

# NetFlow Update

**Benoit Claise**  
**Cisco Systems**



**Internet Security Workshop, Hack.lu, October 2006**

# Using NetFlow for Network Anomaly Diagnosis

- **Abstract:**

Explore, from a technical point of view, the different NetFlow features that could help in network anomaly diagnosis. Specifically the brand new Flexible NetFlow feature will be covered.

- **Focus: network measurements and traffic analysis**

Giving all the information via NetFlow for the security experts to detect anomaly

- **My goal:**

share what we do with the research community for a win-win solution

- **Not a full list of NetFlow new features**

NetFlow background + NetFlow version 9 + Security related features + Flexible NetFlow + IETF front + Capacity planning

- **Note: I don't speak about IOS version and platforms**

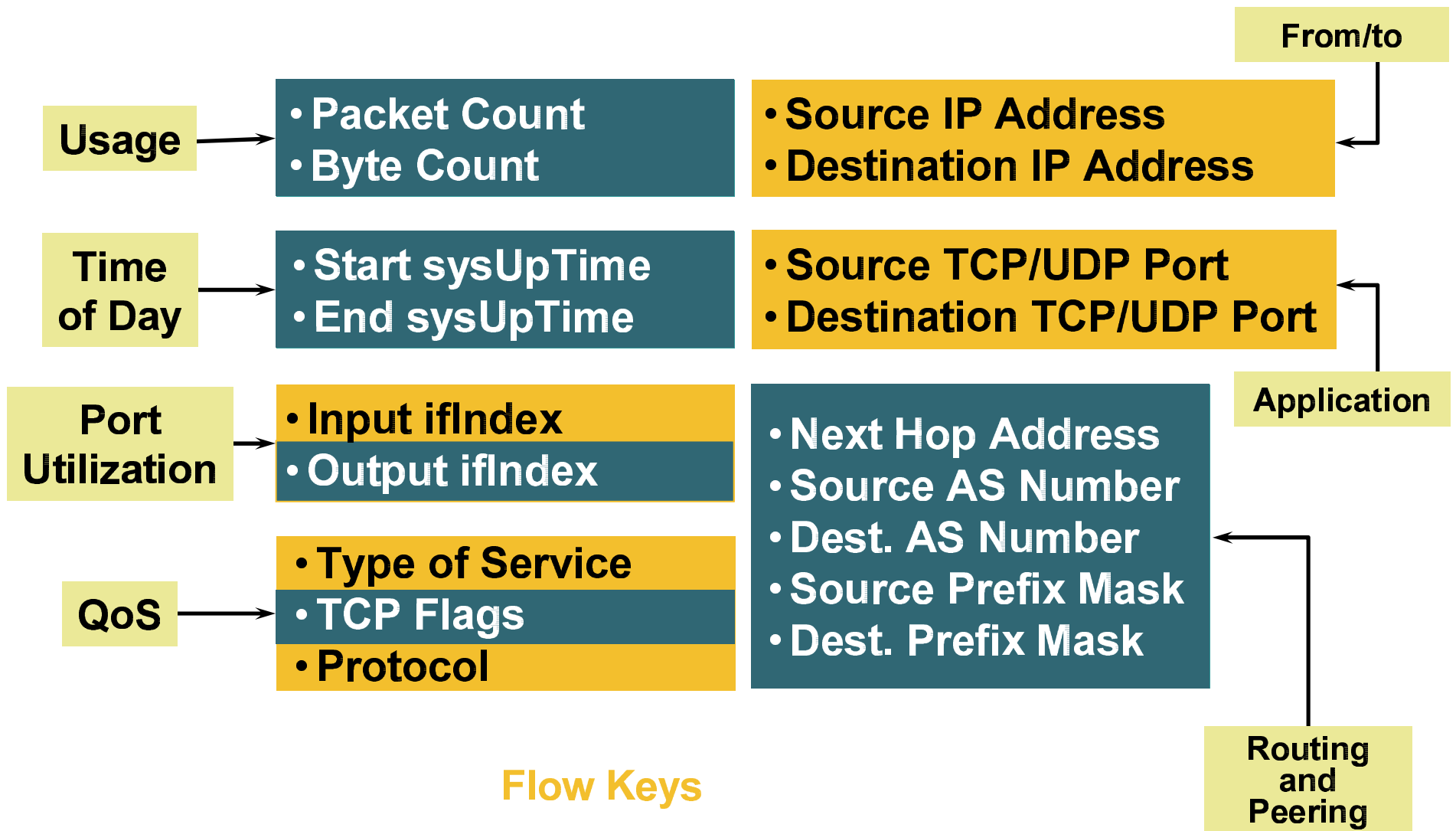
# What is NetFlow?

**NetFlow Subinterface Support**  
**Egress NetFlow Accounting**  
**MPLS Egress NetFlow**  
**NetFlow v9**  
**NetFlow**  
**Output Sampled NetFlow**  
**NetFlow Aggregation**  
**NetFlow Dynamic Top Talkers CLI**  
**NetFlow Input Filters**  
**NetFlow TOS-based Router Aggregation**  
**NetFlow Layer 2 for Security Monitoring**  
**NetFlow MIB and Top Talkers**  
**NetFlow Data Export version 8**  
**NetFlow Top Talkers CLI**

**NetFlow Support per Vlan**  
**Maximum Mask Aggregate Output NetFlow**  
**NetFlow Multicast Support**  
**NetFlow Policy Routing**  
**NetFlow Reliable Export (SCTP)**  
**NetFlow Bridged Flow Statistics**  
**NetFlow Data Export – Sampled**  
**NetFlow Export of BGP Next Hop**  
**NetFlow Multiple Export**  
**NetFlow Data Export Flowmask**  
**NetFlow for IPv6 unicast Traffic**  
**NetFlow Input Filters with multi-sampling rates**  
**NetFlow Data Export version 5**  
**Sampled NetFlow**  
**NetFlow Top Talkers**  
**NetFlow Minimum Prefix Mask for Router-based Aggregation**

**MPLS Egress NetFlow Accounting**  
**MPLS Aware NetFlow**  
**Random Sampled NetFlow**  
**NetFlow Data Export**  
**NetFlow export with BGP AS**  
**NetFlow Data Export Flowmask**

# Version 5 Flow Format



# NetFlow Cache Example

## 1. Create and update flows in NetFlow cache

SrcIface	SrcIPAdd	DstIface	DstIPAdd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	00A1	/24	180	00A1	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

## 2. Expiration

- Inactive Timer Expired (15 Sec Is Default)
- Active Timer Expired (30 Min Is Default)
- NetFlow Cache Is Full (Oldest Flows Are Expired)
- RST or FIN TCP Flag

SrcIface	SrcIPAdd	DstIface	DstIPAdd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1800	4

## 3. Aggregation

## 4. Export version

Non-aggregated flows—export **version 5 or 9**

## 5. Transport protocol

Export Packet



E.g. Protocol-Port Aggregation Scheme Becomes

Protocol	Pkts	SrcPort	DstPort	Bytes/Pkt
11	11000	00A2	00A2	1528

Aggregated Flows—Export **Version 8 or 9**

# 'show ip cache flow'

```
router# show ip cache flow
IP packet size distribution (85435 total packets):
  1-32  64  96 128 160 192 224 256 288 320 352 384 416 448 480
  .000 .000 .125 .125 .250 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

  512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
  .000 .000 .000 .000 .500 .000 .000 .000 .000 .000 .000
```

**Packet Sizes**

```
IP Flow Switching Cache, 278544 bytes
2728 active, 3368 inactive, 85310 added
463824 aging polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
last clearing of statistics never
```

**# of Active Flows**

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active (Sec) /Flow	Idle (Sec) /Flow
TCP-X	2	0.0	1	1440	0.0	0.0	9.5
TCP-other	82580	11.2	1	1440	11.2	0.0	12.0
Total:	82582	11.2			11.2	0.0	12.0

**Rates and Duration**

**Flow Details**

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP	DstP	Pkts
Et0/0	132.122.25.60	Se0/0	192.168.1.1	06	9AEE	0007	1
Et0/0	139.57.220.28	Se0/0	192.168.1.1	06	708D	0007	1
Et0/0	165.172.153.65	Se0/0	192.168.1.1	06	CB46	0007	1

# 'show ip cache verbose flow'

```
router# show ip cache verbose flow
```

```
IP packet size distribution (23597 total packets):
```

```

1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
.000 .000 .000 .000 1.00 .000 .000 .000 .000 .000 .000

```

**Flow Rate and Duration**

```
IP Flow Switching Cache, 278544 bytes
```

```
1323 active, 2773 inactive, 23533 added
```

```
151644 aged polls, 0 flow alloc
```

```
Active flows timeout in 30 minutes
```

```
Inactive flows timeout in 15 seconds
```

```
last clearing of statistics never
```

**Destination Information**

**ToS Byte and TCP Flags**

**Source Mask and AS**

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active (Sec) /Flow	Idle (Sec) /Flow
-----		3.1	1	1440	3.1	0.0	12.9
Total	22210	3.1	1	1440	3.1	0.0	12.9

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	TOS	Flags	Pkts
Port	Msk AS	Port Msk AS	NextHop	B/Pk	Active		
Et0/0	216.120.112.114	Se0/0	192.168.1.1	06	00	10	1
5FA7 /0 0		0007 /0 0	0.0.0.0			1440	0.0
Et0/0	175.182.253.65	Se0/0	192.168.1.1	06	00	10	1

