Advanced NetFlow for Service Providers

Aamer Akhter (aa@cisco.com)
Benoit Claise (bclaise@cisco.com)

Agenda

- Introduction
- NetFlow Version 9
- Interesting Features on Traditional NetFlow
- Flexible NetFlow
- NetFlow for Security
- NetFlow for Application Visibility
- NetFlow Performance

NetFlow – What is it?

- Developed and patented at Cisco® Systems in 1996
- NetFlow is a standard for acquiring IP operational data
- Provides network and security monitoring, network planning, traffic analysis, and IP accounting
- IETF's IPFIX (RFC 5101) based on NetFlow v9 (with changes)

Network World Article – NetFlow Adoption on the raise

http://www.networkworld.com/newsletters/nsm/2005/0314nsm1.html

Key Concept — *Flow Scalability

- Packet capture is like a wiretap
- *Flow is like a phone bill
- This level of granularity allows *Flow to scale for very large amounts of traffic

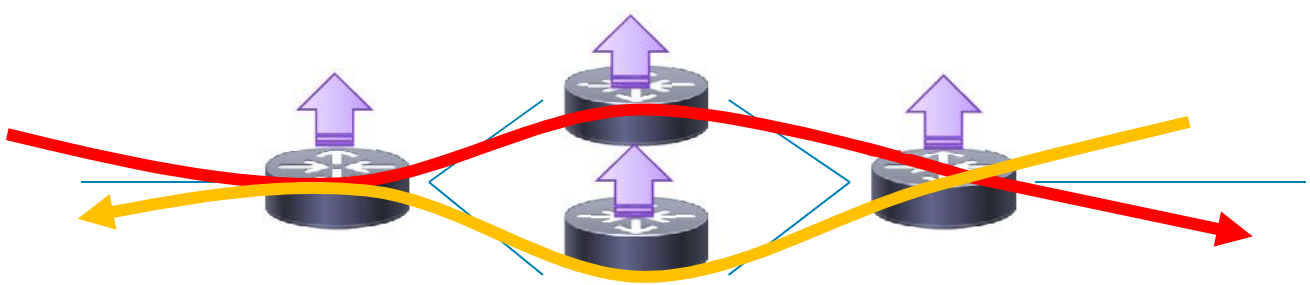
  We can learn a lot from studying the phone bill

  Who's talking to whom, over what protocols and ports, for how long, at what speed, for what duration, etc.

  *Flow is a form of telemetry pushed from the routers/switches — each one can be a sensor

Key Concept — *Flow Follows the Topology

- Network traffic is often asymmetrical, even on small networks
- Probes typically require engineered symmetry
- This means that with *Flow, there's no need to engineer the network around the instrumentation
  - We can follow traffic through the network over its natural path
  - We can see pps, bps, packet-size, QoS markings, TCP flags, etc. for specific apps/services at each point in the network
  - We can validate traffic engineering, policy enforcement, etc. at any point in the topology, as long as *Flow is enabled



Why *Flow?
Network Operator Benefits

- Understand
  - Productivity and utilization of assets in the network
  - Application and network usage
  - Impact of network changes and services
  - NetFlow answers the *who, what, when, where*, and *how* network traffic is flowing
- Detect and classify security incidents with proven threat defence
- Improve network usage and application performance

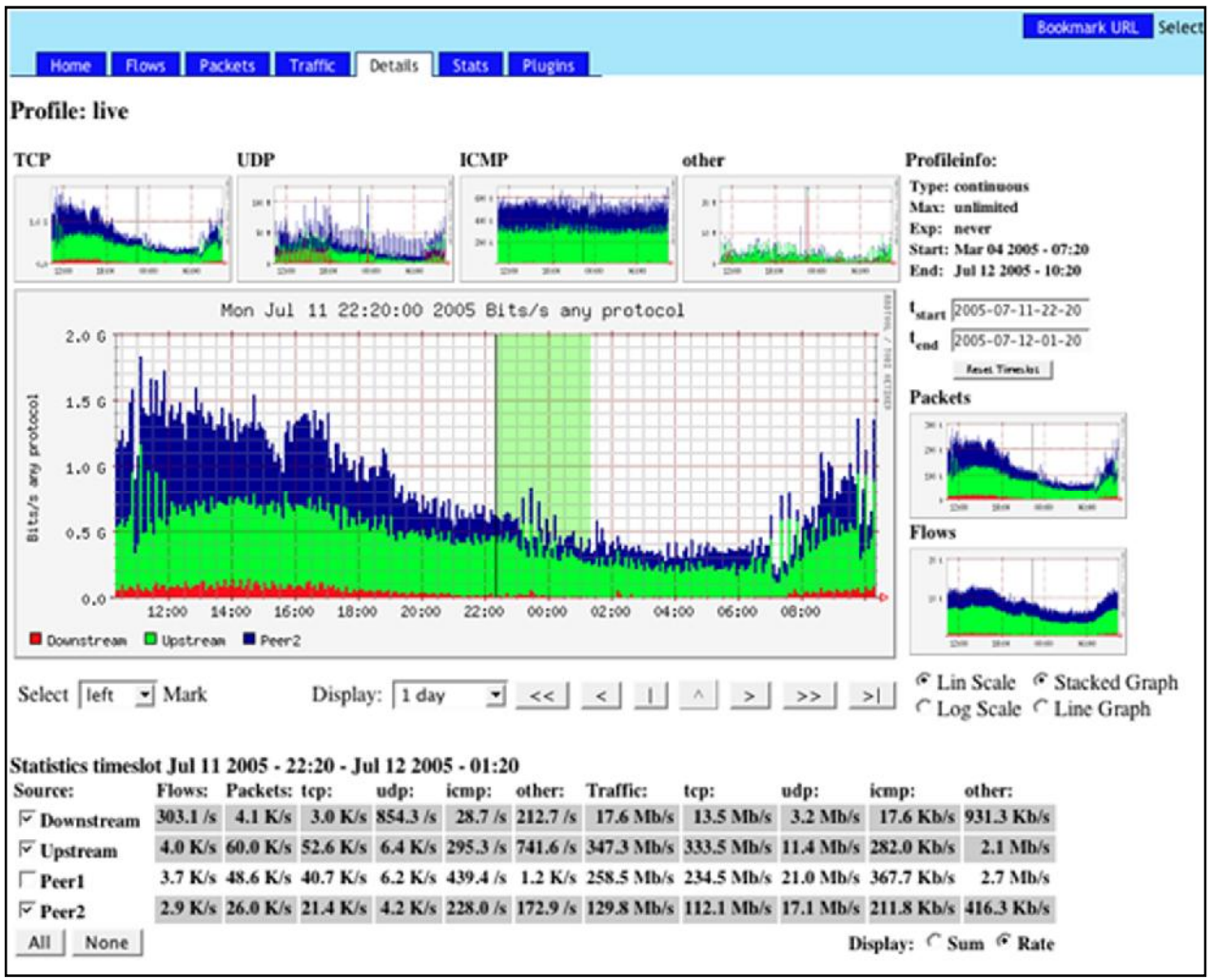
Principal *Flow Uses

Service Provider	Enterprise
Peering Arrangements	Internet access monitoring (protocol distribution, where traffic is going/coming)
Network Planning	User Monitoring
Traffic Engineering	Application Monitoring
Accounting and Billing	Chargeback billing for departments
Security Monitoring	Security Monitoring
Performance Monitoring	Performance Monitoring

Examples of use

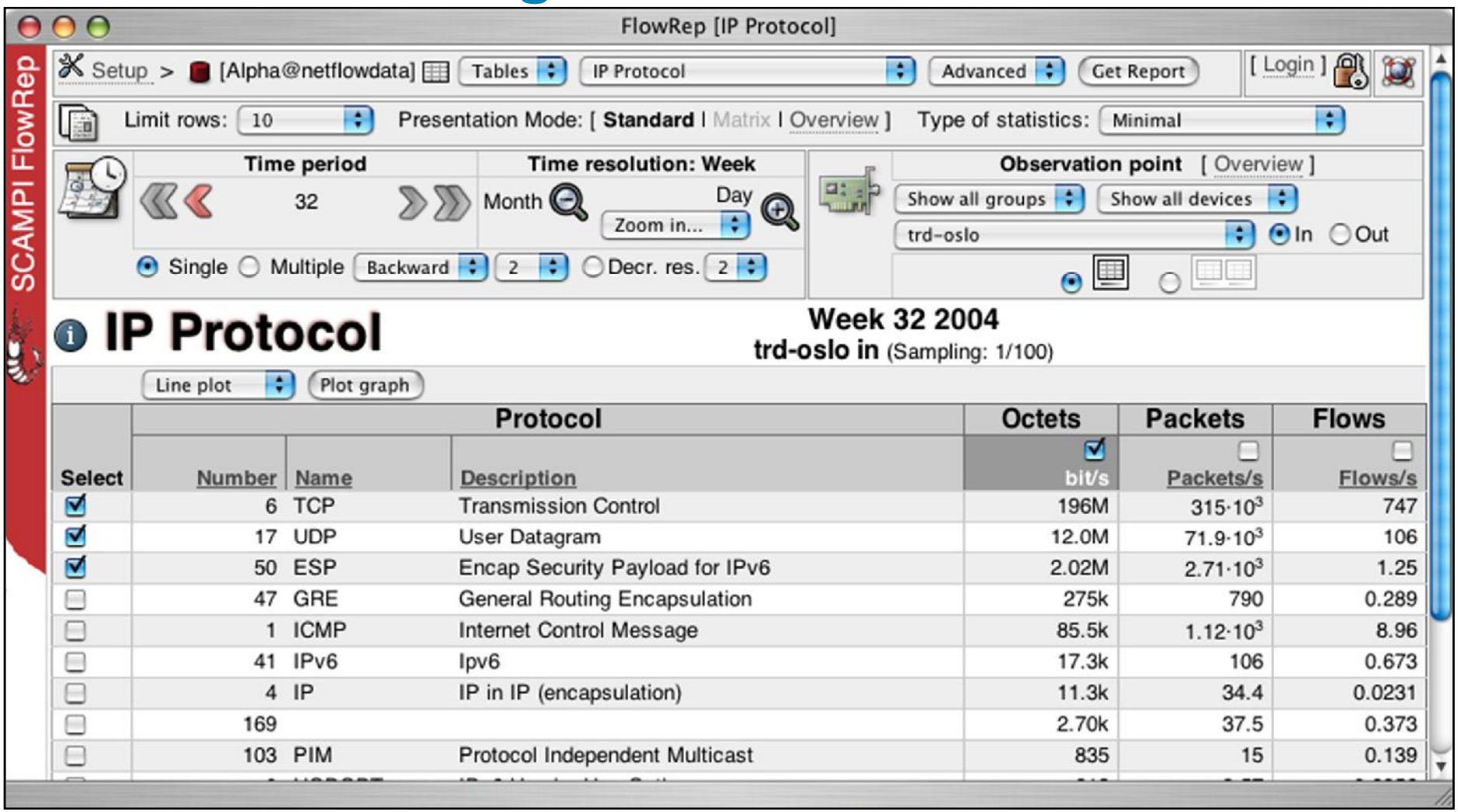
Customer Challenge	Description	Problem Situation	NetFlow Resolution
Security	Detect SQL Slammer on day one	Detrimental incapacity of servers	NetFlow day-zero anomaly detection
Traffic Analysis	Bandwidth Hog	- Sluggish network performance - Single user application monopolizing network	Cost savings of $7k in labor costs
Traffic Analysis	Full Circuit	Circuit 100% utilized	Quickly tracked problem and saved 300 hours = $34k in labor costs
Capacity Planning	Slow network performance	- More servers and bandwidth added - Users still complained - Rented RMON probes - didn't work	Cost savings of $126k in probe costs
Capacity Planning	Poor network performance – low bandwidth	We need more bandwidth	Tracked point of slowdown – saved $36k per yr. circuits

