NetFlow/IPFIX Usage in Network Management

Benoit Claise, Cisco Systems

EMANICS/IRTF-NMRG Workshop, October 2008

Extensibility and Flexibility Requirements Phases Approach

- Traditional NetFlow with the v5, v7, or v8 NetFlow export
  - New requirements: build something flexible and extensible
- Phase 1: NetFlow version 9
  - Advantages: extensibility
    - Integrate new technologies/data types quicker (IPv6, BGP next hop, Layer 2, etc.)
    - Integrate new aggregations quicker
  - Note: for now, the template definitions are fixed

Exporting Process

- Phase 2: Flexible NetFlow
  - Advantages: cache and export content flexibility
    - User selection of flow keys
    - User definition of the records

Metering Process

Flexible Flow Record: Key Fields

IPv4	
IP (Source or Destination)	Payload Size
Prefix (Source or Destination)	Packet Section (Header)
Mask (Source or Destination)	Packet Section (Payload)
Minimum-Mask (Source or Destination)	TTL
Protocol	Options bitmap
Fragmentation Flags	Version
Fragmentation Offset	Precedence
ID	DSCP
Header Length	TOS
Total Length	

Routing
src or dest AS
Peer AS
Traffic Index
Forwarding Status
IGP Next Hop
BGP Next Hop

Flow
Sampler ID
Direction

Interface
Input
Output

Transport	
Destination Port	TCP Flag: ACK
Source Port	TCP Flag: CWR
ICMP Code	TCP Flag: ECE
ICMP Type	TCP Flag: FIN
IGMP Type	TCP Flag: PSH
TCP ACK Number	TCP Flag: RST
TCP Header Length	TCP Flag: SYN
TCP Sequence Number	TCP Flag: URG
TCP Window-Size	UDP Message Length
TCP Source Port	UDP Source Port
TCP Destination Port	UDP Destination Port
TCP Urgent Pointer	

Flexible Flow Record: Key Fields

IPv6	
IP (Source or Destination)	Payload Size
Prefix (Source or Destination)	Packet Section (Header)
Mask (Source or Destination)	Packet Section (Payload)
Minimum-Mask (Source or Destination)	DSCP
Protocol	Extension Headers
Traffic Class	Hop-Limit
Flow Label	Length
Option Header	Next-header
Header Length	Version
Payload Length	