# Extensibility and Flexibility Requirements Phases Approach

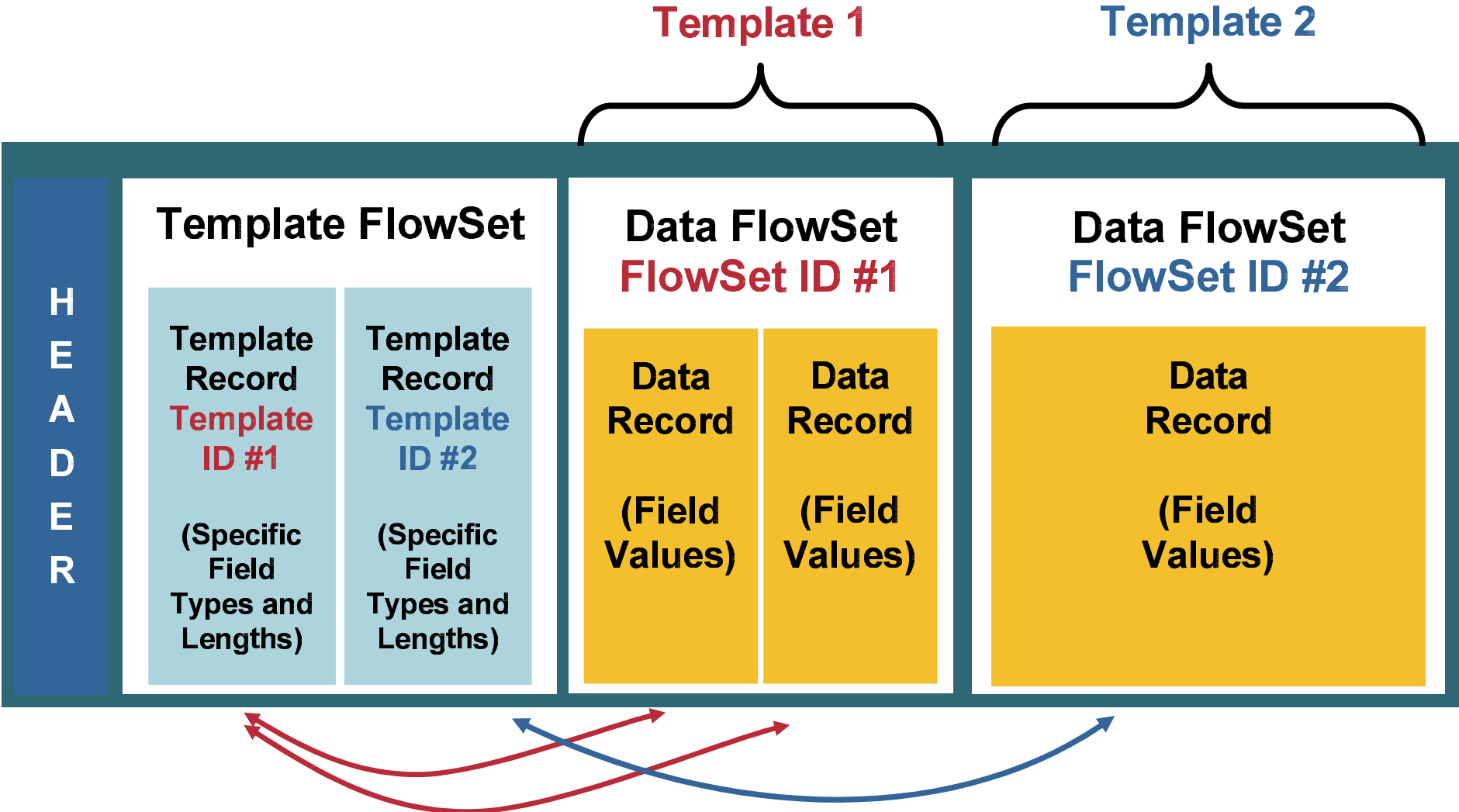
- **New requirements: build a flexible and extensible NetFlow**
  - **Phase 1: NetFlow version 9, completed**
    - Advantages: extensibility**
      - Integrate new technologies/data types quicker (MPLS, IPv6, BGP next hop, etc.)
      - Integrate new aggregations quicker
      - Limitation: template definitions are fixed
  - **Phase 2: flexible NetFlow, completed**
    - Advantages: cache and export content flexibility**
      - User selection of flow keys
      - User definition of the records
- Exporting Process**
- Metering Process**



# NetFlow Version 9

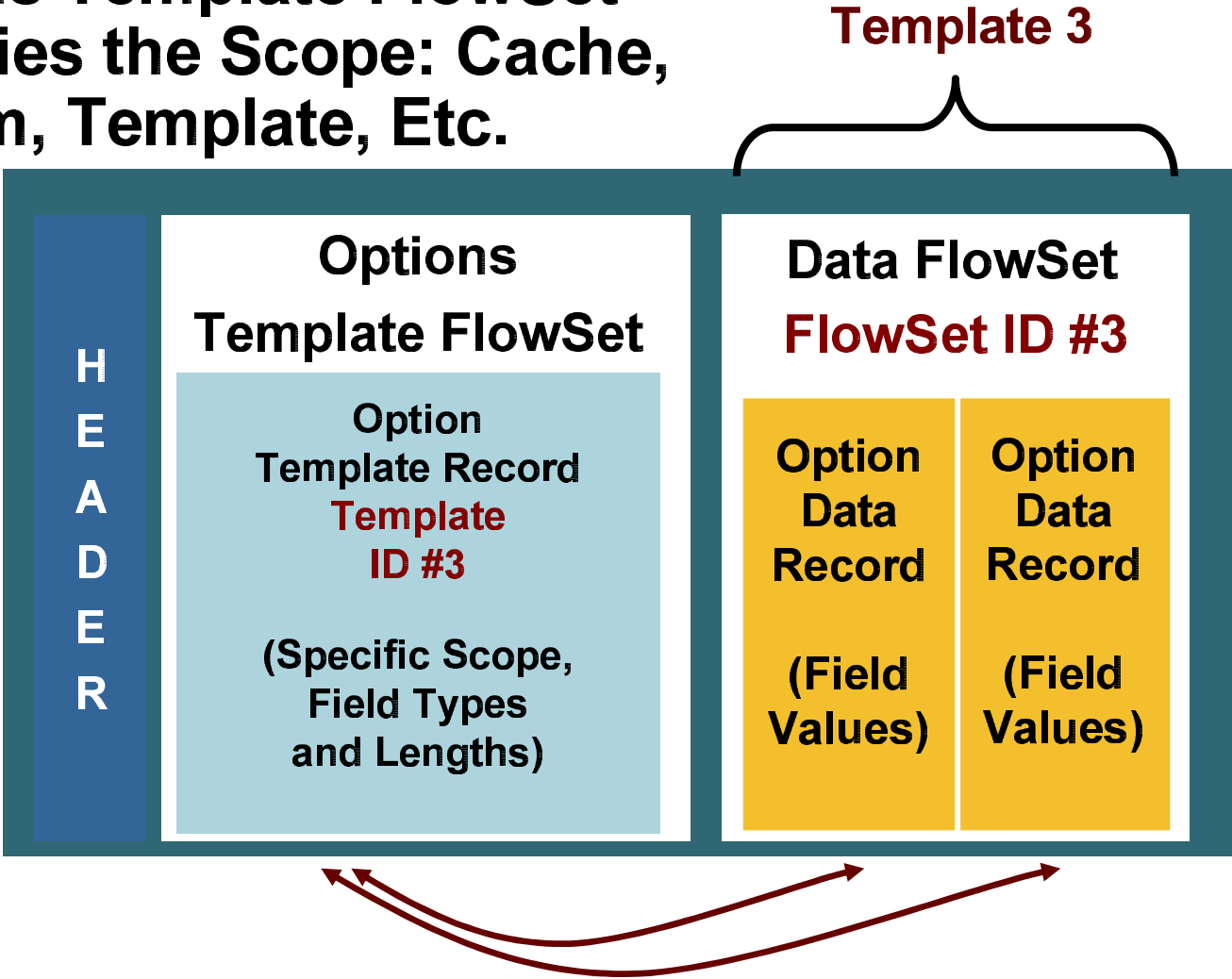
- **Version 9 is an export protocol**
  - No changes to the metering process**
- **Version 9 based on templates and separate flow records**
  - Templates composed of type and length**
  - Flow records composed of template ID and value**
  - Sent the template regularly (configurable), because of UDP**
- **RFC3954 “Cisco Systems® NetFlow Services Export Version 9”**
- **When to use NetFlow version 9?**
  - Each time a new Information Element is required**

# NetFlow Version 9 Export Packet



# NetFlow Version 9 Export Packet

**Options Template FlowSet  
Specifies the Scope: Cache,  
System, Template, Etc.**



# NetFlow L2 and Security Monitoring



# What Does a DoS Attack Look Like?

```
Router# show ip cache flow
```

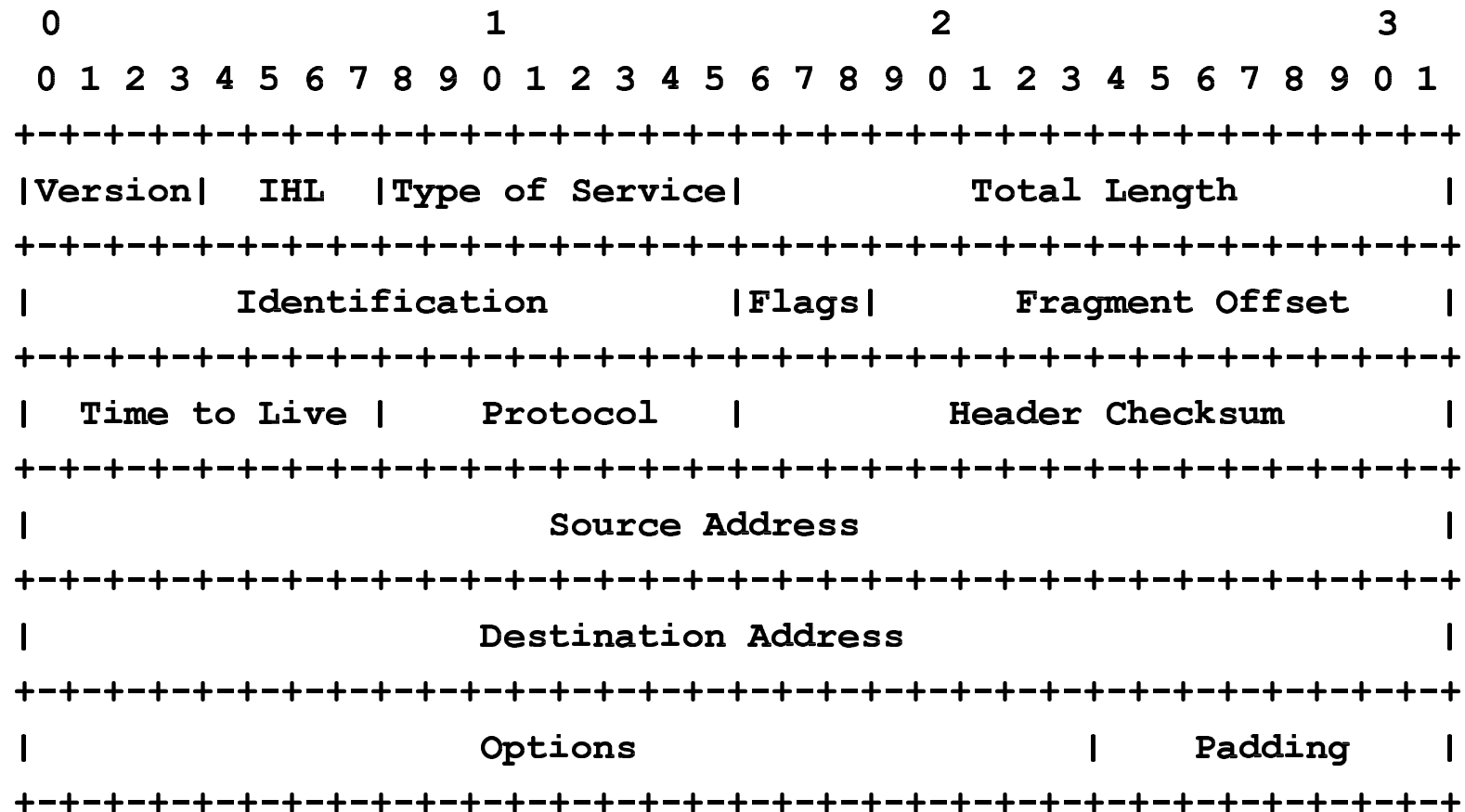
```
...
SrcIf  SrcIPAddress  SrcP  SrcAS  DstIf  DstIPAddress  DstP  DstAS  Pr  Pkts  B/Pk
29     192.1.6.69    77    aaa    49     194.20.2.2    1308  bbb    6   1     40
29     192.1.6.222  1243  aaa    49     194.20.2.2    1774  bbb    6   1     40
29     192.1.6.108  1076  aaa    49     194.20.2.2    1869  bbb    6   1     40
29     192.1.6.159  903   aaa    49     194.20.2.2    1050  bbb    6   1     40
29     192.1.6.54   730   aaa    49     194.20.2.2    2018  bbb    6   1     40
29     192.1.6.136  559   aaa    49     194.20.2.2    1821  bbb    6   1     40
29     192.1.6.216  383   aaa    49     194.20.2.2    1516  bbb    6   1     40
29     192.1.6.111  45    aaa    49     194.20.2.2    1894  bbb    6   1     40
29     192.1.6.29   1209  aaa    49     194.20.2.2    1600  bbb    6   1     40
```