NetFlow—nfdump and nfsen



Source: http://nfsen.sourceforge.net

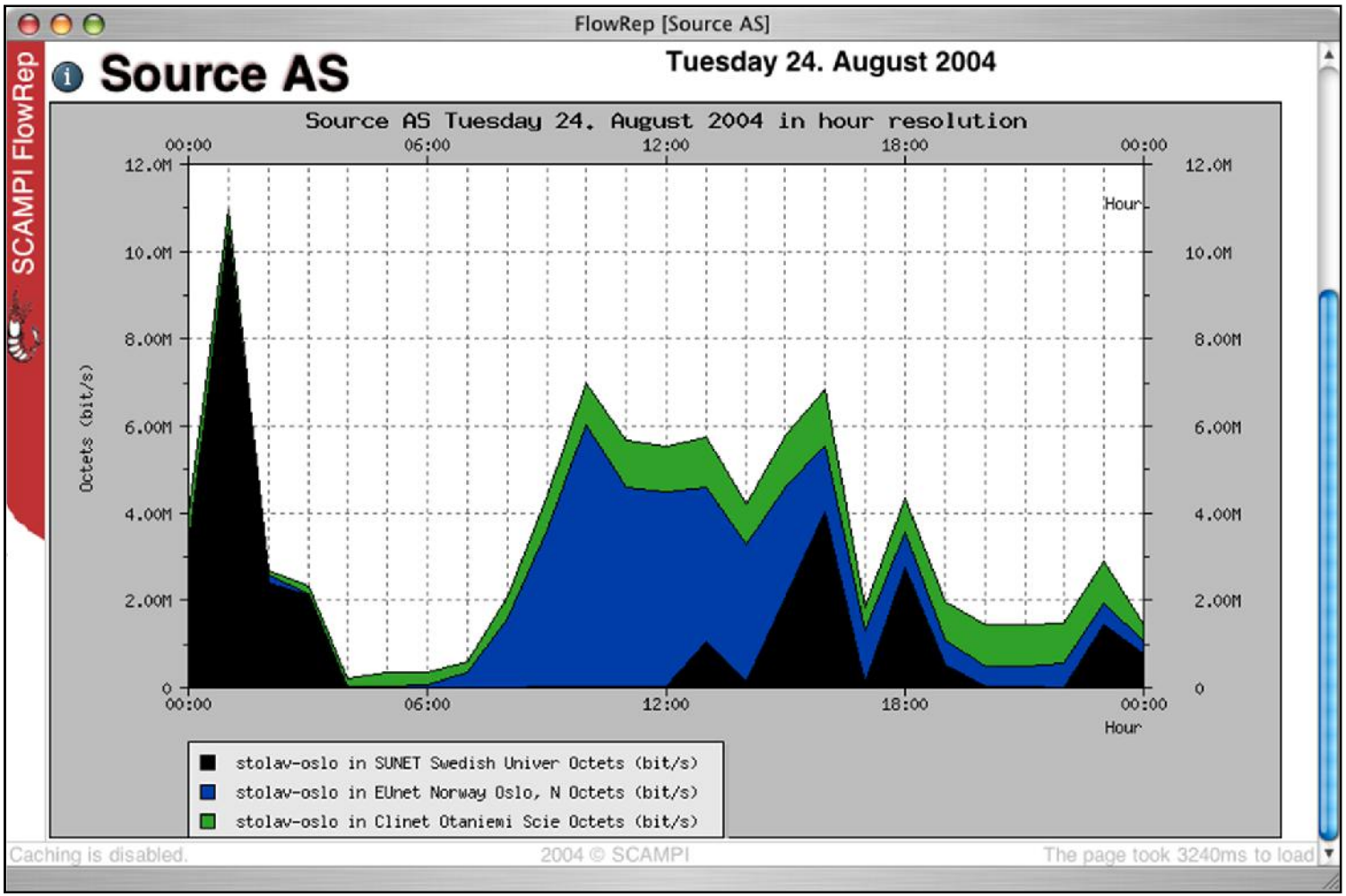
NetFlow—Stager



Select	Protocol Number	Name	Description	Octets bit/s	Packets Packets/s	Flows Flows/s
☑	6	TCP	Transmission Control	196M	$315 \cdot 10^3$	747
☑	17	UDP	User Datagram	12.0M	$71.9 \cdot 10^3$	106
☑	50	ESP	Encap Security Payload for IPv6	2.02M	$2.71 \cdot 10^3$	1.25
☐	47	GRE	General Routing Encapsulation	275k	790	0.289
☐	1	ICMP	Internet Control Message	85.5k	$1.12 \cdot 10^3$	8.96
☐	41	IPv6	Ipv6	17.3k	106	0.673
☐	4	IP	IP in IP (encapsulation)	11.3k	34.4	0.0231
☐	169			2.70k	37.5	0.373
☐	103	PIM	Protocol Independent Multicast	835	15	0.139

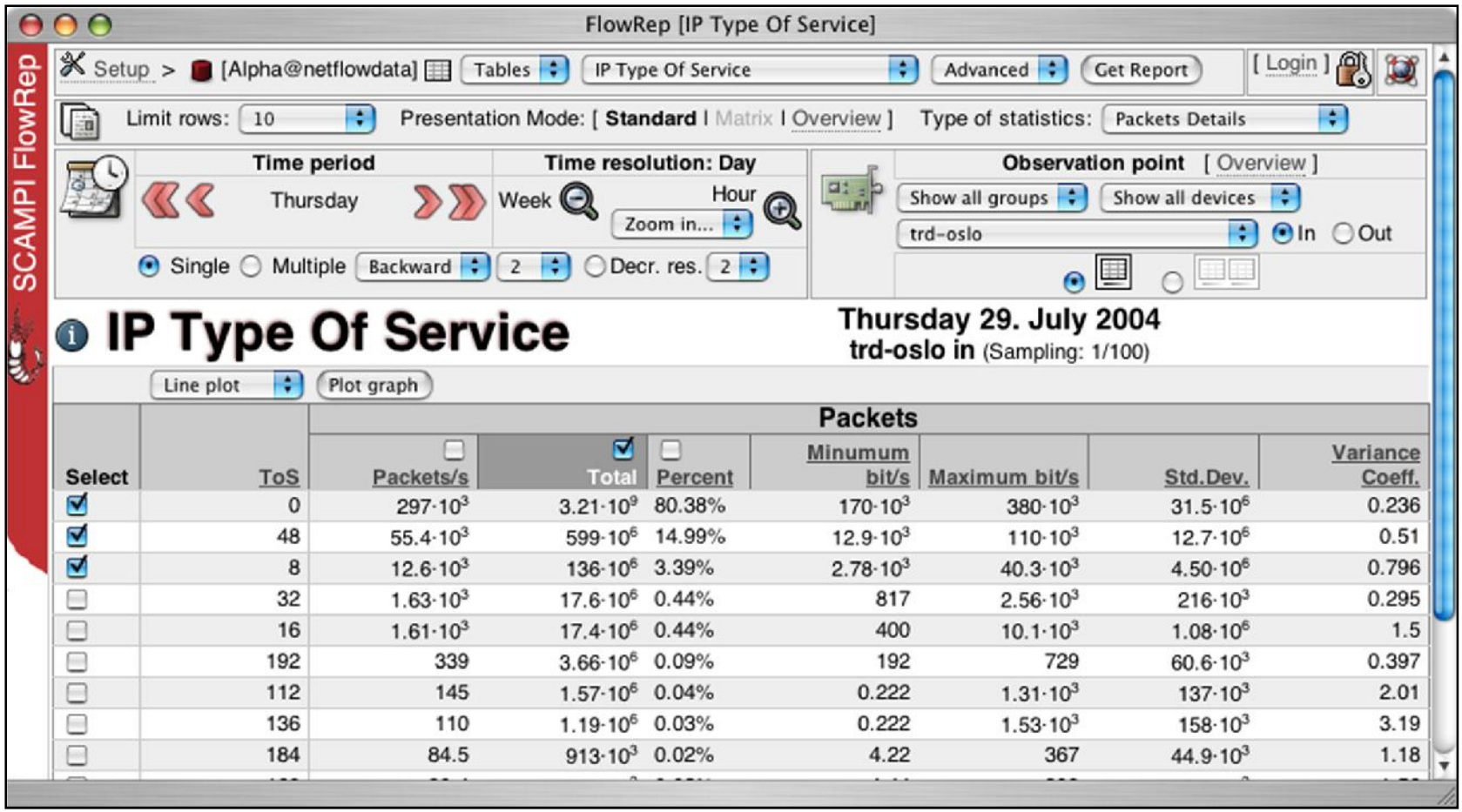
Source: UNINETT

NetFlow—Stager (Cont.)



Source: UNINETT

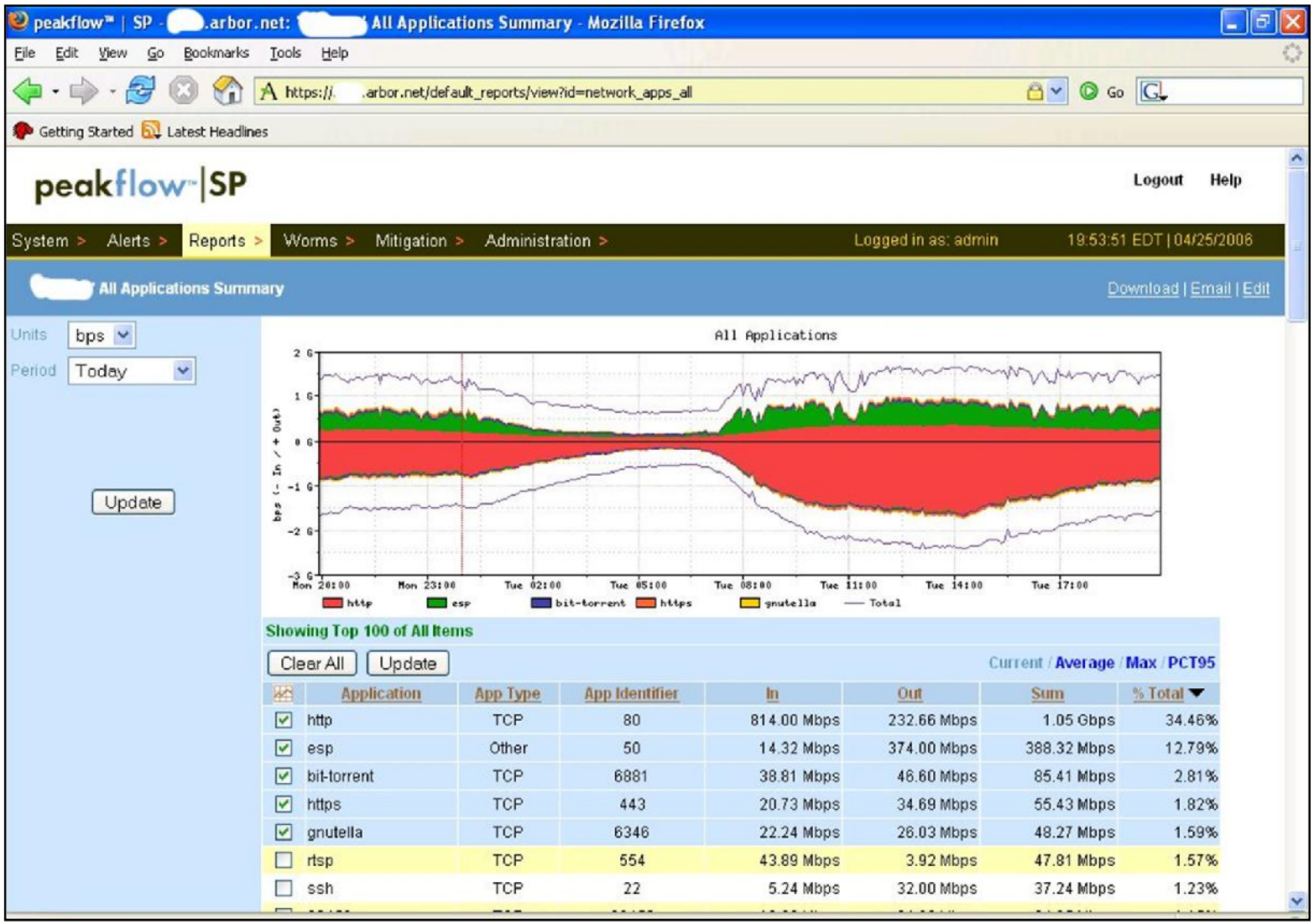
NetFlow—Stager (Cont.)