Multicast
Replication Factor
RPF Check Drop
Is-Multicast

Layer 2
Source VLAN
Destination VLAN
Source MAC address
Destination MAC address

Flexible Flow Record—Non-Key Fields

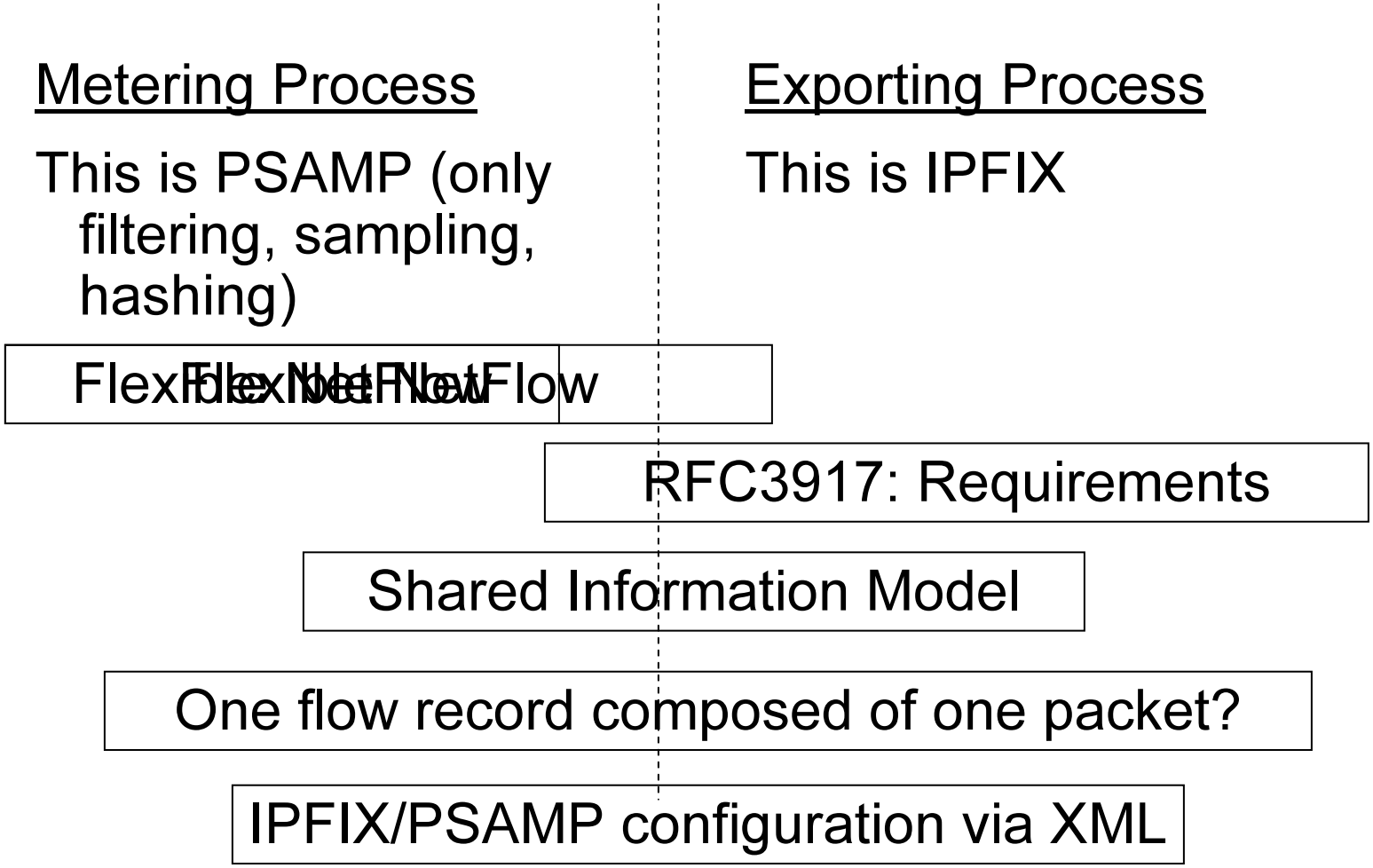
Counters
Bytes
Bytes Long
Bytes Square Sum
Bytes Square Sum Long
Packets
Packets Long