- **Typical DoS attacks have the same (or similar) entries:**  
Input interface, destination IP, 1 packet per flow, constant bytes per packet (B/Pk)
- **Don't forget "show ip cache verbose flow | include ..."**
- **Export to a security oriented collector: CS-MARS, arbor**

# NetFlow L2 and Security Monitoring

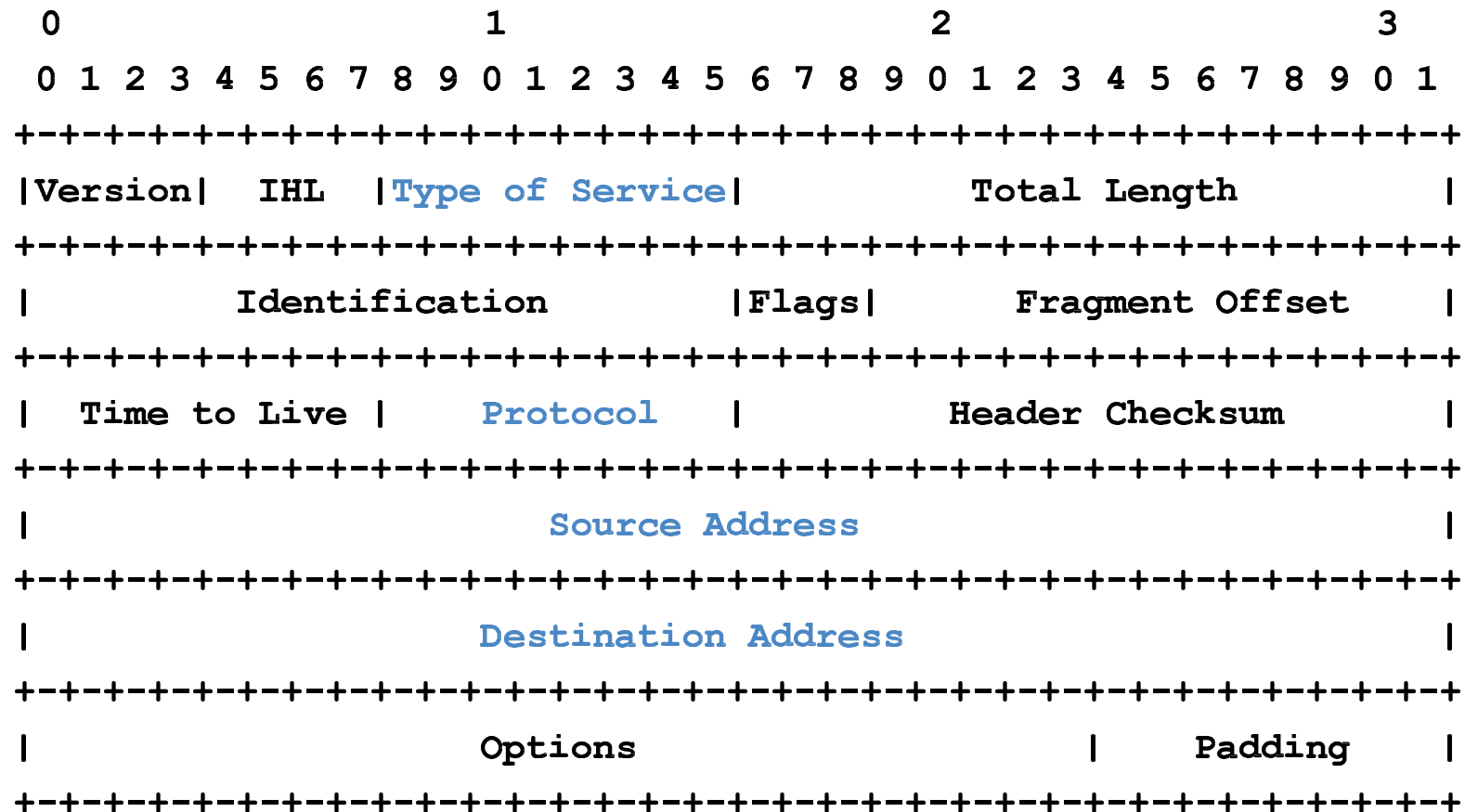
- **Layer 2 IP header fields**
  - **Source MAC address** field from frames that are received by the NetFlow router
  - **Destination MAC address** field from frames that are transmitted by the NetFlow router
  - **Received VLAN ID** field (802.1q and Cisco's ISL)
  - **Transmitted VLAN ID** field (802.1q and Cisco's ISL)
- **Extra layer 3 IP header fields**
  - **Time-to-live field**
  - **Identification field**
  - **Packet length field**
  - **ICMP type and code**
  - **Fragment offset**
- **Targeted for security: to help identify network attacks and their origin**
- **For IPv4 and IPv6**
- **Require NetFlow version 9**
- **Introduced in 12.3(14)T on the low-end router (offset in 12.4(2)T)**

# NetFlow L2 and Security Monitoring L3 Packet Format



# NetFlow L2 and Security Monitoring

## Current NetFlow L3 Fields

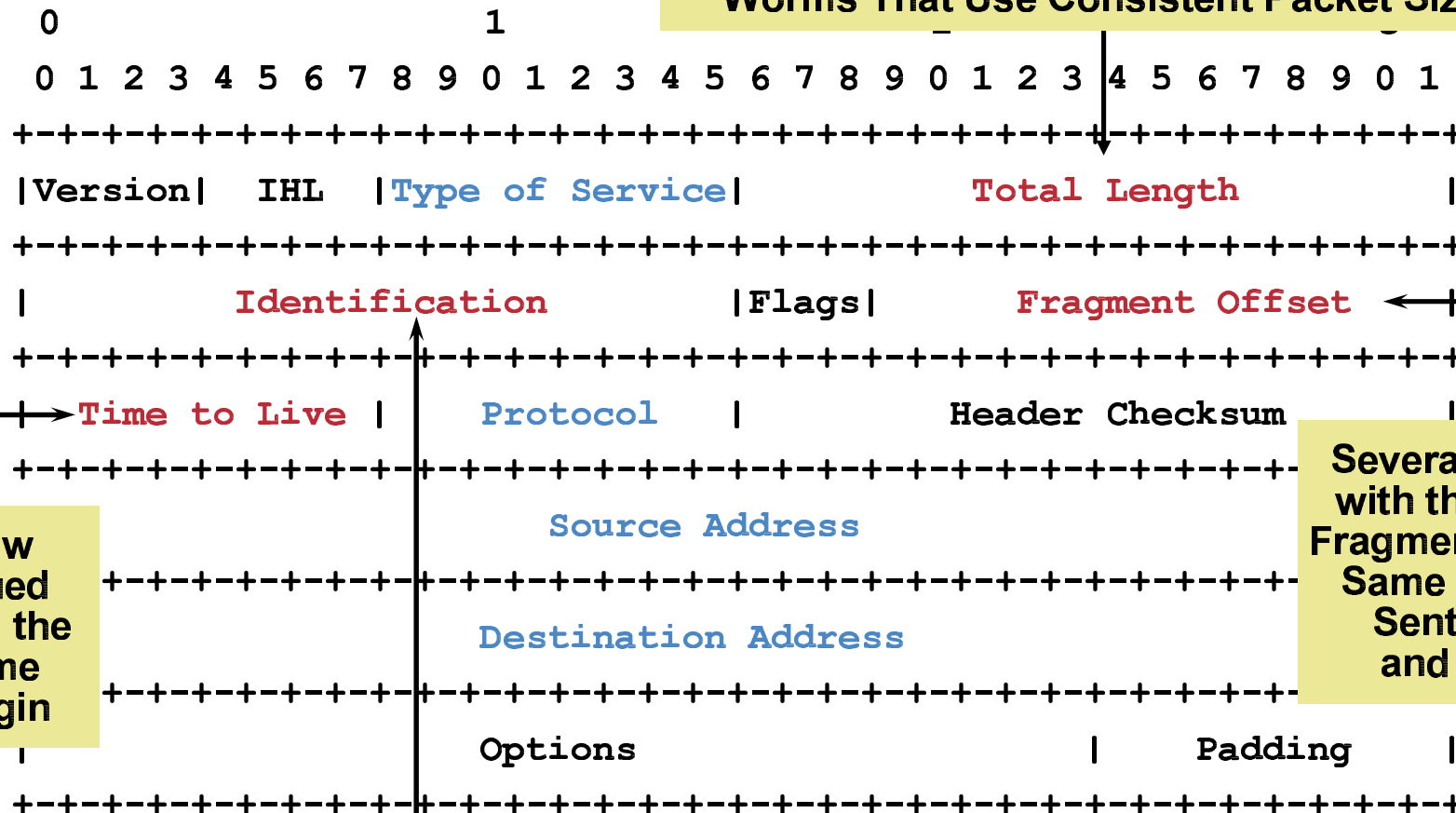




# NetFlow L2 and Security Monitoring

## Extra NetFlow L3 Fields

Attacks That Use Consistent Packet Size or Worms That Use Consistent Packet Size



Flow Issued From the Same Origin

Several Flows with the Same Fragment Offset: Same Packet Sent over and over

Very Large Packets or Attacks That Might Always Have the Same Generated Identification