Select	ToS	Packets						
		Packets/s	Total	Percent	Minumum bit/s	Maximum bit/s	Std.Dev.	Variance Coeff.
☑	0	$297 \cdot 10^3$	$3.21 \cdot 10^9$	80.38%	$170 \cdot 10^3$	$380 \cdot 10^3$	$31.5 \cdot 10^6$	0.236
☑	48	$55.4 \cdot 10^3$	$599 \cdot 10^6$	14.99%	$12.9 \cdot 10^3$	$110 \cdot 10^3$	$12.7 \cdot 10^6$	0.51
☑	8	$12.6 \cdot 10^3$	$136 \cdot 10^6$	3.39%	$2.78 \cdot 10^3$	$40.3 \cdot 10^3$	$4.50 \cdot 10^6$	0.796
☐	32	$1.63 \cdot 10^3$	$17.6 \cdot 10^6$	0.44%	817	$2.56 \cdot 10^3$	$216 \cdot 10^3$	0.295
☐	16	$1.61 \cdot 10^3$	$17.4 \cdot 10^6$	0.44%	400	$10.1 \cdot 10^3$	$1.08 \cdot 10^6$	1.5
☐	192	339	$3.66 \cdot 10^6$	0.09%	192	729	$60.6 \cdot 10^3$	0.397
☐	112	145	$1.57 \cdot 10^6$	0.04%	0.222	$1.31 \cdot 10^3$	$137 \cdot 10^3$	2.01
☐	136	110	$1.19 \cdot 10^6$	0.03%	0.222	$1.53 \cdot 10^3$	$158 \cdot 10^3$	3.19
☐	184	84.5	$913 \cdot 10^3$	0.02%	4.22	367	$44.9 \cdot 10^3$	1.18

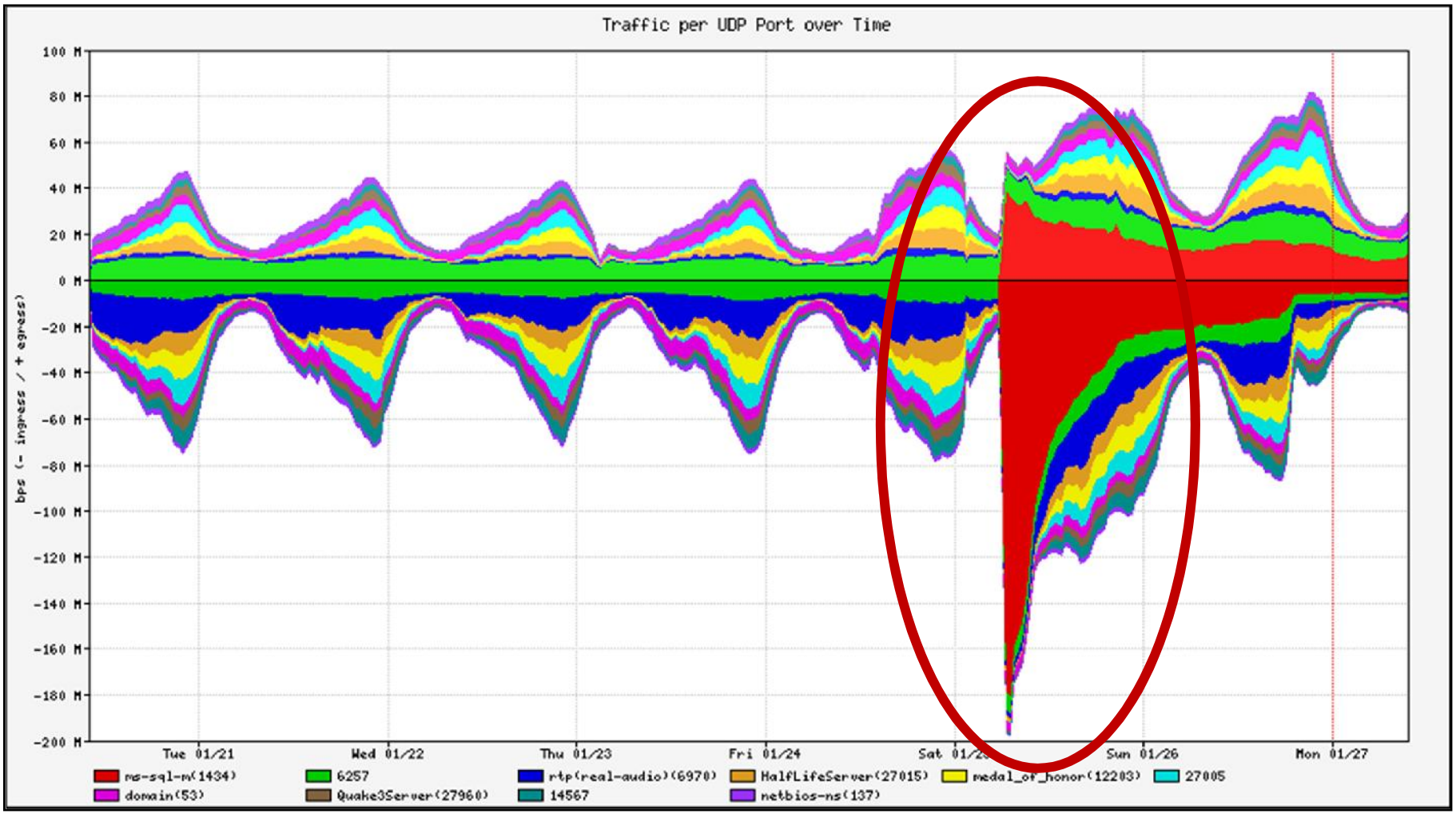
Source: UNINETT

Arbor Peakflow SP — Application Distribution



Application	App Type	App Identifier	In	Out	Sum	% Total
http	TCP	80	814.00 Mbps	232.66 Mbps	1.05 Gbps	34.46%
esp	Other	50	14.32 Mbps	374.00 Mbps	388.32 Mbps	12.79%
bit-torrent	TCP	6881	38.81 Mbps	46.60 Mbps	85.41 Mbps	2.81%
https	TCP	443	20.73 Mbps	34.69 Mbps	55.43 Mbps	1.82%
gnutella	TCP	6346	22.24 Mbps	26.03 Mbps	48.27 Mbps	1.59%
rtsp	TCP	554	43.89 Mbps	3.92 Mbps	47.81 Mbps	1.57%
ssh	TCP	22	5.24 Mbps	32.00 Mbps	37.24 Mbps	1.23%

Example—SQL Slammer



*Flow and SNMP

- All the *flow protocols are essentially push technologies:
  - Information is sent asynchronously from measurement node
  - Post-processing (aggregation) might be done on router/switch
  - Information is exported and usually immediately expired
  - NMS does not decide on rate of information
- SNMP is a pull technology
  - NMS needs to decide when and how often to poll device
  - Device may not retain information when polled
  - Device does not decide on rate of information
  - Correlated information may require multiple transactions

BGP Accounting

- BGP accounting provides a way of getting prefix traffic information
- BGP prefixes are colored in one of X colors
- Each color has counters (byte/packet) associated
- When traffic from/to prefix is received/sent counters are incremented.

- Easy form of aggregation – usually limited 8 buckets
- Information organized only around buckets, no additional information provided.

sFlow

- Created by InMon (sells sFlow collectors)
- sFlow v2, v4 and v5 (v2 and v4 deprecated)
- sFlow somewhere in between NFv8 and NFv9/IPFIX
  - Extendable set of fields – called structures (fixed templates)
  - sampling (sampling is required part of sFlow)
- Supported by (generally switch vendors) Alcatel, Extreme, Force10, HP, Hitachi (*)
- Incompatible with NetFlow V9/IPFIX, but some collectors support both NFv9 and sFlow

* Complete list at sFlow.org

J-Flow / cflow

- J-Flow and cflowd is essentially NetFlow
- NetFlow collectors will support J-Flow/cflow output
- J-Flow and cflow are terms used by Juniper(v5, v8, v9)
- cflow term used by Alcatel (v5, v8, v9)
- Implementations do not support flexible templates

NetStream

- NetStream comes in three formats: v5, v8 and v9
  - Essentially mirroring NetFlow v5, v8 and v9
- Generally easily supported by NetFlow collectors
  - However differences exist between NetStream v9 and NF v9
    - eg: NetFlow represents interfaces using ifIndex (standard MIB), NetStream represents using proprietary interface MIB
  - Collector needs to be explicitly told record is NetStream
- NetStream is supported by 3COM and Huawei
- Implementations do not support flexible templates

NetFlow

- V5, v8 and v9 export formats exist
- IETF standard (IPFIX) is based on v9
- NetFlow v9 Documented in RFC3954 (informational)
  - Lack of regulation has lead to minor (and corrected) collisions in field (nProbe) identifiers
- Enjoys wide collector support.
- NFv9 collectors generally need minor tweaks to support IPFIX

- Supported by cisco, Alcatel, Juniper (as J-Flow/cflow), Packeteer (v5), 3COM/ Huawei (sort of), Riverbed, Adtran, Enterasys, wide open-source support
- Cisco implementations support flexible templates
  - Providing flexible reports down the field level

IETF: IP Flow Information Export WG (IPFIX)

- IPFIX protocol specifications

  Changes in terminology but same NetFlow Version 9 principles (IPFIX version field says '10')

  - Improvements vs. NetFlow v9: SCTP-PR, security, variable length information element, IANA registration, etc.
  - Generic streaming protocol, not flow-centric anymore
  - Security:

    Threat: confidentiality, integrity, authorization

    Solution: DTLS on SCTP-PR

    Anonymization draft

- IPFIX information model

  Most NetFlow v9 information elements ID are kept

  Proprietary information element specification

IETF: IP Flow Information Export WG (IPFIX)

- RFC3954 Cisco Systems NetFlow Services Export Version 9
- RFC3917 Requirements for IP Flow Information Export

  Gathers all IPFIX requirements for the IPFIX evaluation process
- RFC3955 Evaluation of Candidate Protocols for IPFIX
- RFC5101 Specification of the IPFIX Protocol for the Exchange of IP Traffic Flow Information
- RFC5102 Information Model for IP Flow Information Export
- RFC5103 “Bidirectional Flow Export using IP Flow Information Export (IPFIX)”

IPFIX: Interesting Drafts

- Export of Application Information in IPFIX
  draft-claise-export-application-info-in-ipfix
- Exporting MIB variables using the IPFIX Protocol
  draft-johnson-ipfix-mib-variable-export
- Export of Structured Data in IPFIX
  draft-ietf-ipfix-structured-data
- IP Flow Anonymisation Support
  draft-ietf-ipfix-anon
- Information Elements for Flow Performance Measurement
  draft-akhter-ipfix-perfmon

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

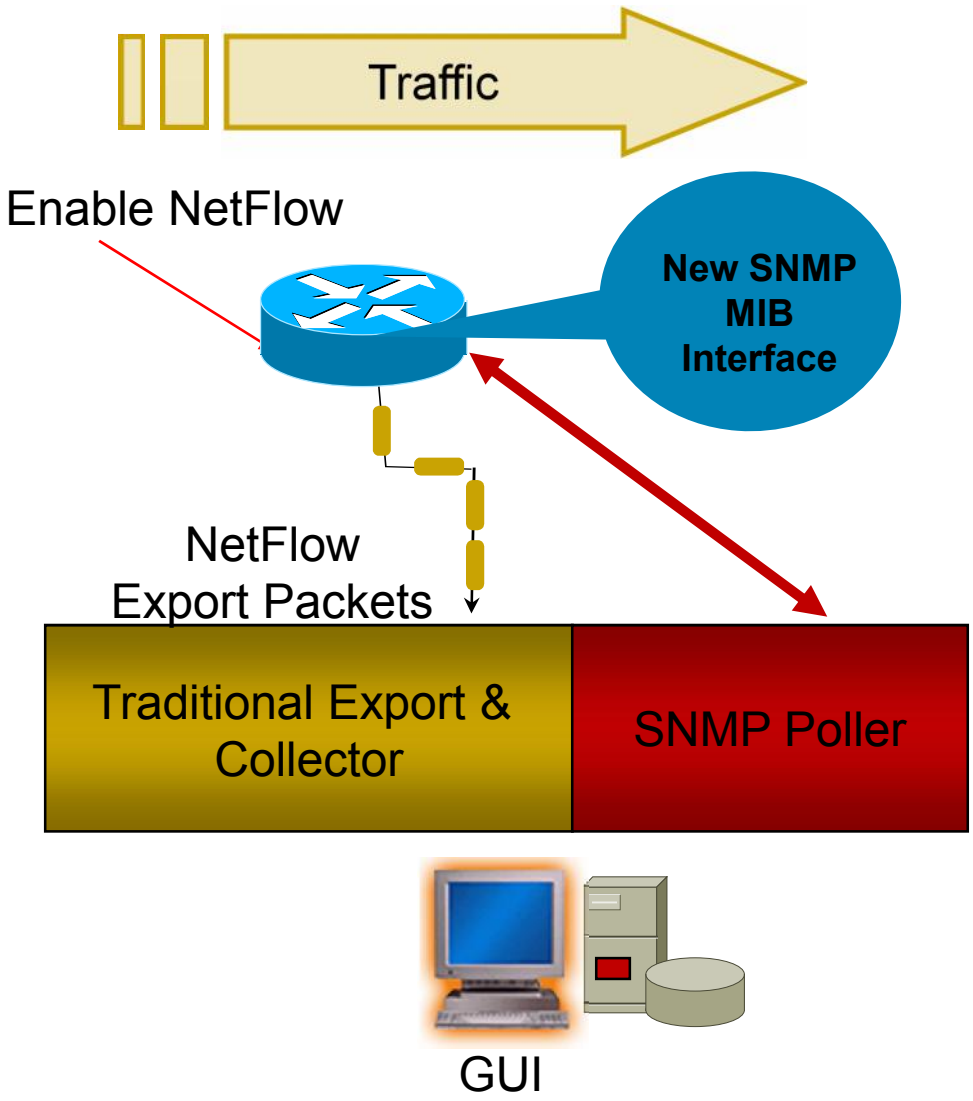
  Agreed to use IPFIX for export protocol

