement test for collector
  - To verify the merit of determining the order

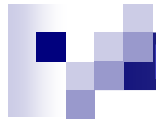
# Performance Test

- Result of the total of the 10 measurements
  - Processing in the **same order between the exporter and the collector can optimize** the performance.
  - Processing in the **suggested order is the fastest** among the three orders.
  
- If the exporters and the collectors agree to use the same order of the IEs, **the implementation optimization can be facilitated to reduce the processing time.**
  - Note: Even if they use the different order, they should work correctly.

Total Time (sec)		Exporter		
		Order same as NFv5	Suggested Order	Random Order
Collector	Order same as NFv5	5.063	5.309	7.674
	Suggested Order	5.375	4.122	7.678
	Random Order	7.258	7.225	6.154
Ratio		Exporter		
		Order same as NFv5	Suggested Order	Random Order
Collector	Order same as NFv5	1.228	1.288	1.862
	Suggested Order	1.304	1.000	1.863
	Random Order	1.761	1.753	1.493

## What's next?

- The fixed order of the IEs is effective for optimizing the collectors' performance.
  
- We should work on:
  - Optimization by a reference order of the IEs
  - Design of the reference order
    - The order in the draft is just an example.



Thank you.

# Appendix: supplementary materials of detailed results

## Works of a exporter and a collector in the test

- An exporter sends
  - data records using 3 kind of templates
    - These Templates contains same set of IEs
      1. Template using order same as NetFlow version 5
      2. Template using suggested order
      3. Template using random order
    - data records are created from 10 million packets
- The collector program repeats as follows
  1. Reading data records
  2. convert format into collector's internal unified format and buffering
    - The internal unified format using 3 kind of orders are tested
      1. Order same as NetFlow version 5
      2. Order using suggested order
      3. Order using random order
  3. Writing to file from buffering memory



# Tested ordering exporter's templates and collector's format

## An exporter program uses 3 kind of templates

1. Order same as NetFlow v5
2. Suggested Order
3. random Order

sourceIPv4Address(8)	
destinationIPv4Address(12)	
ipNextHopIPv4Address(15)	
ingressInterface(10)	egressInterface(14)
packetDeltaCount(2)	
octetDeltaCount(1)	
flowStartSysUpTime(22)	
flowEndSysUpTime(21)	
srcTransportPort(7)	dstTransportPort(11)
pad(210)	tcpCtrl(6) protocol(4) ipClass(5)
bgpSrcAsNum(16)	bgpDstAsNum(17)
srcv4PrefLen	dstv4PrefLen padding(210)

sourceIPv4Address(8)	
destinationIPv4Address(12)	
ipNextHopIPv4Address(15)	
srcTransportPort(7)	dstTransportPort(11)
srcv4PrefLen	dstv4PrefLen protocol(4) ipClass(5)
tcpCtrl(6)	padding(210)
ingressInterface(10)	egressInterface(14)
bgpSrcAsNum(16)	bgpDstAsNum(17)
flowStartSysUpTime(22)	
flowEndSysUpTime(21)	
octetDeltaCount(1)	
packetDeltaCount(2)	

sourceIPv4Address(8)	
ingressInterface(10)	destination
IPv4Address(12)	egressInterface(14)
ipNextHopIPv4Address(15)	
bgpSrcAsNum(16)	srcTransportPort(7)
bgpDstAsNum(17)	dstTransportPort(11)
flowStartSysUpTime(22)	
srcv4PrefLen	flowEndSysUp
Time(21)	dstv4PrefLen octet
DeltaCount(1)	tcpCtrl(6) packet
DeltaCount(2)	protocol(4)
Padding(210)	
ipClass(5)	

## A collector program uses 3 kind of format to store data

1. Order same as NetFlow v5
2. Suggested Order
3. random Order

sourceIPv4Address(8)	
destinationIPv4Address(12)	
ipNextHopIPv4Address(15)	
ingressInterface(10)	
egressInterface(14)	
packetDeltaCount(2)	
octetDeltaCount(1)	
flowStartSysUpTime(22)	
flowEndSysUpTime(21)	
srcTransportPort(7)	dstTransportPort(11)
protocol(4) ipClass(5)	bgpSourceAs
Number(16)	bgpDestinationAs
Number(17)	srcv4PrefLen dstv4PrefLen

sourceIPv4Address(8)	
destinationIPv4Address(12)	
ipNextHopIPv4Address(15)	
srcTransportPort(7)	dstTransportPort(11)
srcv4PrefLen	dstv4PrefLen protocol(4) ipClass(5)
ingressInterface(10)	
egressInterface(14)	
bgpSourceAsNumber(16)	
bgpDestinationAsNumber(17)	
flowStartSysUpTime(22)	
flowEndSysUpTime(21)	
octetDeltaCount(1)	
packetDeltaCount(2)	