# NetFlow L2 and Security Monitoring

```
Router(config)# ip flow-capture icmp
```

```
Router(config)# ip flow-capture ip-id
```

```
Router(config)# ip flow-capture mac-addresses
```

```
Router(config)# ip flow-capture packet-length
```

```
Router(config)# ip flow-capture ttl
```

```
Router(config)# ip flow-capture vlan-id
```

```
Router(config)# ip flow-capture fragment-offset
```

- **Not flow keys, the value of the first packet of the flow**
  - Exception for packet length: min/max
  - Exception for the TTL: min/max
  - Fragment-offset: the first fragmented packet
- **Complete the main cache, not the aggregation caches**
  - Info lost if an aggregation cache is used
- **Currently not available with the MIB**

# NetFlow L2 and Security Monitoring

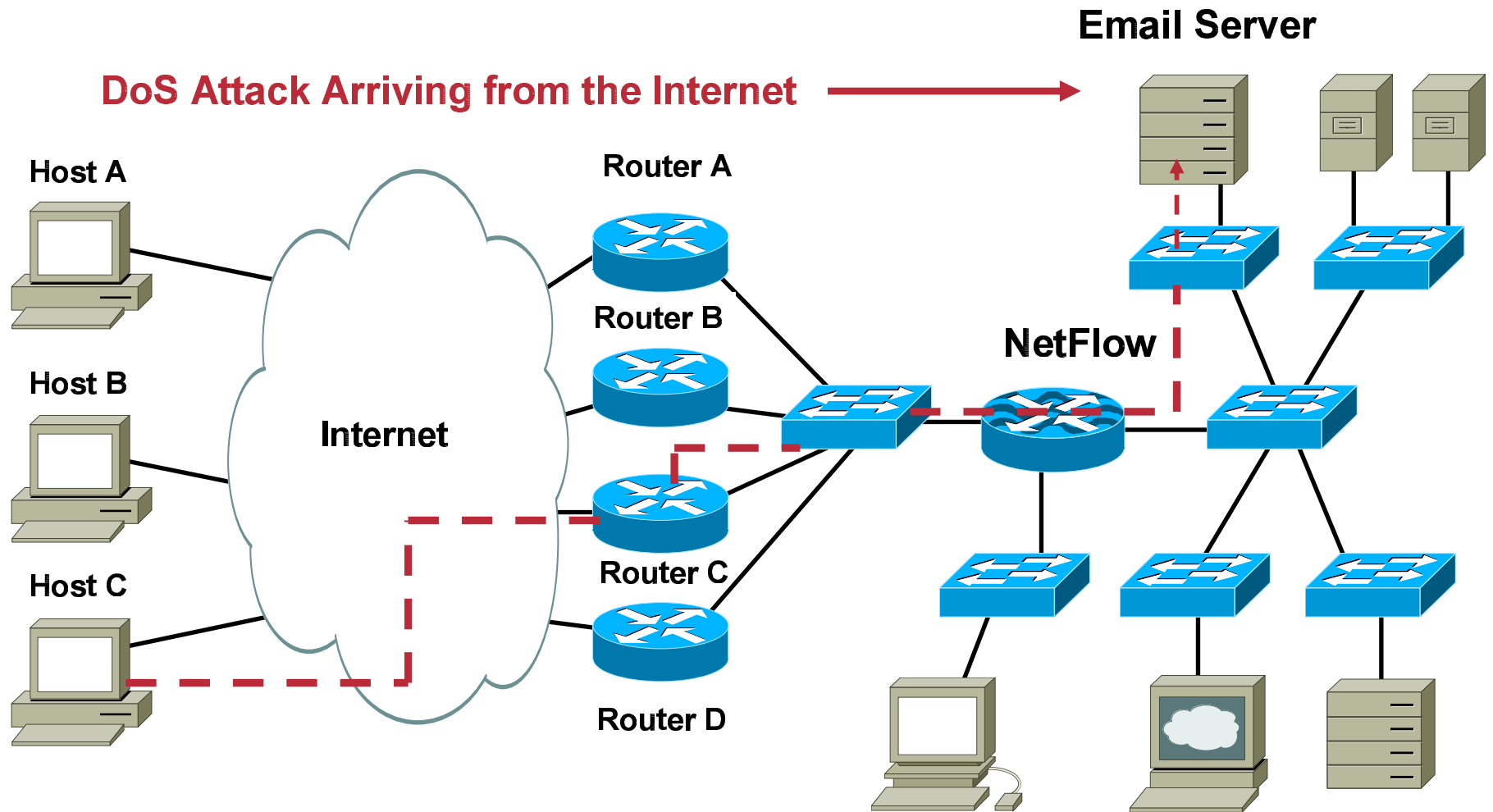
```
Router# show ip cache verbose flow
```

```
...
```

SrcIf Port Msk AS	SrcIPAddress	DstIf Port Msk AS	DstIPAddress NextHop	Pr TOS Flgs B/Pk	Pkts Active
Et0/0.1 0015 /0 0	10.251.138.218	Et1/0.1 0015 /0 0	172.16.10.2 0.0.0.0	06 80 00 840	65 10.8
MAC: (VLAN id)	aaaa.bbbb.cc03	(005)	aaaa.bbbb.cc06	(006)	
Min plen:	840		Max plen:	840	
Min TTL:	59		Max TTL:	59	
IP id:	0				

One Flow Entry

# NetFlow L2 and Security Monitoring Source MAC Address



**Report the MAC Address for Ethernet, FastEthernet, and Gigethernet**

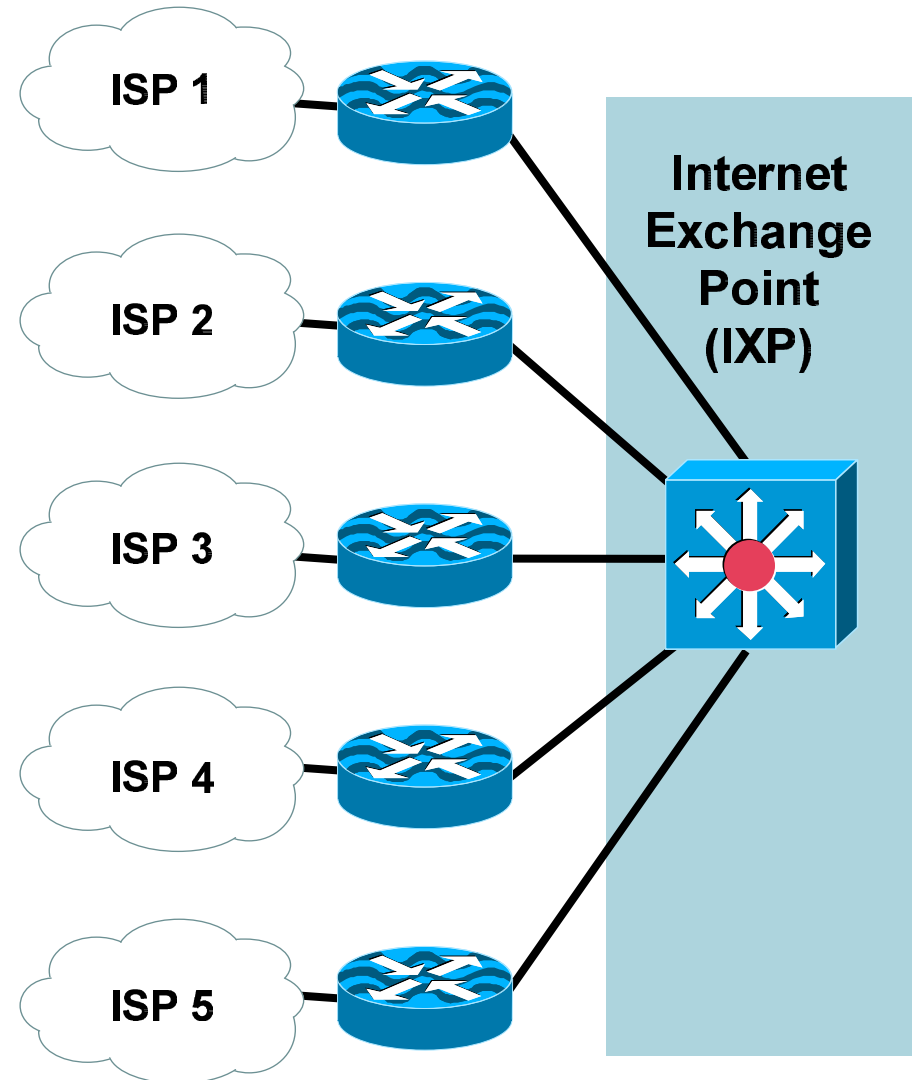
# NetFlow L2 and Security Monitoring Internet eXchange Point

- Internet exchange points require the accounting per MAC address:

Incoming

Outgoing

- NetFlow solution is more granular than the “IP accounting MAC address” feature



# NetFlow Top Talkers



# NetFlow Top Talkers

- **The flows that are generating the heaviest traffic in the cache are known as the "top talkers"**
- **Allows flows to be sorted by either of the following criteria:**
  - By the total number of packets in each top talker
  - By the total number of bytes in each top talker
- **Match criteria for the top talkers, work like a filter**
- **The top talkers can be retrieved via the CISCO-NETFLOW-MIB (cnfTopFlowsTable)**
- **A new separate cache**
  - Similar output of the show ip cache flow or show ip cache verbose flow command
  - Generated on the fly
  - Frozen for the "cache-timeout" value
- **Introduced in 12.2(25)S and 12.3(11)T on the low-end routers**

# NetFlow Top Talkers Configuration

```

Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by <packets | bytes>
Router(config-flow-top-talkers)# cache-timeout 2000
    
```

```
Router# show ip flow top-talkers verbose
```