- Information model for packet sampling export

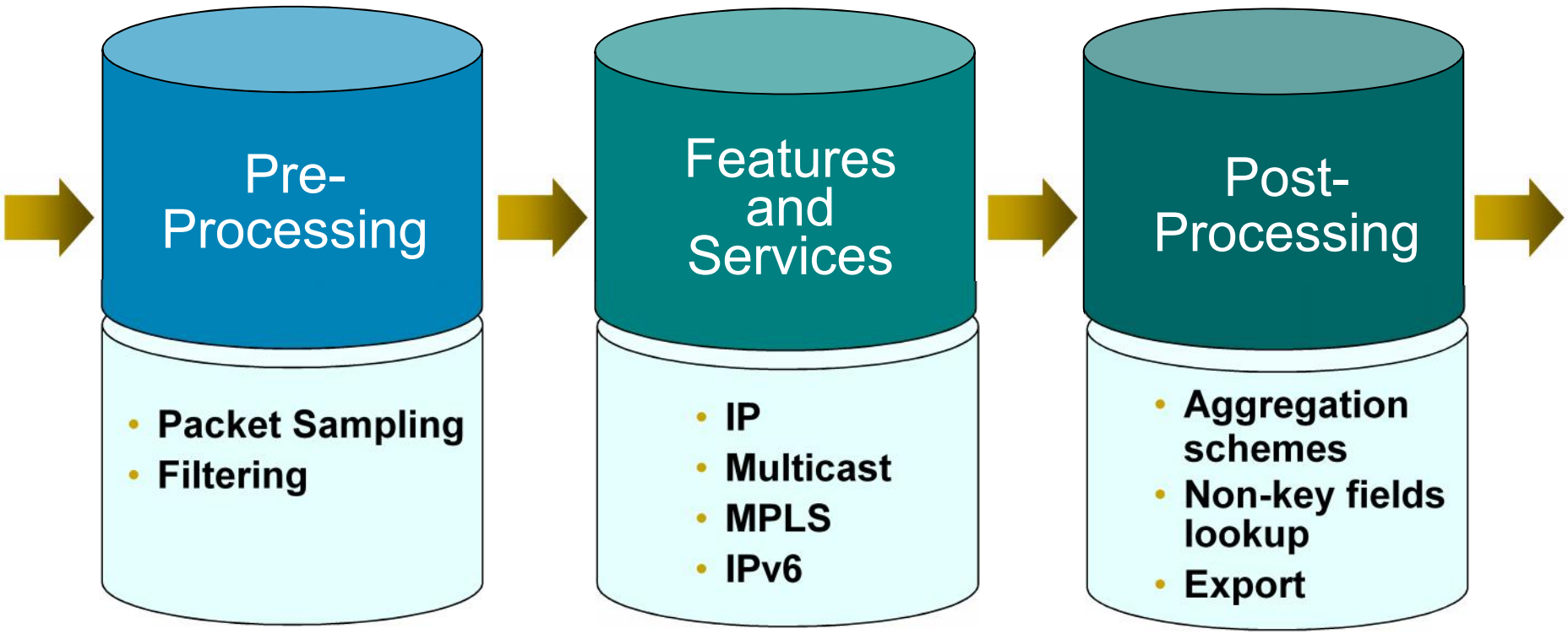
  Extension of the IPFIX information model

In the olden times…
Flow was Defined By Seven Unique Keys

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 3 protocol type
- TOS byte (DSCP)
- Input logical interface (ifIndex)



NetFlow Processing Order



NetFlow Cache Example

Key fields

1. Create and update flows in NetFlow cache

SrcIf	SrcIPadd	DstIf	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	00A 2	/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	00A 1	/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

SrcIf	SrcIPadd	DstIf	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

No

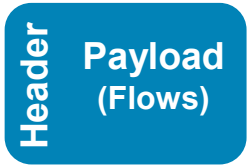
Yes

4. Export version

Non-aggregated flows—export version 5 or 9

5. Transport protocol (UDP, SCTP)

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

Extensibility and Flexibility Requirements Phases Approach

- Traditional NetFlow with the v5 or v8 NetFlow export
  - Really needed something flexible and extensible
- Phase One: NetFlow Version 9
  - Advantages: extensibility
    - Integrate new technologies/data types quicker (MPLS, IPv6, BGP next hop, etc.)
    - Integrate new aggregations quicker

Exporting Process

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

Metering Process

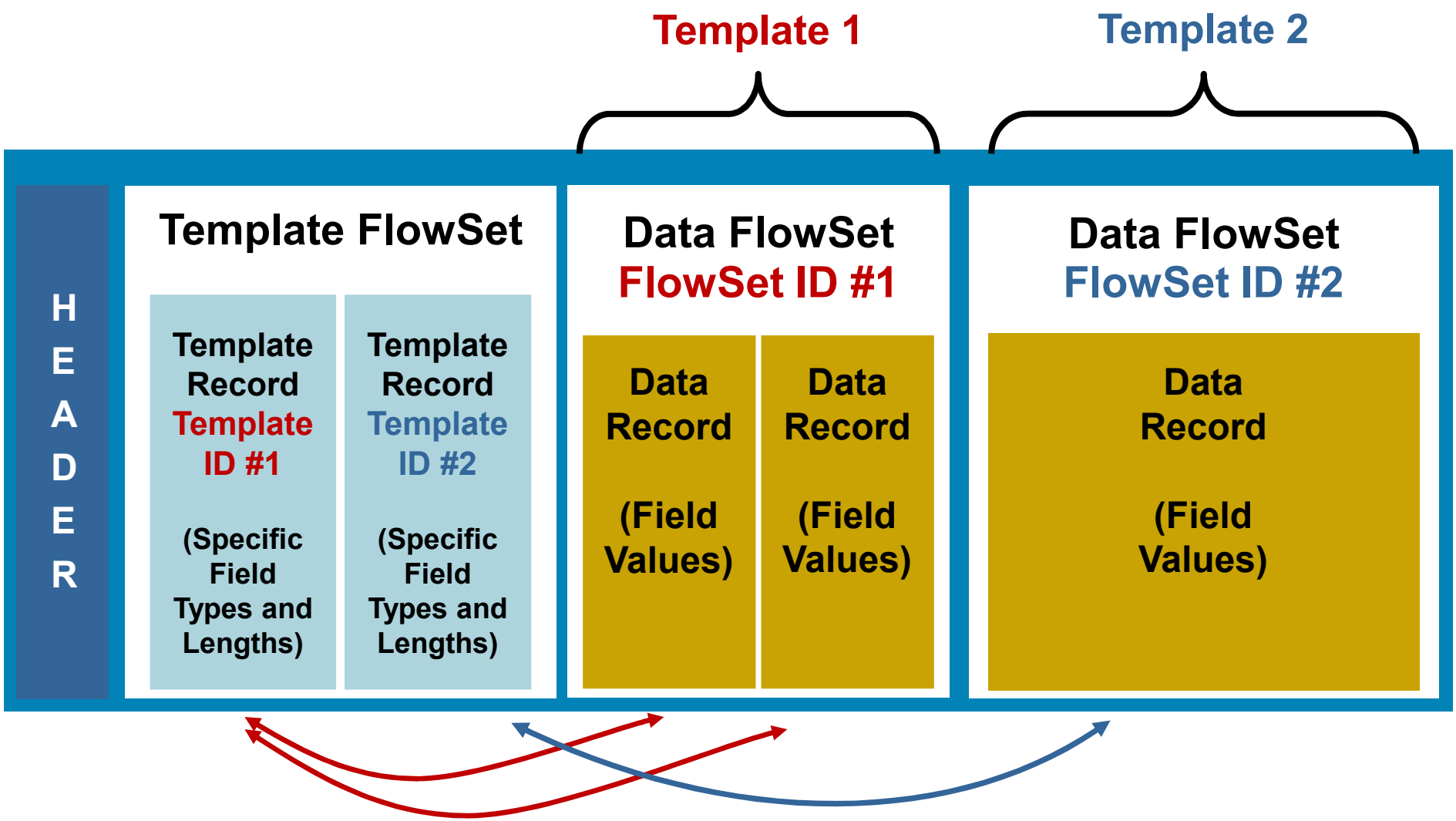
NetFlow Open Source Tools

Product Name	Primary Use	Comment	OS
Cflowd	Traffic Analysis	No longer supported v5, v8	UNIX
Flow-tools	Collector Device	v5, v8, v9 (only old fields)	UNIX
Flowd	Collector Device	V5, v7, and v9	BSD, Linux
FlowScan	Reporting for Flow-Tools	-	UNIX
IPFlow	Traffic Analysis	Support V9, IPv4, IPv6, MPLS, SCTP, etc..	Linux, FreeBSD, Solaris
NetFlow Guide	Reporting Tools		BSD, Linux
NetFlow Monitor	Traffic Analysis	Supports V9	UNIX
Netmet	Collector Device	v5, support v9	Linux
NTOP	Security Monitoring	v9	UNIX
Stager	Reporting for Flow-Tools		UNIX
Nfdump/nfsen	Traffic Analysis	V5, v7, v9	UNIX

Different costs: implementation and customization

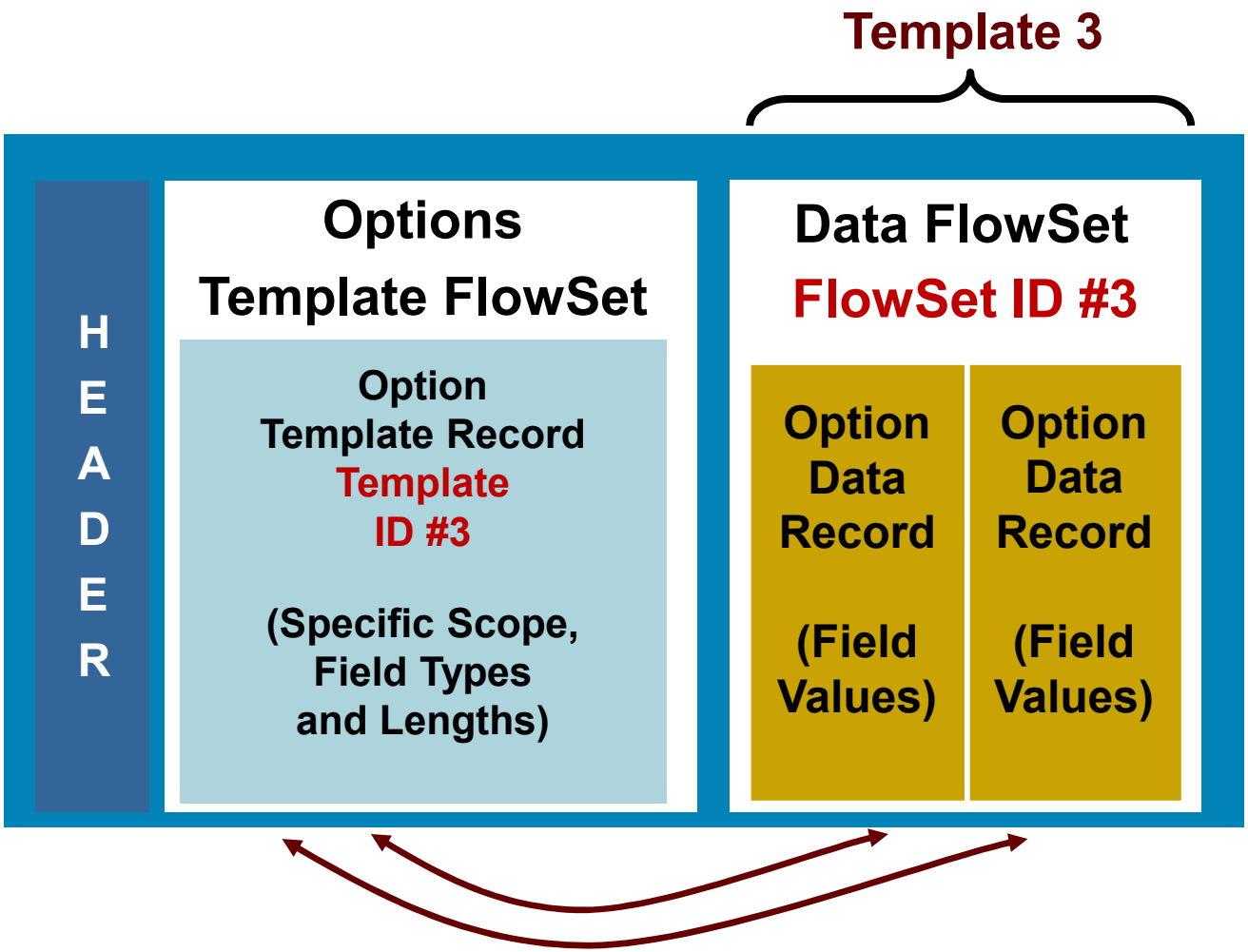
NetFlow v9

NetFlow Version 9 Export Packet



NetFlow Version 9 Export Packet

Options Template FlowSet Specifies the Scope: Cache, System, Template, etc.