Timestamp
sysUpTime First Packet
sysUpTime First Packet

IPv4
Total Length Minimum
Total Length Maximum
TTL Minimum
TTL Maximum

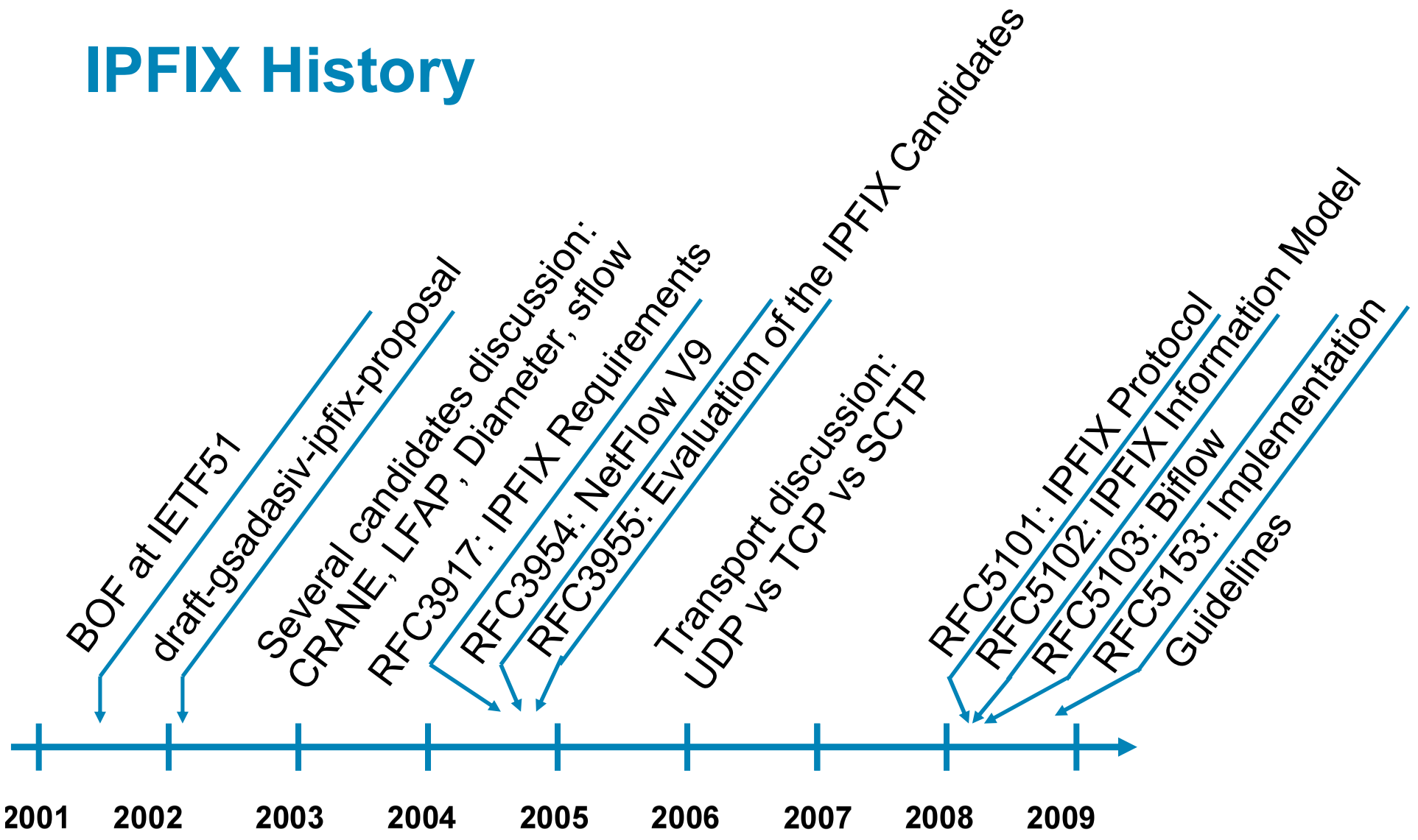
- Plus any of the potential “key” fields: will be the value from the first packet in the flow

What about IPFIX and PSAMP?