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	TOS	Flgs	Pkts
Port Msk AS		Port Msk AS	NextHop			B/Pk	Active
IPM: OPkts	OBytes						
{ Fa1/0	10.48.71.9	Local	10.48.71.9	01	C0	10	56
		0000 /24 0	0303 /24 0			56	171.0
		ICMP type: 3	ICMP code:		3		
{ Se0/0	192.1.1.97	Se0/3	192.1.1.110	01	00	00	12
		0000 /30 0	0000 /30 0			1436	2.8
		ICMP type: 0	ICMP code:		0		



# NetFlow Top Talkers Example 2


```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by packets
Router(config-flow-top-talkers)# cache-timeout 2000
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match destination address 192.1.1.110/32
```

```
Router# show ip flow top-talkers verbose
```

SrcIf Port Msk AS	SrcIPAddress	DstIf Port Msk AS	DstIPAddress NextHop	Pr	TOS	Flgs	Pkts B/Pk Active
Se0/0 0000 /30 0	192.1.1.97	Se0/3 0000 /30 0	192.1.1.110 192.1.1.108	01	00	00	12 1436 2.8
ICMP type:	0		ICMP code:	0			

# NetFlow Top Talkers Example 2

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by packets
Router(config-flow-top-talkers)# cache-timeout 2000
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match destination address 192.1.1.110/32
```



```
match [[source address | destination address | nexthop address]
[ip-address] [mask | /nn]] [[source port | destination port] [port-number |
min port | max port | min port max port]] [[source as | destination as]
as-number] [[input-interface | output-interface] interface] [tos
[tos-value | dscp dscp-value | precedence precedence-value]]
[protocol [protocol-number | tcp | udp]] [flow-sampler flow-sampler-name]
[class-map class] [packet-range | byte-range [[min-range-number
max-range-number] [min minimum-range | max maximum-range |
min minimum-range max maximum-range]]]
```

# Egress NetFlow and Top Talkers

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match direction ?
    egress  Match egress flows
    ingress Match ingress flows
```

- **The direction match statement added**
- **The “direction” is a new information element**
  - Egress value added in the template**
  - Egress value not added for the aggregation caches**
  - Existing ingress templates are not modified**

# NetFlow Dynamic Top Talkers



# NetFlow Dynamic Top Talkers

- **Somehow similar to the top talkers**
  - But dynamic, done on the fly with show commands
  - But does not require modifications to the router config
  - But does not create a new cache
  - But no available with the MIB—obviously
- **Even more useful than top talkers for security**
- **“show ip flow top” command:**
  - `show ip flow top <N> <aggregate-field> <sort-criteria>  
<match-criteria>`
- **Introduced in 12.4(4)T on the low-end routers**

# NetFlow Dynamic Top Talkers Examples

- Top ten protocols currently flowing through the router:

```
Router# show ip flow top 10 aggregate protocol
```

- Top ten IP addresses which are sending the most packets

```
Router# show ip flow top 10 aggregate source-address sorted-by packets
```

- Top five destination addresses to which we're routing most traffic from the 10.10.10.0/24 prefix

```
Router# show ip flow top 5 aggregate destination-address match source-prefix 10.10.10.0/24
```

- 50 VLAN's that we're sending the least bytes to:

```
Router# show ip flow top 50 aggregate destination-vlan sorted-by bytes ascending
```

- Top 20 sources of 1-packet flows:

```
router# show ip flow top 50 aggregate source-address match packets 1
```

# NetFlow MIB

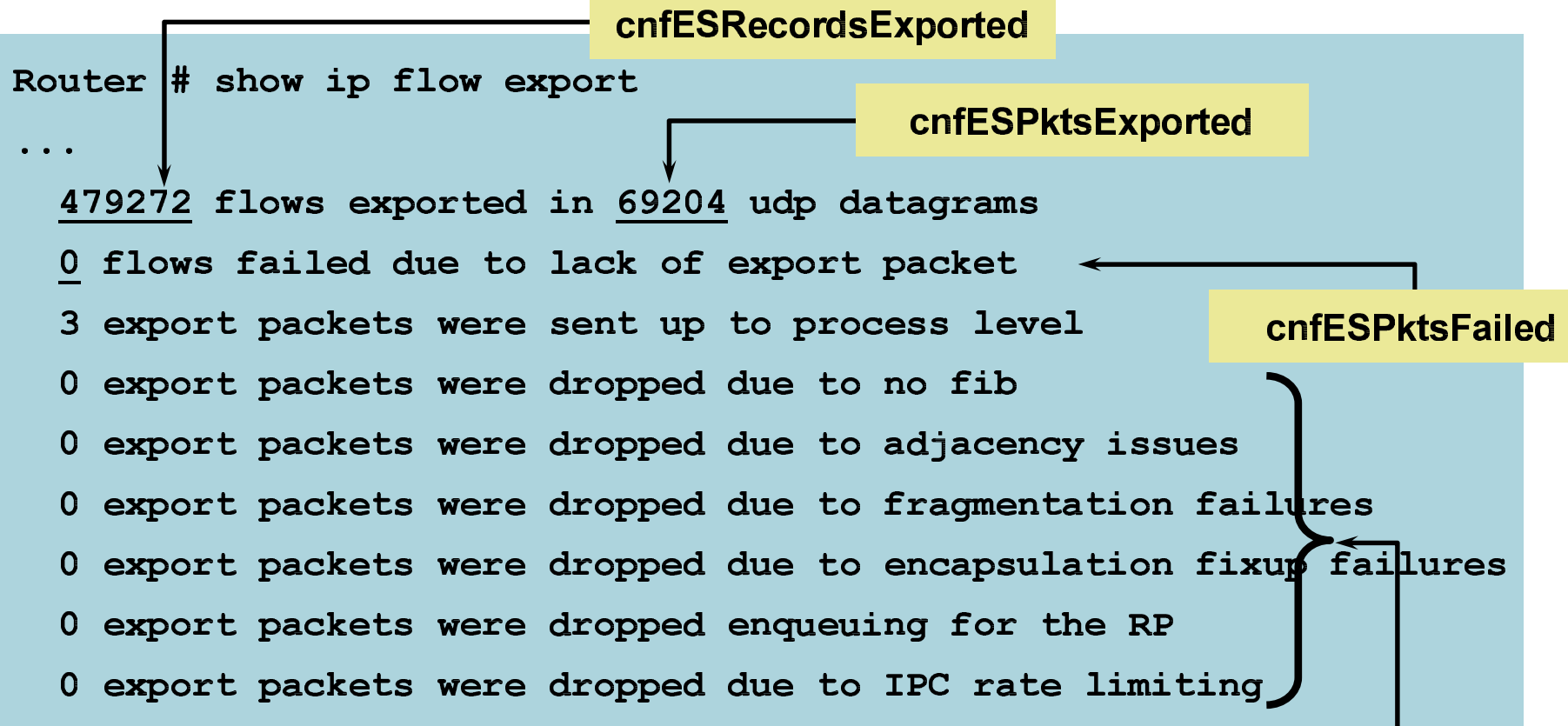


# CISCO-NETFLOW-MIB

- **Managed objects to **configure** the following NetFlow information:**
  - Flow cache, interface, export, peer-as versus origin-as
  - Exception: no sampled NetFlow configuration
- **Managed objects to **monitor** the following NetFlow information:**
  - Packet size distribution, number of bytes exported per second, number of flows/UDP datagrams exported, number of template active, export statistics, protocol statistics, etc.
- **Monitor the **top flows****
- **The CISCO-NETFLOW-MIB.my is **not**:**
  - A replacement for the traditional method of exporting a flow cache
- **Note that CISCO-SWITCH-ENGINE-MIB, on the Cisco Catalyst, allows to query the multilayer switching flow records**
- **Introduced in 12.2(25)S and 12.3(7)T**
- **Don't forget the threshold mechanism with the RMON event/alarm or the EVENT-MIB**



# NetFlow MIB Monitoring



- **The export rate ratecnfESEExportRate**

**Useful to estimate the required bandwidth**

# NetFlow MIB Monitoring

Router# show ip cache flow

IP packet size distribution (311656 total packets): ← **cnfPSPacketSizeDistribution**

```

1-32  64  96  128  160  192  224  256  288  320  352  384  416  448  480
.356 .316 .144 .115 .004 .003 .000 .007 .001 .000 .002 .017 .018 .009 .000

512  544  576  1024  1536  2048  2560  3072  3584  4096  4608
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
  
```

**cnfPSProtocolStatTable**

...

Protocol	Total	Flows	Packets	Bytes	Packets	Active(Sec)	Idle(Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-Telnet	33	0.0	65	40	0.0	18.4	10.0
TCP-WWW	3	0.0	5	45	0.0	3.0	1.2
TCP-BGP	5343	0.0	2	47	0.0	5.1	11.1
TCP-other	411	0.0	2	48	0.0	1.0	10.9
UDP-other	98614	0.4	2	76	0.9	2.1	10.8
ICMP	9519	0.0	9	71	0.4	21.3	11.5
<b>Total:</b>	<b>113923</b>	<b>0.5</b>	<b>2</b>	<b>73</b>	<b>1.4</b>	<b>3.8</b>	<b>10.9</b>