Interesting Features on Traditional NetFlow

Multicast & NetFlow

- Multicast NetFlow ingress
  - Sees incoming mcast flow
    - fan out is not represented as there are multiple interfaces
    - Byte counts do not include replication
- Multicast NetFlow egress
  - Sees outgoing multicast packets
    - fan out is represented by multiple cache entries (one per output interface)
- New fields that represent the size of OIL (output interface list)
- Display the multicast data that fails the Reverse Path Forwarding (RPF) check
- No NetFlow export over multicast

IPv6 and NetFlow

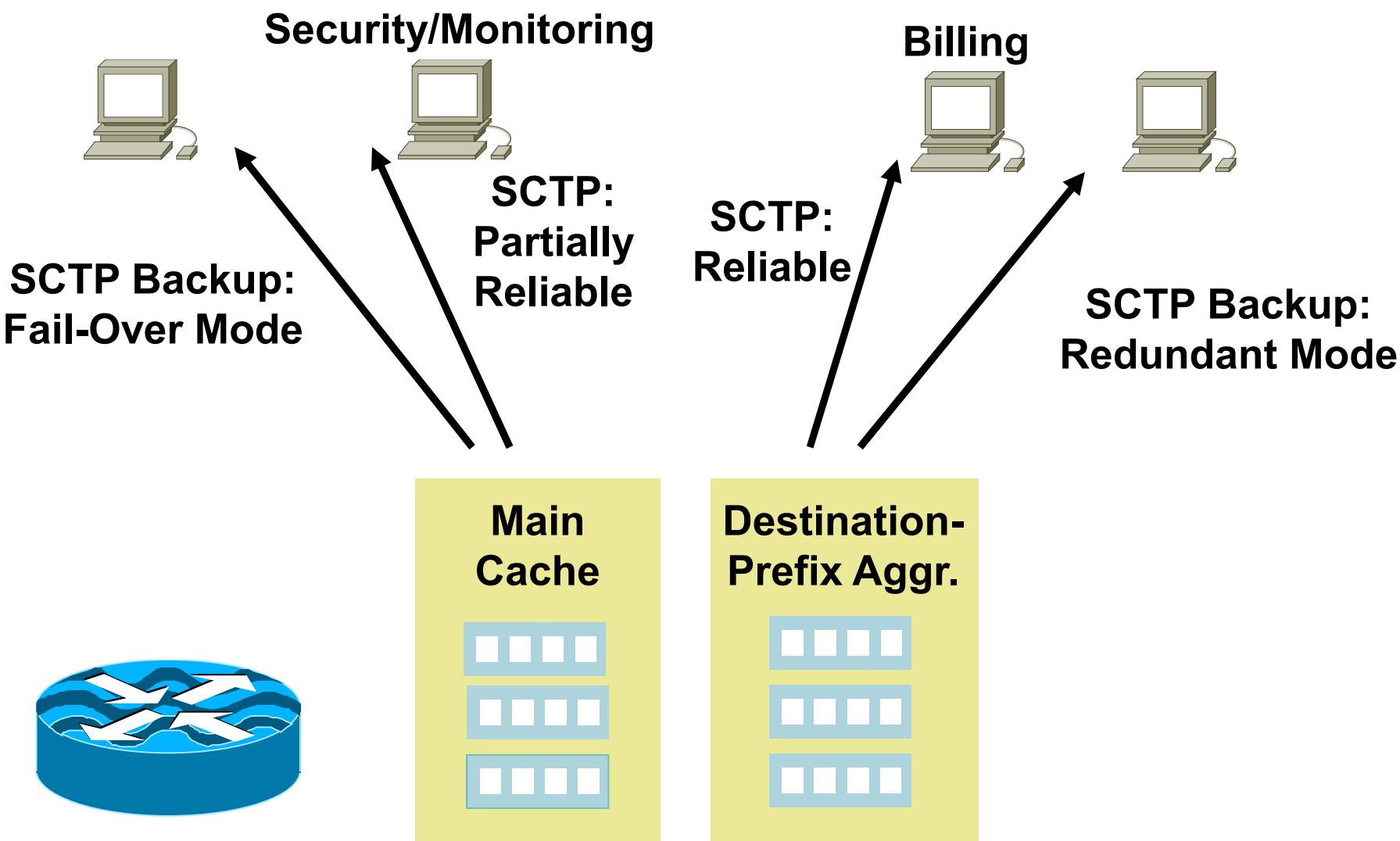
- New NetFlow fields represent IPv6 header fields
  - Needs NFv9 to export
  - Lack of IPv6 capable NetFlow collectors (chicken or egg situation)
    - Currently need it export records about IPv6 via IPv4
- A flow is either IPv4 or IPv6!
  - Separate metering and export for v4 vs. v6, otherwise waste of export bandwidth.

NetFlow Reliable Export with SCTP

- SCTP: stream control transport protocol (RFC4960)
  - Reliable data transfer
  - Congestion control and avoidance
  - Multihoming support
  - One association support for multi-streams
- SCTP-PR: SCTP partially reliable (RFC3578)
  - Three modes of reliability: reliable, partial reliable, unreliable
- Advantages: (Options) templates sent reliably
- Backup Options:
  - Fail-over mode: open the backup connection when the primary fails
  - Redundant mode: open the backup connection in advance, and already send the templates

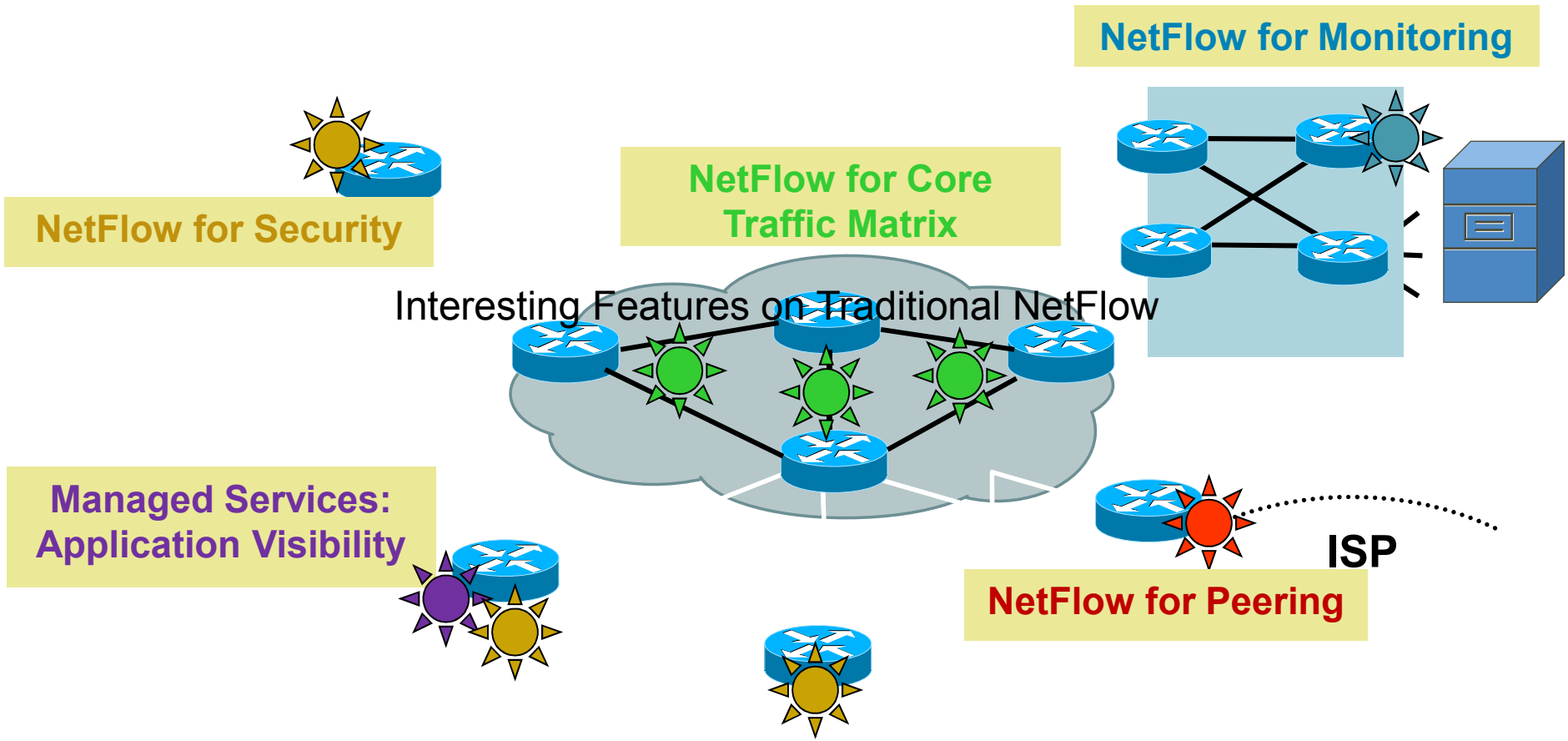
Note: “An Introduction to SCTP”, RFC 3286

NetFlow Reliable Export with SCTP



Flexible NetFlow

Typical NetFlow Deployment



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 in traditional NetFlow

  Different NetFlow caches: per subinterface, per direction (ingress, egress), per sampler, per …

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

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 Oort	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