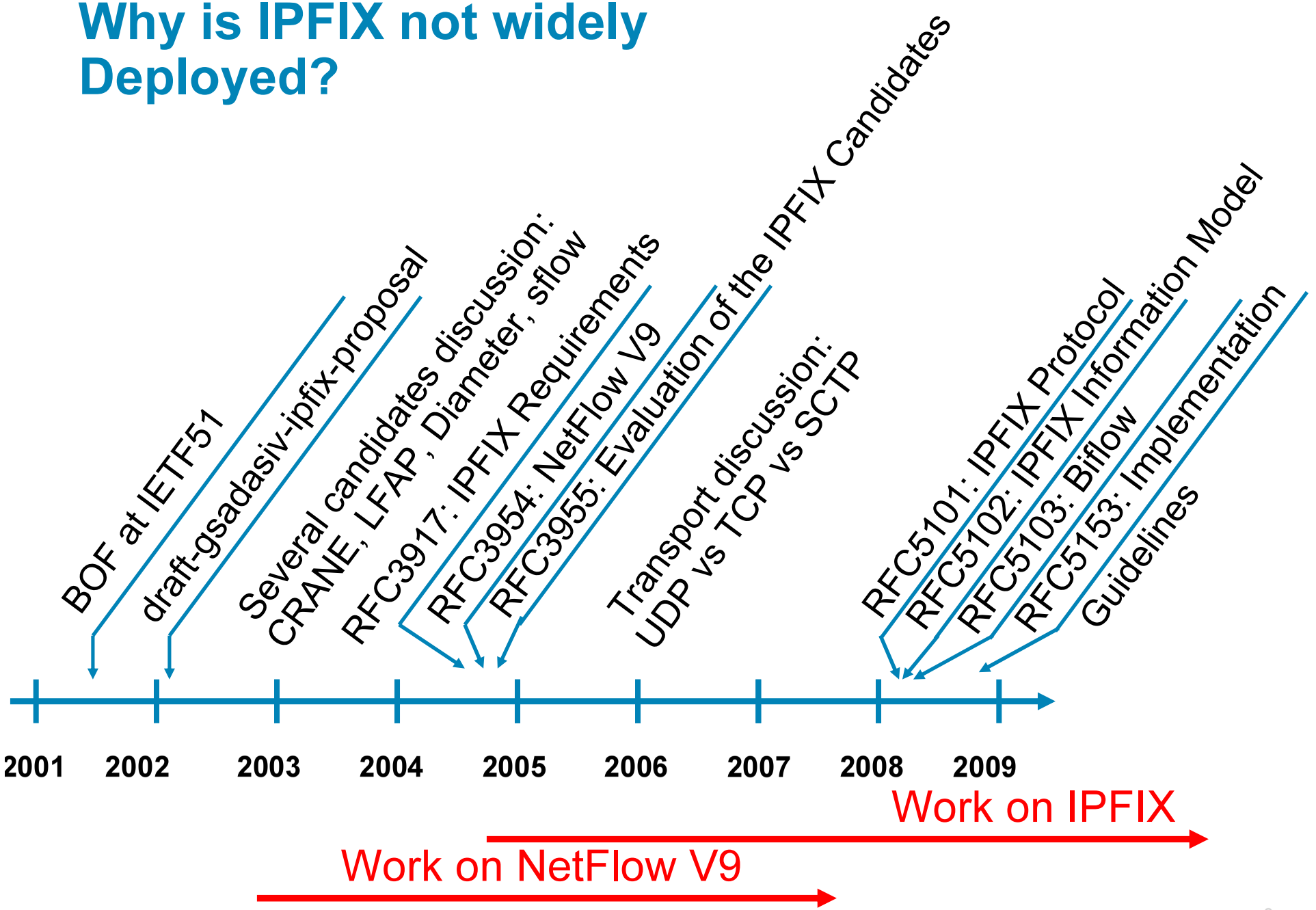


IPFIX and PSAMP are complimentary: no more boundary

IPFIX History



Why is IPFIX not widely Deployed?



Why is IPFIX not widely Deployed Yet?

- What's missing in NetFlow v9:
  - Variable length, a problem with strings
  - Security
  - Template Withdrawal Message
- The value is in the Metering Process
- Note: we will implement IPFIX Export per SCTP Stream
  draft-ietf-ipfix-export-per-sctp-stream-00

NetFlow as Alternative to syslog?

- Firewall: better throughput and connection/s by using NetFlow as opposed to Syslog
- What's in a name?

  We do export MAC address, VLAN, MPLS info

  Remove IP ;-)

  We can define anything we want as Information Elements

  IPFIX became a generic streaming protocol

  Change to "IP Flexible Information eXport?"