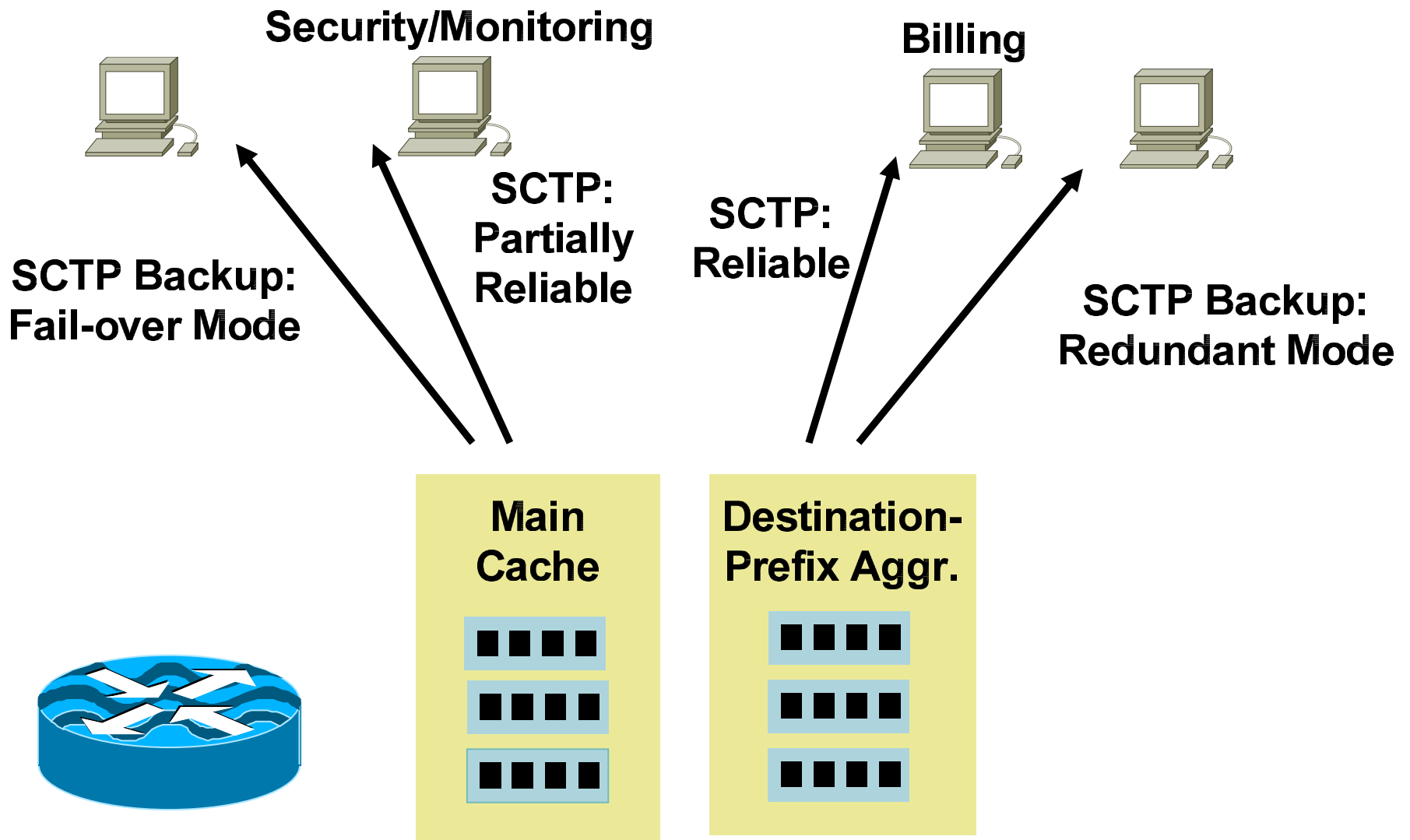
# NetFlow over SCTP



# NetFlow Reliable Export with SCTP

- **SCTP-PR support for NetFlow version 5, 8, 9**
- **(Options) templates sent reliably**
- **Two primary SCTP export destinations (collectors) and two backup SCTP export destinations**
  - For each cache: either main cache or aggregation cache(s)
- **Backup**
  - Fail-over mode: open the backup connection when the primary fails
  - Redundant mode: open the backup connection in advance, and already send the templates
  - Note that the backup inherits the reliability level from the primary
- **12.4(4)T for the low-end routers**
- **NetFlow collector SCTP support in version 6.0**

# NetFlow Reliable Export with SCTP Example



# Reliable Export with SCTP Example

```
Router(config)# ip flow-export destination 10.10.10.10 9999 sctp
Router(config-flow-export-sctp)# reliability partial buffer-limit 100
Router(config-flow-export-sctp)# backup destination 11.11.11.11 9999
Router(config-flow-export-sctp)# backup fail-over 1000
Router(config-flow-export-sctp)# backup mode fail-over
```

```
Router(config)# ip flow-aggregation cache destination-prefix
Router(config-flow-cache)# export destination 12.12.12.12 9999 sctp
Router(config-flow-cache)# backup destination 13.13.13.13 9999
Router(config-flow-cache)# backup mode fail-over
Router(config-flow-cache)# enabled
```

# Reliable Export with SCTP

```
Router# show ip flow export sctp verbose
IPv4 main cache exporting to 10.10.10.10, port 9999, partial
status: fail-over
backup mode: fail-over
0 flows exported in 0 sctp messages.
0 packets dropped due to lack of SCTP resources
fail-over time: 1000 milli-seconds
restore time: 25 seconds
backup: 11.11.11.11, port 9999
    status: initialising
    fail-overs: 0
    0 flows exported in 0 sctp messages.
    0 packets dropped due to lack of SCTP resources
destination-prefix cache exporting to 12.12.12.12, port 9999, full
status: fail-over
backup mode: fail-over
0 flows exported in 0 sctp messages.
0 packets dropped due to lack of SCTP resources
fail-over time: 25 milli-seconds
restore time: 1 seconds
backup: 13.13.13.13, port 9999
    status: initialising
    fail-overs: 920
    0 flows exported in 0 sctp messages.
    0 packets dropped due to lack of SCTP resources
```

# Flexible NetFlow





# NetFlow Flow Keys on the Router

- **By default, the flow keys are:**  
Source IP address, destination IP address, source port, destination port, layer 3 protocol type, TOS byte (DSCP), input interface
- **The 12 NetFlow aggregation allows to reduce/change the number of flow keys**  
Example: source prefix aggregation = source network, source interface  
Can be seen as a different view of the main cache
- **Egress NetFlow, MPLS aware NetFlow, etc.**  
Will specify new flow keys
- **Note: on the Cisco Catalyst<sup>®</sup>, we speak of the flow mask**  
Define the flow keys

# Flexible NetFlow

## High Level Concepts and Advantages

- **Flexible NetFlow feature allows user configurable NetFlow record formats, selecting from a collection of fields:**

- Key**

- Non-key**

- Counter**

- Timestamp**

- **Advantages:**

- Tailor a cache for specific applications, not covered by existing 21 NetFlow features**

- Better scalability since flow record customization for particular application reduces number of flows to monitor**

- Different NetFlow configuration:**

- Per subinterface**

- Per direction (ingress/egress)**

- Per sampler**

- Etc.**

# Flow Key and Non-Key Fields

- **Choice of fields includes IPv4 header, transport (TCP, UDP), routing, flow (direction, sampler), interface**
- **Non-key fields are not used to define a flow and are exported along with the flow and provide additional information**

**Traditional IP NF non-key fields:**

**Source and destination AS's**

**Source and destination IP prefix masks**

**IP address of next hop router**

**TCP flags**

**Output interface**

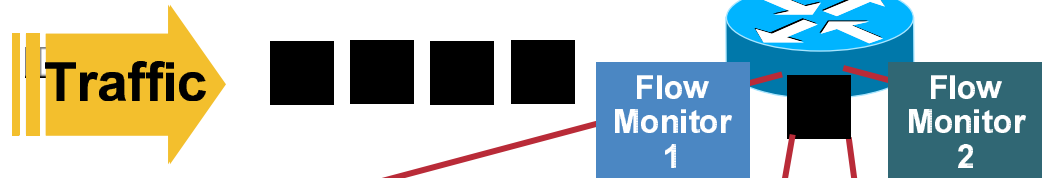
**Note: given by the value of the first packet of the flow**

**NF features provide per flow statistics:**

**Number of packets and bytes in flow**

**Timestamps for first and last packets in flow**

# Flexible NetFlow Multiple Monitors with Unique Key Fields



Key Fields	Packet 1
Source IP	3.3.3.3
Destination IP	2.2.2.2
Source Port	23
Destination Port	22078
Layer 3 Protocol	TCP - 6
TOS Byte	0
Input Interface	Ethernet 0

Non-Key Fields
Packets
Bytes
Timestamps
Next Hop Address

Key Fields	Packet 1
Source IP	3.3.3.3
Dest IP	2.2.2.2
Input Interface	Ethernet 0
SYN Flag	0

Non-Key Fields
Packets
Timestamps

**Traffic Analysis Cache**

Source IP	Dest. IP	Source Port	Dest. Port	Protocol	TOS	Input I/F	...	Pkts
3.3.3.3	2.2.2.2	23	22078	6	0	E0	...	1100

**Security Analysis Cache**

Source IP	Dest. IP	Input I/F	Flag	...	Pkts
3.3.3.3	2.2.2.2	E0	0	...	11000

# Flexible NetFlow Configuration

## Configure the Exporter

- Where do I want my data sent?

## Configure the Flow Record

- What data do I want to meter?

## Configure the Flow Monitor

- Creates a new NetFlow cache
- Attach the flow record
- Exporter is attached to the cache
- Potential sampling configuration

## Configure the Interface

- Configure NetFlow on the interface

# Flexible NetFlow Components

- **Flow record—defines what is captured by NetFlow**
  - Two kinds of flow records: predefined or user-defined
  - Include key and non-key fields
- **Flow exporter—where NetFlow will be exported**
  - Multiple flow exporters per flow monitor
- **Flow monitor—a flow cache containing flow records**
  - Cache creation for a specific flow record
  - Applied to an interface
  - Bound to one or more flow exporter(s)
  - Packet sampling possible per flow monitor

# Configure a User-Defined Flow Record

## Configure the Exporter

```
Router(config)#flow exporter my-exporter  
Router(config-flow-exporter)#destination 1.1.1.1
```

## Configure the Flow Record

```
Router(config)#flow record my-record  
Router(config-flow-record)#match ipv4 icmp type  
Router(config-flow-record)#match ipv4 icmp code  
Router(config-flow-record)#collect counter bytes
```

## Configure the Flow Monitor

```
Router(config)#flow monitor my-monitor  
Router(config-flow-monitor)#exporter my-exporter  
Router(config-flow-monitor)#record my-record
```

## Configure the Interface

```
Router(config)#int s3/0  
Router(config-if)#ip flow monitor my-monitor input
```

# Flexible NetFlow

## User Defined Record Configuration

```
Router(config)# flow record my-record  
Router(config-flow-record)# match      -> Specify a key field  
Router(config-flow-record)# collect    -> Specify a non-key field
```

```
Router(config-flow-record)# match ?  
  flow          Flow identifying fields  
  interface     Interface fields  
  ipv4          IPv4 fields  
  routing       routing attributes  
  transport     Transport layer field
```

```
Router(config-flow-record)# collect ?  
  counter       Counter fields  
  flow          Flow identifying fields  
  interface     Interface fields  
  ipv4          IPv4 fields  
  routing       IPv4 routing attributes  
  timestamp     Timestamp fields  
  transport     Transport layer fields
```



# Flexible Flow Record Key Fields for Security

IPv4		Routing		Transport	
IP (Source or Destination)	Payload Size	Destination AS	Peer AS	Destination Port	TCP Flag: ACK
Prefix (Source or Destination)	Packet Section (Header)	Traffic Index	Forwarding Status	Source Port	TCP Flag: CWR
Mask (Source or Destination)	Packet Section (Payload)	Is-Multicast	IGP Next Hop	ICMP Code	TCP Flag: ECE
Minimum-Mask (Source or Destination)	TTL	BGP Next Hop		ICMP Type	TCP Flag: FIN
Protocol	Options			IGMP Type	TCP Flag: PSH
Fragmentation Flags	Version			TCP ACK Number	TCP Flag: RST
Fragmentation Offset	Precedence			TCP Header Length	TCP Flag: SYN
ID	DSCP			TCP Sequence Number	TCP Flag: URG
Header Length	TOS			TCP Window-Size	UDP Message Length
Total Length				TCP Source Port	UDP Source Port
				TCP Destination Port	UDP Destination Port
				TCP Urgent Pointer	