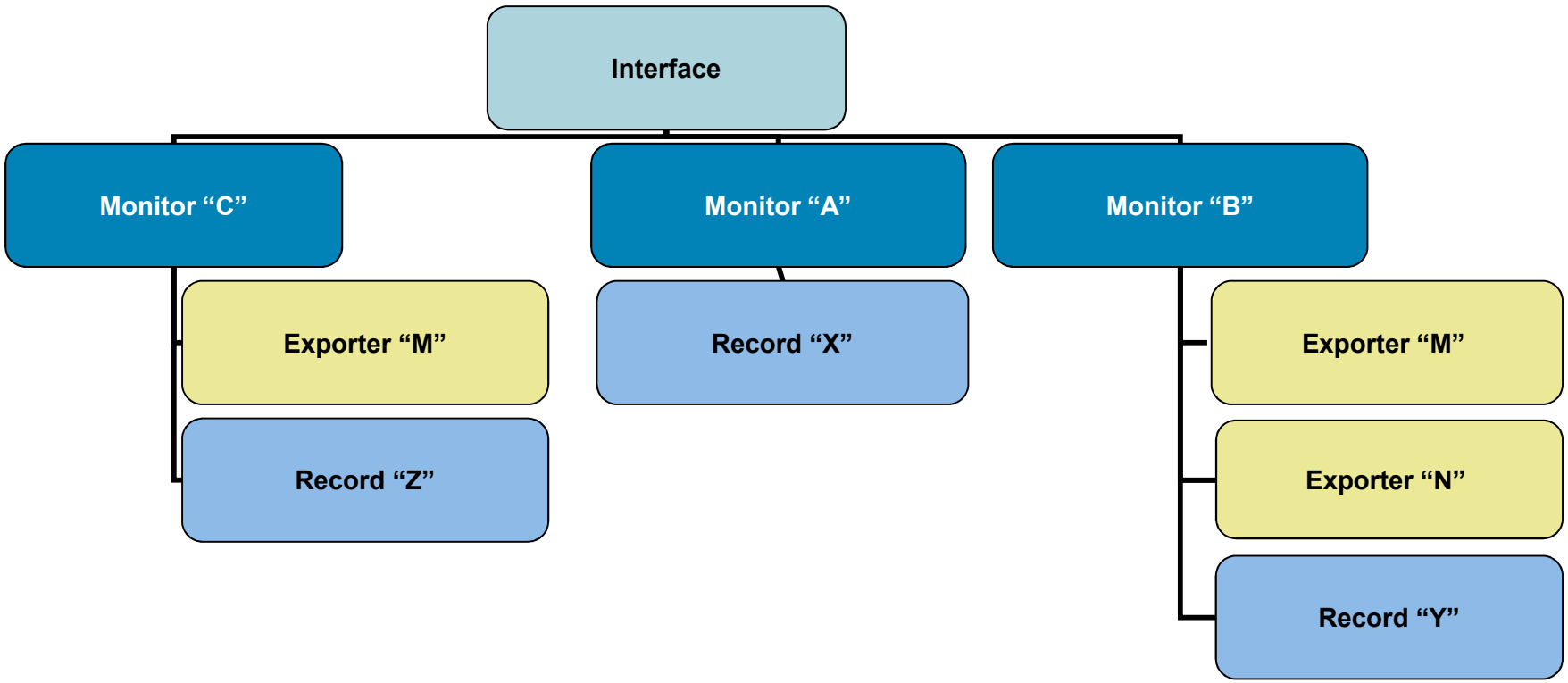
Traffic Analysis Cache

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 Model



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

Flexible Flow Record: Key Fields

Flow	
Sampler ID	Payload Size
Direction	
Prefix (Source or	Packet Section (Header)
Interface	
Input	Packet Section (Payload)
Output	
Minimum-Mask (Source or Destination)	
Protocol	ions
	nap
Fragment Flags	sion
Fragment Offset	cedence
Identification	CP
Header Length	S
Total Length	

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

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	

Flexible Flow Record: Key Fields

NEW

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

NEW

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	

Application
Application ID*

Multicast
Replication Factor*
RPF Check Drop*
Is-Multicast

***: IPv4 Flow only**

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

IPv4 and IPv6
Total Length Minimum (**)
Total Length Maximum (**)

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

(*) IPV4_TOTAL_LEN_MIN, IPV4_TOTAL_LEN_MAX
(**)IP_LENGTH_TOTAL_MIN, IP_LENGTH_TOTAL_MAX

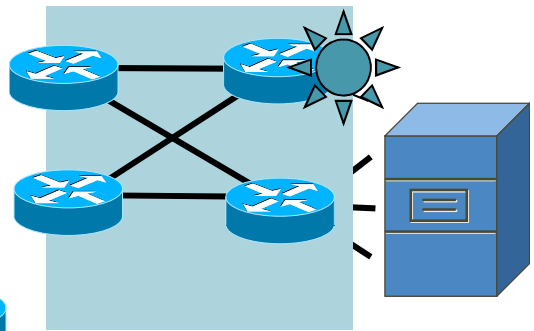
Three Types of NetFlow Caches

- Normal cache (traditional 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

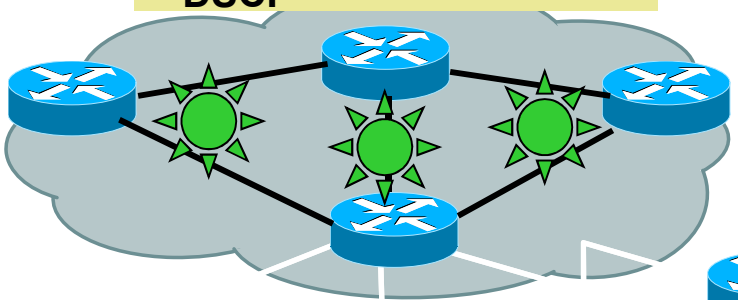
NetFlow Deployment Scenarios

NetFlow for Core Traffic Matrix
- Source/destination AS
- IP addresses (src/dest)
- BGP next hop
- Protocols
- DSCP

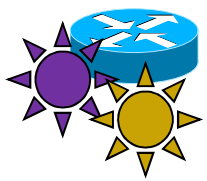
Server Flow Monitor
- Standard seven keys



Security Flow Monitor
- Protocol
- Ports
- IP addresses
- TCP flags



Managed Service Application Visibility
- IP addresses
- Application
- DSCP

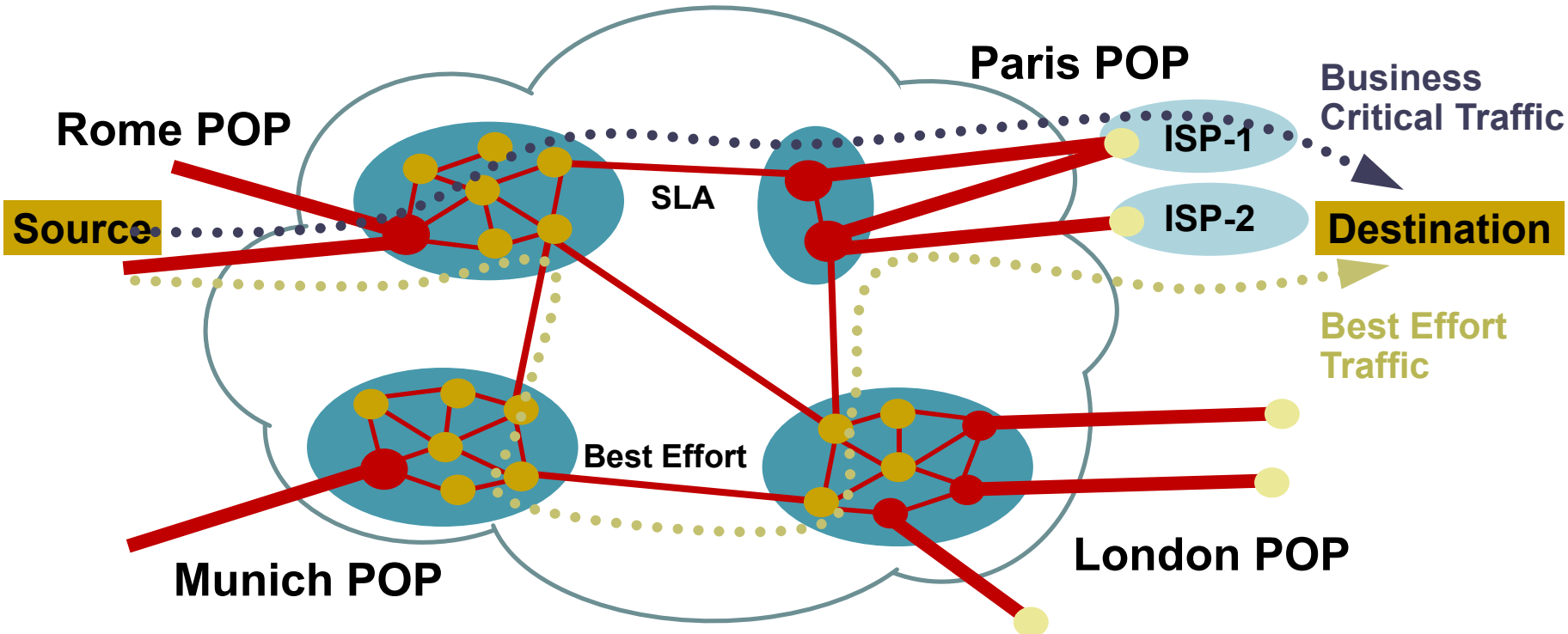


ISP

Peering Flow Monitor
- Destination AS
- Source traffic index
- BGP next hop
- DSCP

NetFlow and Capacity Planning

The Core Traffic Matrix Traffic Engineering and Capacity Planning



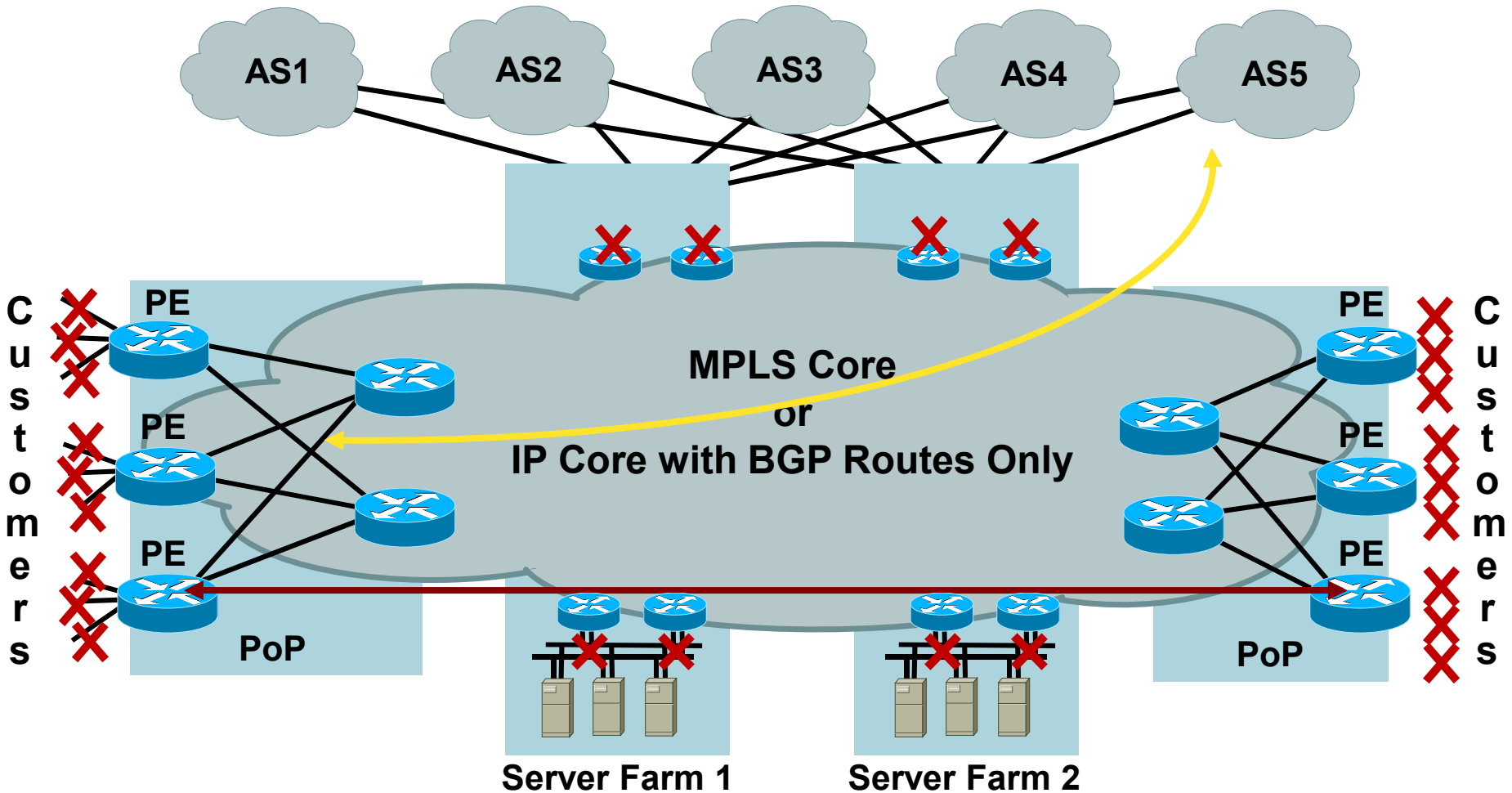
	Rome Exit Point	Paris Exit Point	London Exit Point	Munich Exit Point
Rome Entry Point	NA (*)	…Mb/s	…Mb/s	…Mb/s
Paris Entry Point	…Mb/s	NA (*)	…Mb/s	…Mb/s
London Exit Point	…Mb/s	…Mb/s	NA (*)	…Mb/s
Munich Exit Point	…Mb/s	…Mb/s	…Mb/s	NA (*)