sourceIPv4Address(8)	
ingressInterface(10)	
destinationIPv4Address(12)	
egressInterface(14)	
ipNextHopIPv4Address(15)	
bgpSourceAsNumber(16)	
srcTransportPort(7)	bgpDestinationAS
Number(17)	dstTransportPort(11)
flowStartSysUpTime(22)	
srcv4PrefLen	flowEndSysUp
Time(21)	dstv4PrefLen octet
Delta	
Count(1)	packet
Delta	
Count(1)	protocol(4) ipClass(5)

# Offset and Length Table

(Collector's format: same as NetFlow version 5 order )

	Order same as NFv5					Suggested Order					Random Order				
	C	RO	WO	RL	WL	C	RO	WO	RL	WL	C	RO	WO	RL	WL
Order same as NFv5	10	0	0	12	12	11	0	0	12	12	17	0	0	4	4
		12	12	2	4		12	44	4	4		4	12	2	4
		14	16	2	4		16	58	2	2		6	4	4	4
		16	20	4	8		18	48	2	2		10	16	2	4
		20	28	4	8		24	12	2	4		12	8	4	4
		24	36	12	12		26	16	2	4		16	50	2	4
		38	48	2	2		28	50	2	4		18	44	2	2
		40	50	2	4		30	54	2	4		20	54	2	4
		42	54	2	4		32	36	8	8		22	46	2	2
		44	58	2	2		40	28	4	8		24	36	4	4
						44	20	4	8		28	58	1	1	
											29	40	4	4	
											33	59	1	1	
											34	28	4	8	
											39	20	4	8	
											43	48	1	1	
											47	49	1	1	

C: Count for Processing

RO: Read Offsets, WO: Write Offsets, RL: Read Length, WL: Write Length

# Offset and Length Table

(Collector's format: Suggested Order)

	Order same as NFv5					Suggested Order					Random Order				
	C	RO	WO	RL	WL	C	RO	WO	RL	WL	C	RO	WO	RL	WL
Suggested Order	11	0	0	12	12	8	0	0	20	20	17	0	0	4	4
		12	20	2	4		24	20	2	4		4	20	2	4
		14	24	2	4		26	24	2	4		6	4	4	4
		16	52	4	8		28	28	2	4		10	24	2	4
		20	44	4	8		30	32	2	4		12	8	4	4
		24	36	8	8		32	36	8	8		16	28	2	4
		32	12	4	4		40	44	4	8		18	12	2	2
		38	18	2	2		44	52	4	8		20	32	2	4
		40	28	2	4							22	14	2	2
		42	32	2	4							24	36	4	4
		44	16	2	2							28	16	1	1
												29	40	4	4
												33	17	1	1
												34	44	4	8
												39	52	4	8
												43	18	1	1
												47	19	1	1

C: Count for Processing

RO: Read Offsets, WO: Write Offsets, RL: Read Length, WL: Write Length

# Offset and Length Table

(Collector's format: Random Order)

	Order same as NFv5					Suggested Order					Random Order				
	C	RO	WO	RL	WL	C	RO	WO	RL	WL	C	RO	WO	RL	WL
Random Order	16	0	0	4	4	16	0	0	4	4	13	0	0	4	4
		4	8	4	4		4	8	4	4		4	4	2	4
		8	16	4	4		8	16	4	4		6	8	4	4
		12	4	2	4		12	24	2	2		10	12	2	4
		14	12	2	4		14	30	2	2		12	16	4	4
		16	50	4	8		16	36	1	1		16	20	2	4
		20	42	4	8		17	41	1	1		18	24	2	2
		24	32	4	4		18	58	2	2		20	26	2	4
		28	37	4	4		24	4	2	4		22	30	12	12
		32	24	2	2		26	12	2	4		34	42	4	8
		34	30	2	2		28	20	2	4		39	50	4	8
		38	58	2	2		30	26	2	4		43	58	1	1
		40	20	2	4		32	32	4	4		47	59	1	1
		42	26	2	4		36	37	4	4					
		44	36	1	1		40	42	4	8					
		45	41	1	1		44	50	4	8					

C: Count for Processing

RO: Read Offsets, WO: Write Offsets, RL: Read Length, WL: Write Length

## Computing Environment for the evaluation

### ■ Software Collector Program

- runs on Intel Xeon 3.06 GHz HT Architecture
- runs on Linux (debian/gnu Linux 4.0)
- is compiled by gcc4
  - optimized option: -O3