# Flexible Flow Record

## Non-Key Fields for Security

- Any of the potential “key” field: will be the value of the first packet in the flow
- Plus

Counters	Timestamp	IPv4
Bytes	sysUpTime First Packet	Total Length Minimum
Bytes Long	sysUpTime First Packet	Total Length Maximum
Bytes Square Sum		TTL Minimum
Packet		TTL Maximum
Packet Long		

# Packet Section Fields

- **Contiguous chunk of a packet of a user configurable size, used as a key or a non-key field**
- **Sections used for detailed traffic monitoring, DDoS attack investigation, worm detection, other security applications**
- **Chunk defined as flow key, should be used in sampled mode with immediate aging cache**
- **Starts at the beginning of the IPv4 header**

```
collect or match ipv4 header <size in bytes>
```

- **Immediately follows the IPv4 header**

```
collect or match ipv4 payload <size in bytes>
```

# Flexible NetFlow Activation on Interface

**Send the “sampler-table”  
Option**

```
Router(config-if)# ip flow monitor <monitor-name>  
                    [sampler <sampler-name>]  
                    [input | output]
```

**For the Input or Output Traffic.  
Does Not Determine the Flow Key**

- **Deterministic or random is available**

```
Router(config)# sampler <sampler-name>  
mode [deterministic | random] <value N> out-of <value M>
```

# Flexible Monitor Configuration

**Potentially Multiple**

**3 Types of Cache:  
See Next Slides**

```
flow monitor <monitor-name>  
  record <record-name>  
  exporter <exporter-name>  
  cache type {normal | immediate | permanent}  
  cache entries <number-of-entries>  
  cache timeout {active | inactive | update} <value-in-sec>  
  statistics packet protocol  
  statistics packet size
```

**Collect Size  
Distribution Statistics**

**Collect Protocol  
Distribution Statistics**

# Three Types of NetFlow Caches

- **Normal cache**

  - Similar to today's NetFlow

  - More flexible active and inactive timers: one second minimum

- **Immediate cache**

  - Flow accounts for a single packet

  - Desirable for real-time traffic monitoring, DDoS detection, logging

  - Desirable when only very small flows are expected (ex: sampling)

  - Caution: may result in a large amount of export data

- **Permanent cache**

  - To track a set of flows without expiring the flows from the cache

  - Entire cache is periodically exported (update timer)

  - After the cache is full (size configurable), new flows will not be monitored

  - Uses update counters rather than delta counters

# Core Traffic Matrix with Flexible NetFlow Configuration Example

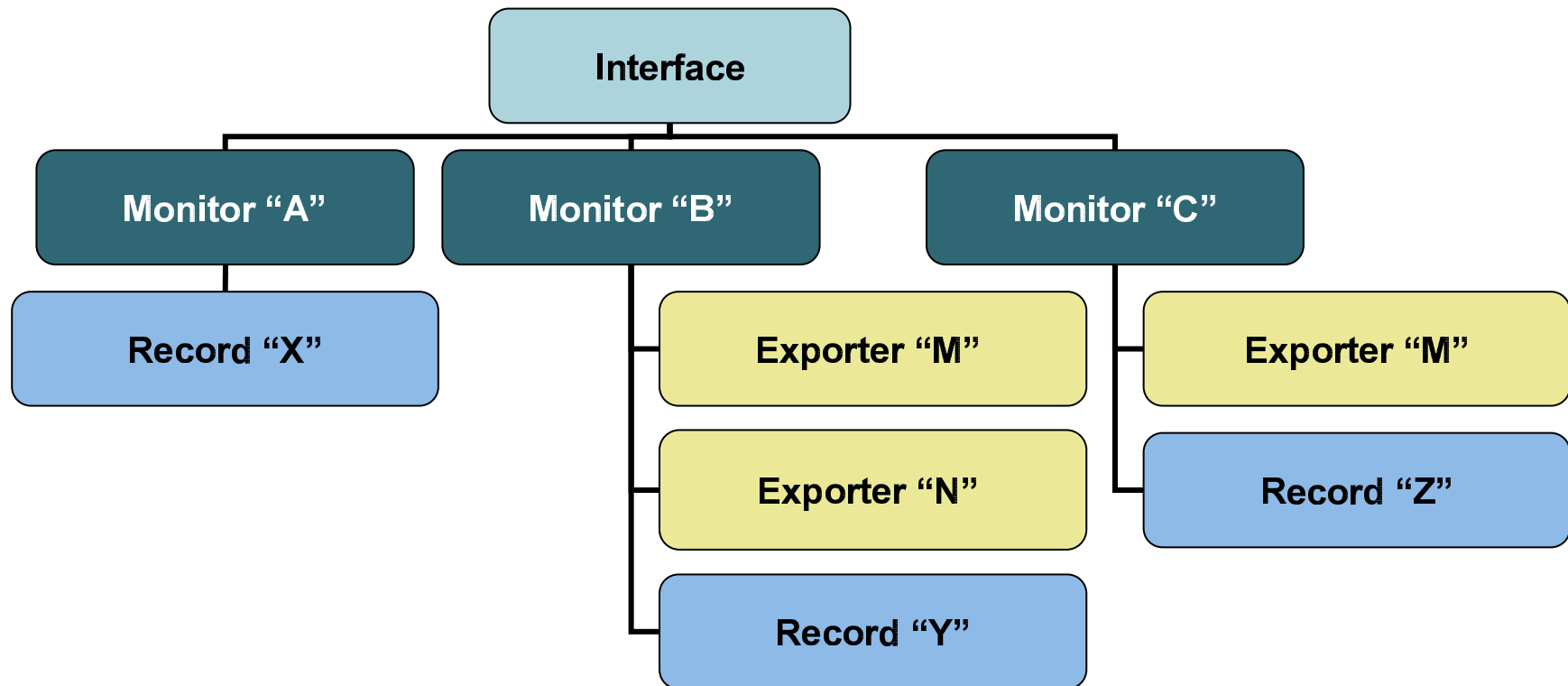
```
flow record icmp-record
  match ipv4 icmp type
  match ipv4 icmp code
  collect counter bytes long
  collect counter packets long

flow monitor traffic-matrix-monitor
  record icmp-record
  cache entries XXXX
  cache type permanent
  cache timeout update 3600
  exporter capacity-planning-collector

interface pos3/0
  ip flow monitor traffic-matrix-monitor
```

**Permanent Cache, with a Record Sent Every Hour**

# Flexible NetFlow Model



- **A single record per monitor**
- **Potentially multiple monitors per interface**
- **Potentially multiple exporters per monitor**



# Flexible NetFlow

- **Version 9 is the only export format supported**
  - IPFIX in the future
- **Current NetFlow features not supported**
  - Top talkers
  - IPv6 in the future
  - Input filters
  - SCTP
  - MIB
- **Platforms:**
  - 800, 1700, 1800, 2600, 2800, 3700, 3800, 7200, 7301
- **Cisco IOS:**
  - 12.4(9)T (June 2006), 12.2(31)SB (3rd quarter 2006), 12.0(33)S

# IETF Front



# IETF: IP Flow Information Export WG (IPFIX)

- **RFC3954 “Cisco Systems NetFlow Services Export Version 9”**
  - NetFlow patent: intellectual property right statement on the IETF website
- **IPFIX is an effort to:**
  - Define the notion of a “standard IP flow”, along with data encoding for IP flows
  - <http://www.ietf.org/html.charters/ipfix-charter.html>
- **RFC3917 “Requirements for IP Flow Information Export”**
  - Gathers all IPFIX requirements for the IPFIX evaluation process
- **RFC3955 “Evaluation of Candidate Protocols for IPFIX”**

# IETF: IP Flow Information Export WG (IPFIX)

- **IPFIX protocol specifications**

  - Changed in terminology but same principles as NetFlow version 9

  - Improvements versus NetFlow version 9: SCTP-PR, security, variable length information element, IANA registration, etc.

  - Generic streaming protocol**, not flow-centric anymore

- **IPFIX information model**

  - Most NetFlow version 9 information elements ID are kept

  - Proprietary information element specification

- **All IPFIX drafts transmitted to the IESG (Internet engineering task force)**

# IETF: Packet Sampling WG (PSAMP)

- **PSAMP is an effort to:**

  - Specify a set of selection operations by which packets are sampled, and describe protocols by which information on sampled packets is reported to applications

- **Sampling and filtering techniques for IP packet selection**

  - To be compliant with PSAMP, we must implement at least one of the mechanisms: sampled NetFlow, NetFlow input filters are already implemented

- **PSAMP protocol specifications**

  - Agreed to use IPFIX for export protocol

- **Information model for packet sampling export**

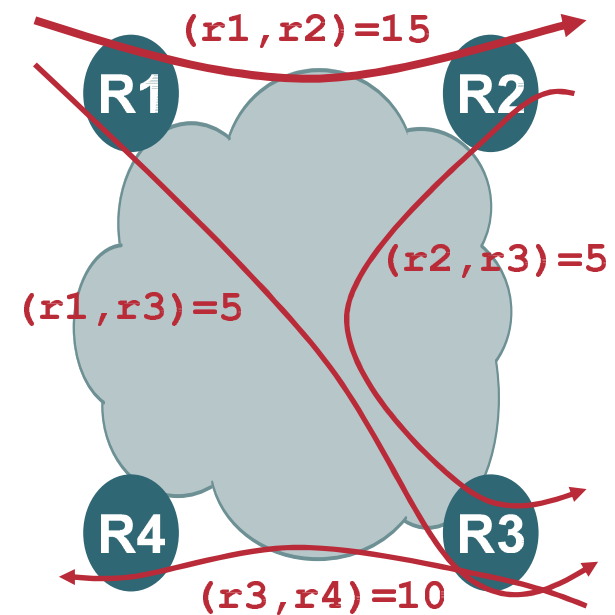
  - Extension of the IPFIX information model