(*) 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

BGP Next Hop TOS Aggregation Typical Example



Internal Traffic: PE to PE

External Traffic Matrix PE to BGP AS

NetFlow BGP Next Hop TOS Aggregation Flow Keys

Key Fields (Uniquely Identifies the Flow)	Additional Export Fields
▪ Origin AS	▪ Flows
▪ Destination AS	▪ Packets
▪ Inbound Interface	▪ Bytes
▪ Output Interface	▪ First SysUptime
▪ ToS/DSCP (*)	▪ Last SysUptime
▪ BGP Next Hop	

(*) Before Any Recoloring

Core Traffic Matrix with Flexible NetFlow

Key Fields (Uniquely Identifies the Flow)	Additional Export Fields
~~Origin AS~~	~~Flows~~
Destination AS	~~Packets~~
Inbound Interface	Bytes
~~Output Interface~~	First SysUptime
ToS/DSCP (*)	Last SysUptime
BGP Next Hop	

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

(*) Before Any Recoloring

NetFlow and Security Analysis

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, one 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, Lancope, Arbor

Flexible Flow Record: Key Fields

Flow	
Sampler ID	Payload Size
Direction	
Prefix (Source or	Packet Section (Header)
Interface	
Input	Packet Section (Payload)
Output	
Minimum-Mask (Source or Dest	
Layer 2	
Source VLAN	
Proto	ions
Destination VLAN	nap
Frag Flags	sion
Source MAC address	
Frag Offse	cedence
Ident	CP
Destination MAC address	S
Head	
Total	

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	

Flexible Flow Record: Key Fields

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

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	

Application
Application ID*

Multicast
Replication Factor*
RPF Check Drop*
Is-Multicast

***: IPv4 Flow only**

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

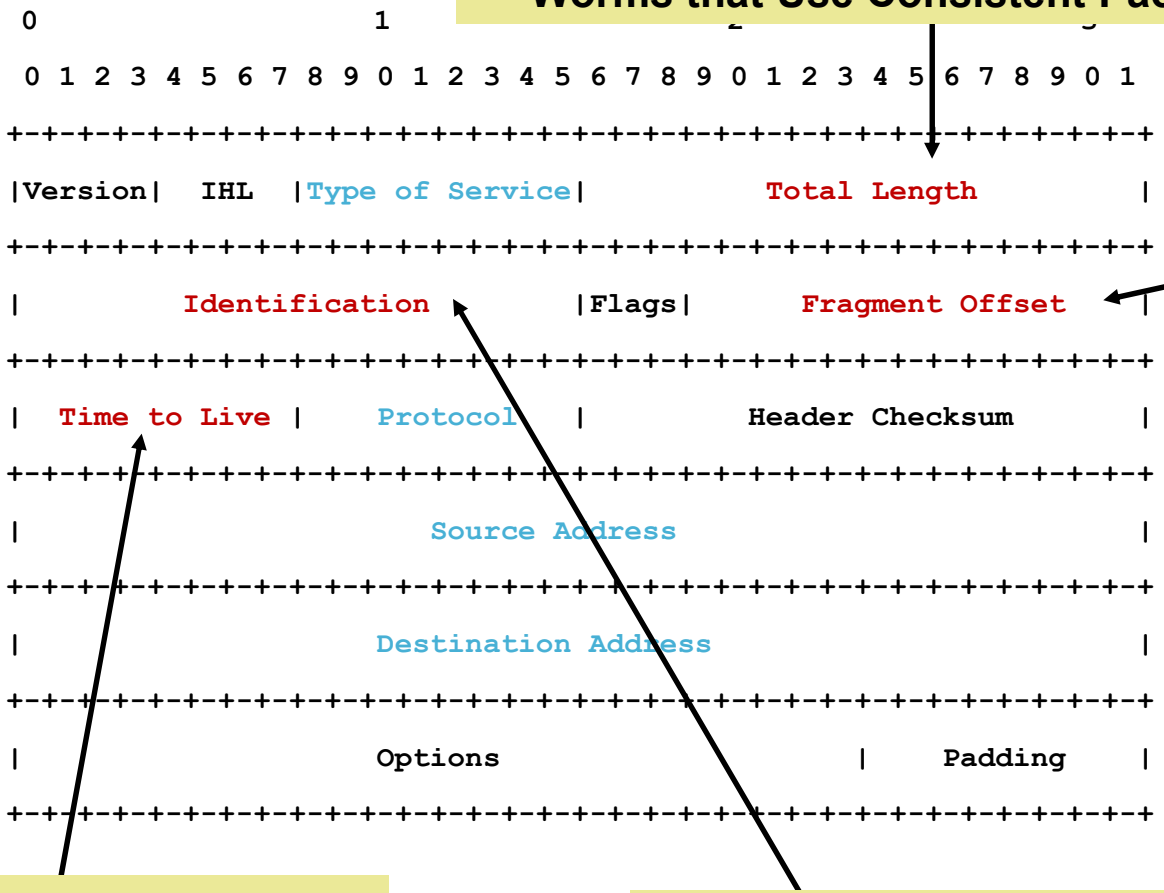
IPv4 and IPv6
Total Length Minimum (**)
Total Length Maximum (**)

(*) IPV4_TOTAL_LEN_MIN, IPV4_TOTAL_LEN_MAX
(**)IP_LENGTH_TOTAL_MIN, IP_LENGTH_TOTAL_MAX

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

Useful Fields for Security Monitoring

Attacks that Use Consistent Packet Size or Worms that Use Consistent Packet Size

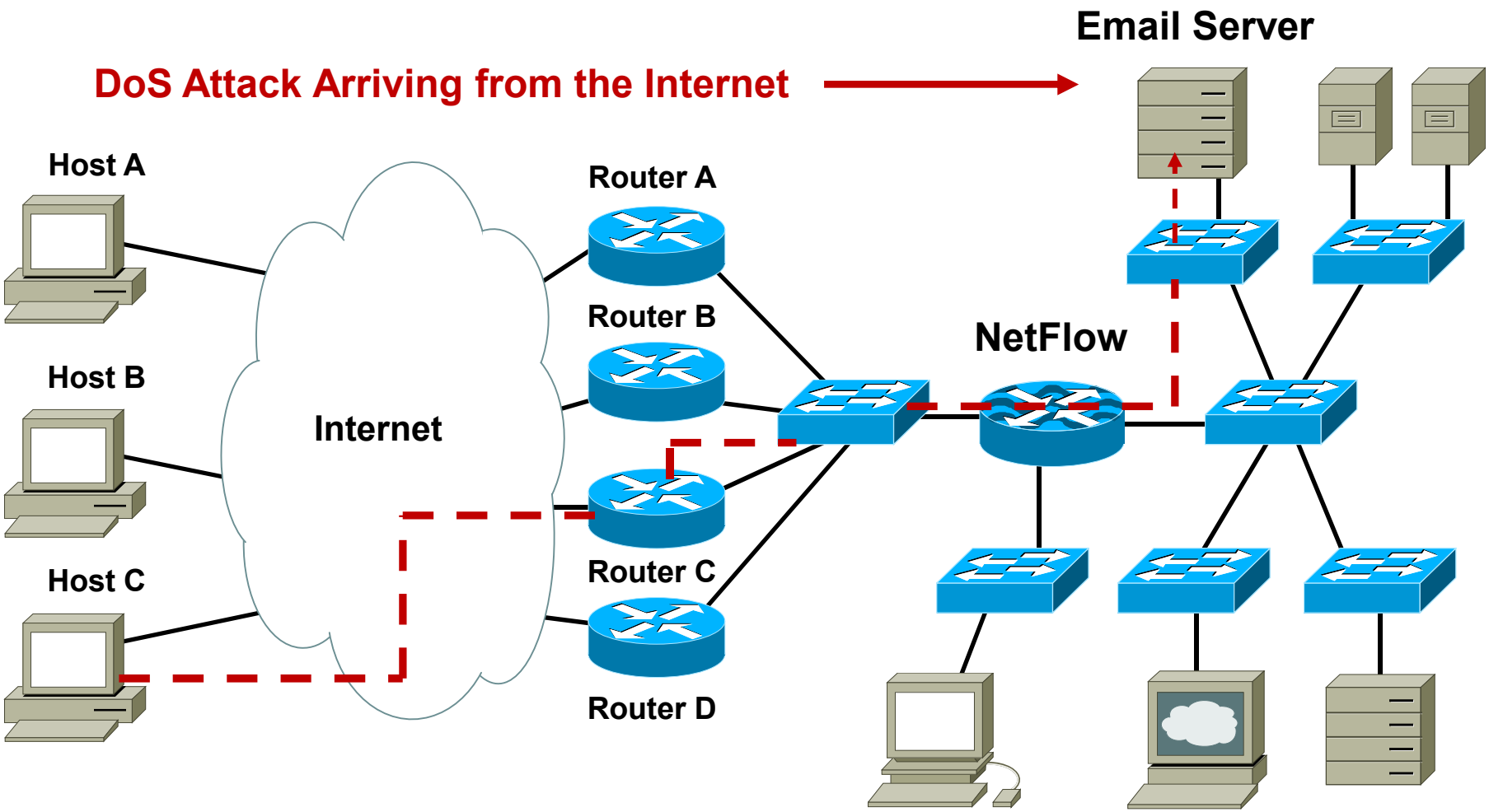


Several Flows with the Same Fragment Offset: Same Packet Sent Over and Over

Flow Issued From the Same Origin

Very Large Packets or Attacks that Might Always Have The Same Generated Identification

Source MAC Address Example



Report the MAC Address for Ethernet, FastEthernet, and GigabitEthernet

The Forwarding Status Field

- What did the router do with the packet?
- Why did it drop it?

Unknown (00b)

Forwarded (01b)

Dropped (10b) → ACL, QoS

Consumed (11b) → Destined to the router
(ex: management traffic)

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

NetFlow L2 and Security Monitoring (for Traditional NetFlow)

- 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
- For IPv4 and IPv6

Embedded Applications of NetFlow
NetFlow Top Talkers

- The flows that are generating the heaviest traffic in the cache are known as the top talkers; prefer top flows
- 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

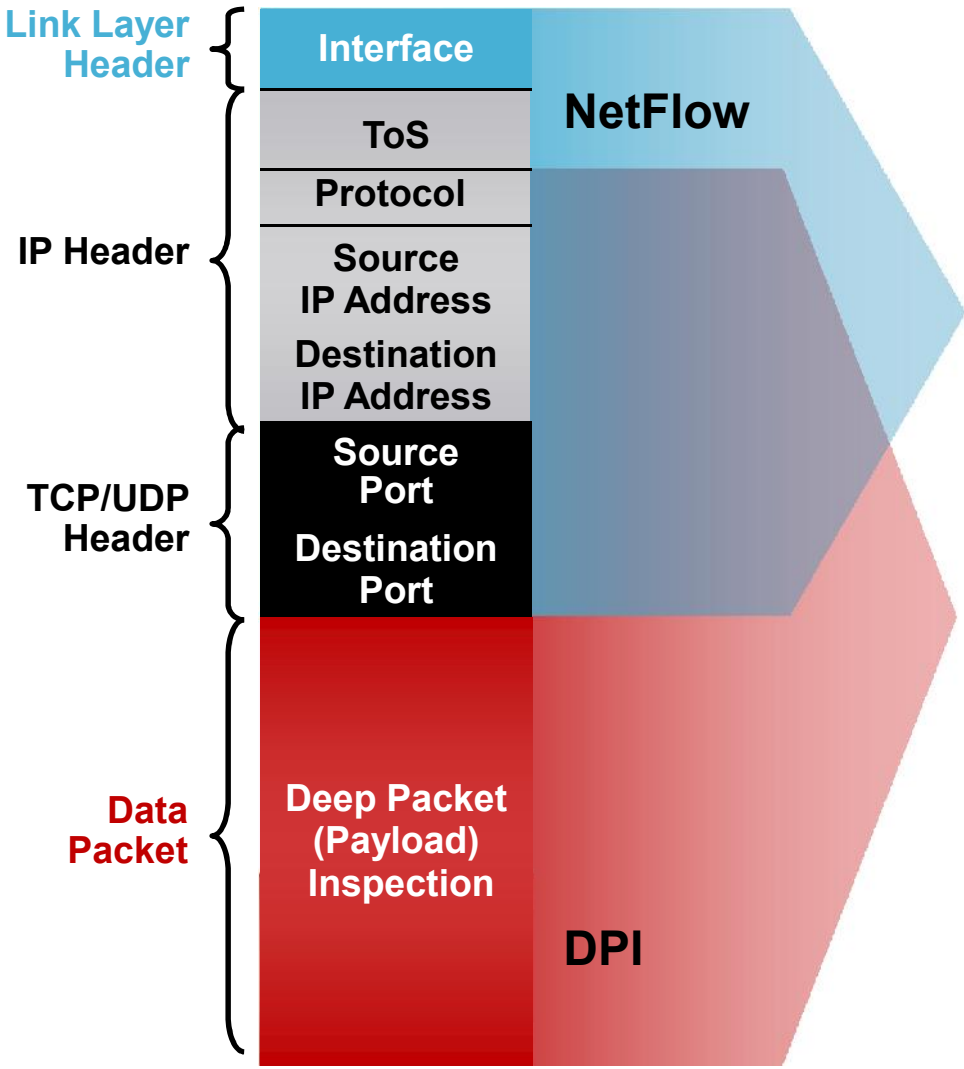
NetFlow beyond routers and Switches
Firewall Application

- Firewalls process large number of ‘transactions’
- Need for logging transactions and stitching on ‘inside’ and ‘outside’ to counter anonymization of flows.
- Traditionally handled via syslog
  - Data to text, text needs to be parsed, back to structured data
- Flow event information can now be exported through NetFlow v9
  - Information about NAT modifications to the traffic
  - Information about Flows denied by security policy
  - Information about AAA/usernames associated with flows
  - bidirectional flows
- Provides scalable logging
  - 10-Gbps flows, 100-k connections per second = lots of logs

NetFlow and Application Visibility

Network Based Application Recognition

What's running on my network?



NetFlow

- ✓ Monitors data in Layers 2 thru 4
- ✓ Determines applications by port
- ✓ Utilizes a seven-tuple for flow
- ✓ Flow information who, what, when, where

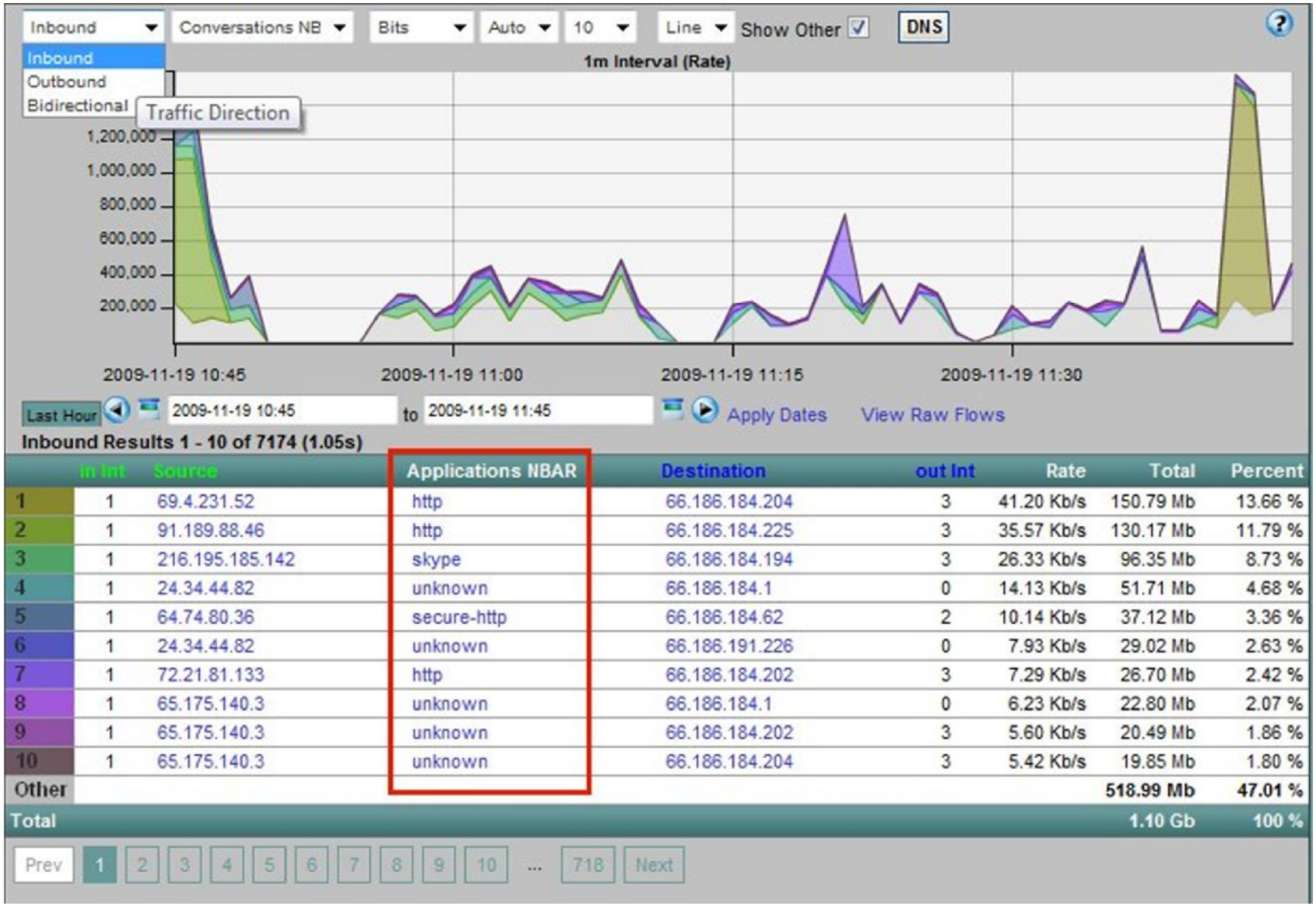
DPI

- ✓ Examines data from Layers 3 thru 7
- ✓ Utilizes Layers 3 and 4 plus packet inspection for classification
- ✓ Stateful inspection of dynamic-port traffic
- ✓ Packet and byte counts

NetFlow and DPI Integration

- NetFlow is the de-facto mechanism to provide visibility on network utilization—who/what/where/when
- Applications can no longer be identified by just L3/L4 information
  - Application visibility is a **must**
  - Example: port 80 is overloaded
- Deep packet inspection boxes to identify applications a cottage industry
- With NetFlow + DPI integration provides single report mapping L2-L7 information

Reporting Example (Plixer)



	in Int	Source	Applications NBAR	Destination	out Int	Rate	Total	Percent
1	1	69.4.231.52	http	66.186.184.204	3	41.20 Kb/s	150.79 Mb	13.66 %
2	1	91.189.88.46	http	66.186.184.225	3	35.57 Kb/s	130.17 Mb	11.79 %
3	1	216.195.185.142	skype	66.186.184.194	3	26.33 Kb/s	96.35 Mb	8.73 %
4	1	24.34.44.82	unknown	66.186.184.1	0	14.13 Kb/s	51.71 Mb	4.68 %
5	1	64.74.80.36	secure-http	66.186.184.62	2	10.14 Kb/s	37.12 Mb	3.36 %
6	1	24.34.44.82	unknown	66.186.191.226	0	7.93 Kb/s	29.02 Mb	2.63 %
7	1	72.21.81.133	http	66.186.184.202	3	7.29 Kb/s	26.70 Mb	2.42 %
8	1	65.175.140.3	unknown	66.186.184.1	0	6.23 Kb/s	22.80 Mb	2.07 %
9	1	65.175.140.3	unknown	66.186.184.202	3	5.60 Kb/s	20.49 Mb	1.86 %
10	1	65.175.140.3	unknown	66.186.184.204	3	5.42 Kb/s	19.85 Mb	1.80 %
Other							518.99 Mb	47.01 %
Total							1.10 Gb	100 %

NetFlow and Performance Measurement

- RTP (voice/video) and TCP user flows analyzed on routers to report:
  - RTP: per packet loss, loss bursts, jitter
  - TCP: loss bursts, round trip time
- Back to benefits of *Flow:
  - Summary reports (*Flow is not a packet capture)
  - Follows topology
- Integrated performance measurements provide easy validation of ‘network signal’, accelerated fault isolation

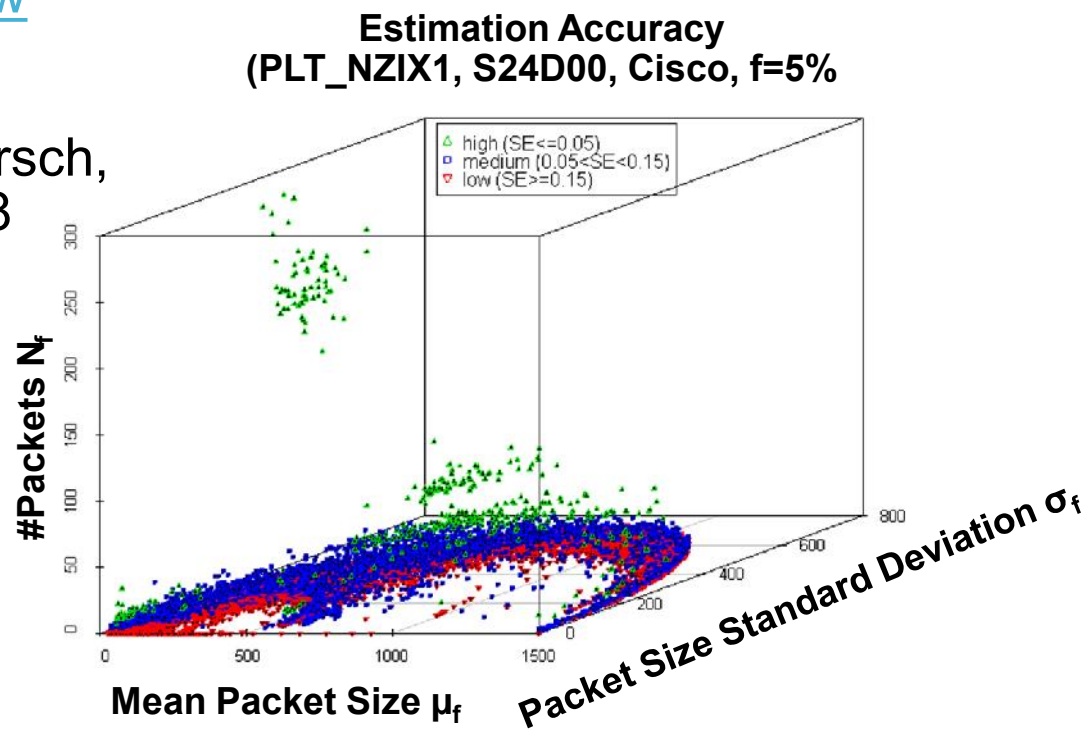
The Cost of NetFlow

Local box measurement impact

- How is the measurement done (ASIC, CPU, etc)
  - Might have TCAM impact (number of flow entries)
  - Might have CPU impact (reducing forwarding performance)
- Does every packet need to be measured?
  - Various sampling methodologies (random time, random packet, 1 in X, etc)
  - Drastic reduction in measurement hit
- Where is the export done?
  - Distributed across line cards vs. centralized
- Considerations and tests laid out in:
  - draft-novak-bmwg-ipflow-meth-*

Accuracy Impact
Random Packet NetFlow Sampling

- Packet Sampling for Flow Accounting: Challenges and Limitations, Tanja Zseby, Thomas Hirsch, Benoit Claise, PAM 2008
- Square sum of bytes available in flexible NetFlow



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

NetFlow Summary and Conclusion

- NetFlow is a mature feature (in Cisco IOS since 1996)
- NetFlow provides input for accounting, performance, security, and billing applications
- NetFlow has IETF and industry leadership
- NetFlow v9 eases the exporting of additional fields
- Flexible NetFlow is a major enhancement
- A lot of features have been added

  Stay tuned for more
- NetFlow export will become **THE** push mechanism ☺

References

- IPFIX
  - http://datatracker.ietf.org/wg/ipfix/charter/
- NetFlow analysis tools
  - http://bit.ly/netflow-freeware (cisco)
  - http://www.switch.ch/network/projects/completed/TF-NGN/floma/software.html
- Linux NetFlow reports HOWTO
  - http://www.linuxgeek.org/NetFlow-howto.php
- Arbor Networks Peakflow SP and Peakflow/X
  - http://www.arbornetworks.com
- nfdump and nfsen
  - http://nfdump.sourceforge.net
  - http://nfsen.sourceforge.net
- Stager
  - http://software.uninett.no/stager/
- 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


CISCO