Metering Process Challenge
Flexible NetFlow is very Flexible…

- Easier to shoot yourself in the foot
- Let's not forget that the router still has to route packets
- Might need some consulting services for every customers

  No one size fits all

Metering Process Challenge
Flexible NetFlow is very Flexible…

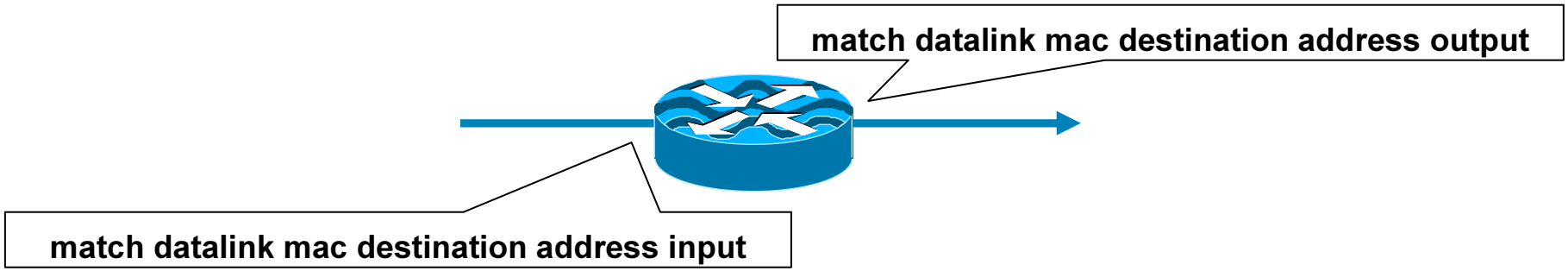
```
match datalink mac {destination address input | source
address {input | output}}
```

destination address input destination MAC address of packets received by the router

source address source MAC address

input Packets received by the router

output Packets transmitted by the router



Example of MPLS PE with QoS … and a distributed system

Performance Challenge Moving Bottleneck

- “consume a lot of CPU”
  -> packet sampling
  -> metering process in hardware
- “collision in the cache”
  -> improved the hash function
  -> increased the cache size
- “consume much bandwidth”
  -> flexible flow record
  -> per interface, per direction
  -> export cache type per collector
- Next one: the collector?
  -> do we need an overlay network of collectors?
  for example, for data retention

IPFIX Limitation
Field Not Observed

- No specific value for “not observed”

  Example: ICMP type, port number, TCP window size for UDP traffic

  Which one to choose?

  Specific value per information element?

- “If a specific Information Element is required by a Template, but is not available in observed packets, the Exporting Process MAY choose to export Flow Records without this Information Element in a Data Record defined by a new Template” RFC5101

IPFIX Limitation
No Structure Data

- "MPLS Label and MPLS Label Position" lesson learned

  The content of one value depends on the content of another one: this breaks the design

  RFC5102: mplsTopLabelStackSection,
  mplsLabelStackSection[2-9]

- List export

  output interface for multicast

  AS in the AS-PATH

  RFC5101 "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."

  Hardcode the intelligence in the collector?

Challenge
Overload The Options Template

- Options Template is used for …

  Statistics Information in the IPFIX protocol (The Metering Process Statistics, The Metering Process Reliability Statistics, Exporting Process Reliability Statistics)

  The Flow Keys Option Template

  Reducing Redundancy

  ifIndex/interface name matching

  Etc…

- Yes everything is possible with Options Template Record
- This complicates the collecting process

IPFIX/PSAMP Future Architecture?

- Enable several cache type for different purposes, exported to different exporters
- Optimized cache type

  For example, the core traffic matrix
- Generic cache type

  The collector would quickly configure a new cache type, with a specific filter, for verification

  For example, security

"Advanced Network Monitoring Brings Life to the Awareness Plane"
Andreas Kind, Spyros Denazis, Xenofontas Dimitropoulos, Benoit Claise
IEEE Communications Magazine, 2008

IPFIX/PSAMP Future Challenge

- More and more observation points
  - Before/after QoS
  - Before/after WAN optimization
  - On different line cards (i.e. different observation domain)
- Flow record correlation is a MUST
  - Biflow is good start
- Two different aspects to the IPFIX Mediation Function current work …

What if we could start IPFIX from scratch?

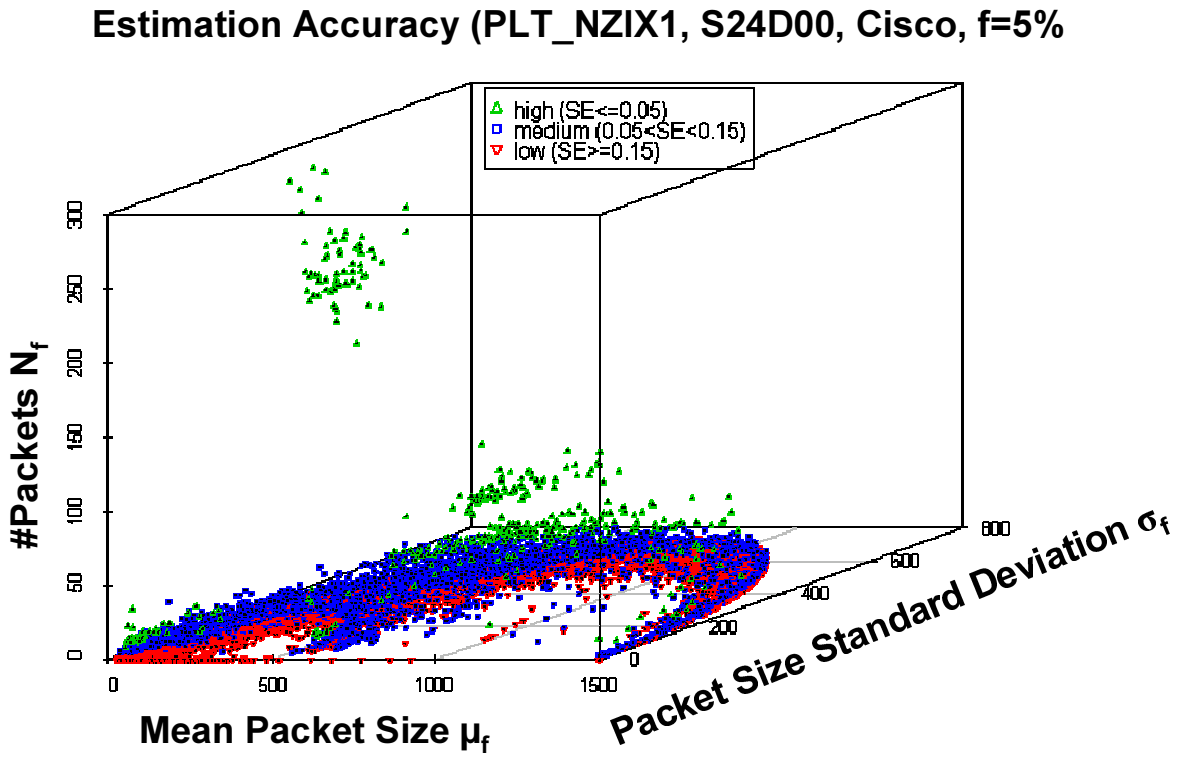
- Flexible Header
  - Containing information elements
- Some more information element attributes
  - in the template definition: pre or post, key-field or not, etc.
  - In the flow record: observed or not
- XML templates as an answer to structure data?

Accuracy of (Packet) Sampled NetFlow Research Project

- Mathematical model valid for random sampled NetFlow
- Square sum of bytes available in Flexible NetFlow

  "collect counter bytes squared long" in the CLI

- "Packet Sampling for Flow Accounting: Challenges and Limitations", Tanja Zseby , Thomas Hirsch, Benoit Claise, PAM 2008



$$StdErr_{rel}[\hat{Sum}_f] = \frac{StdErr_{abs}[\hat{Sum}_f]}{Sum_f} = \frac{\sqrt{\frac{N^2}{n} \cdot \left(\sigma^2_{x_f} \cdot P_f + \mu^2_{x_f} \cdot (P_f - P_f^2)\right)}}{N_f \cdot \mu_{x_f}}$$

Some more Reading