# NetFlow for Capacity Planning



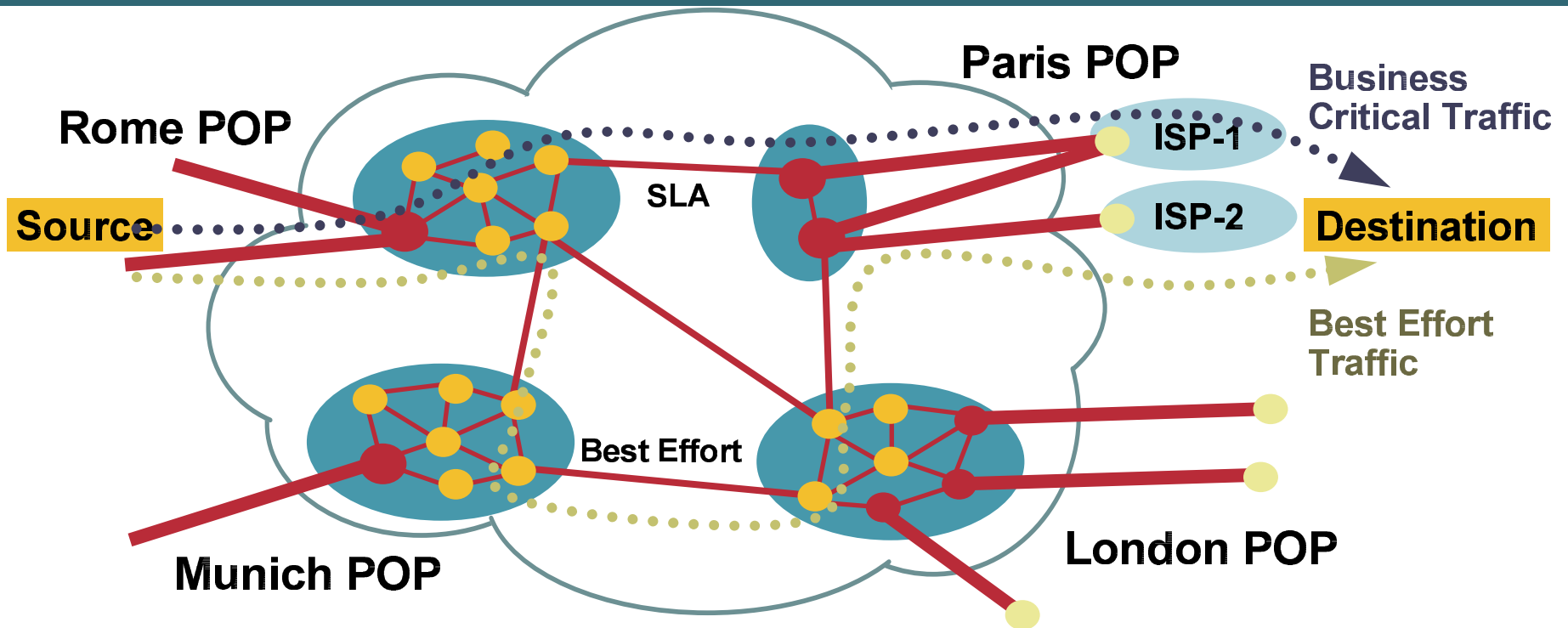
# What Is the Traffic Matrix?

From/to	R1	R2	R3	R4
R1	0	15	5	0
R2	0	0	5	0
R3	0	0	0	10
R4	0	0	0	0



# The Core Traffic Matrix

## Traffic Engineering and Capacity Planning



	Rome Exit Point	Paris Exit Point	London Exit Point	Munich Exit Point
Rome Entry Point	NA (*)	...Mb/s	...Mb/s	...Mb/s
Paris Entry Point	...Mb/s	NA (*)	...Mb/s	...Mb/s
London Exit Point	...Mb/s	...Mb/s	NA (*)	...Mb/s
Munich Exit Point	...Mb/s	...Mb/s	...Mb/s	NA (*)

(\*) Potentially Local Exchange Traffic

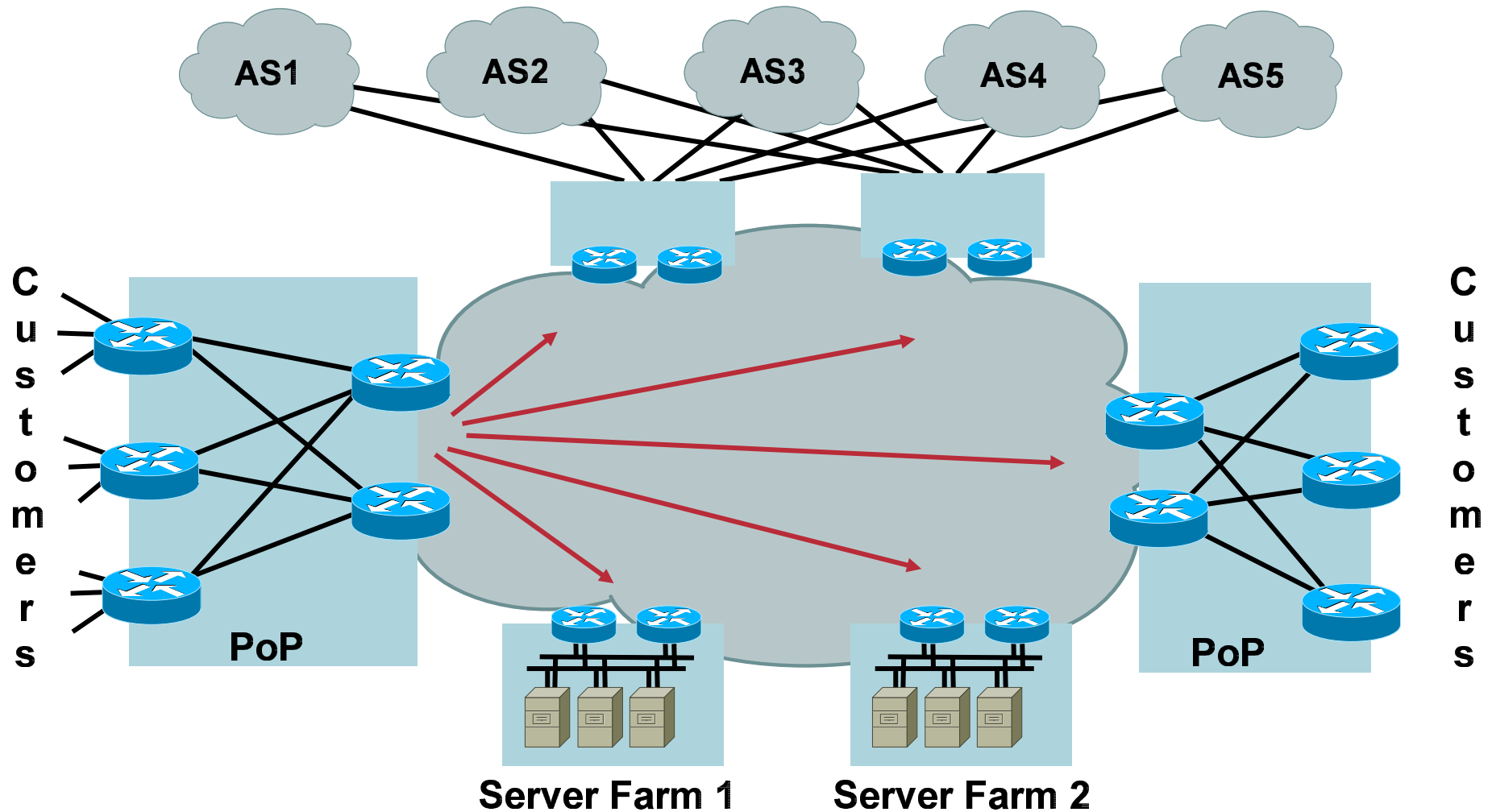


# Core Capacity Planning

## The Big Picture

1. The ability to offer **SLAs is dependent** upon ensuring that core network bandwidth is adequately provisioned
2. Adequate provisioning (without gross over provisioning) is dependent upon **accurate core capacity planning**
3. Accurate core capacity planning is dependent upon understanding the **core traffic matrix** and flows and mapping these to the underlying topology
4. A tool for “what if” scenarios

# We Need the Core Traffic Matrix



**“PoP to PoP”: Access Router or Core Router**

# NetFlow BGP Next Hop TOS Aggregation Flow Keys

## Key Fields (Uniquely Identifies the Flow)

- **Origin AS**
- **Destination AS**
- **Inbound Interface**
- **Output Interface**
- **ToS/DSCP (\*)**
- **Next BGP Hop**

**(\*) Before Any Recoloring**

## Additional Export Fields

- **Flows**
- **Packets**
- **Bytes**
- **First SysUptime**
- **Last SysUptime**

# Core Traffic Matrix with Flexible NetFlow

## Key Fields (Uniquely Identifies the Flow)

- ~~Origin AS~~
- Destination AS
- Inbound Interface
- ~~Output Interface~~
- ToS/DSCP (\*)
- Next BGP Hop

## Additional Export Fields

- ~~Flows~~
- ~~Packets~~
- Bytes
- First SysUptime
- Last SysUptime

(\*) Before Any Recoloring

- Less flow records, less CPU
- Potentially higher sampling rate for a better accuracy

# Core Traffic Matrix with Flexible NetFlow Configuration Example

```
flow record traffic-matrix-record
  match routing destination as
  match interface input
  match ipv4 dscp
  match routing next-hop address ipv4 bgp
  collect counter bytes long
  collect timestamp sys-uptime first
  collect timestamp sys-uptime last
```

```
flow monitor traffic-matrix-monitor
  record traffic-matrix-record
  cache entries 10000
  cache type normal
  exporter capacity-planning-collector
```

```
interface pos3/0
  ip flow monitor traffic-matrix-monitor
```

**Note: Export Less Flow Records with a Permanent Cache**

# Conclusion



# NetFlow Summary and Conclusion

- **NetFlow provides input for **security**, accounting, performance, and billing applications**
- **Flexible NetFlow is a major enhancement, as we have user-defined flow records**
- **IPFIX and PSAMP IETF WG based on NetFlow version 9**
- **A lot of new features have been added**
- **Stay tuned for more 😊**

# References

- **NetFlow**

<http://www.cisco.com/go/netflow>

- **Cisco Network Accounting Services**

**Comparison of Cisco NetFlow versus other available accounting technologies**

[http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/nwact\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/nwact_wp.htm)

- **Cisco IT Case Study**

[http://business.cisco.com/prod/tree.taf%3Fasset\\_id=106882&IT=104252&public\\_view=true&kbns=1.html](http://business.cisco.com/prod/tree.taf%3Fasset_id=106882&IT=104252&public_view=true&kbns=1.html)

- **A complete white paper**

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm>