Net-flow

APRICOT 2008
Network Management
Taipei, Taiwan
February 20-24, 2008

Agenda

- Netflow
  - What it is and how it works
  - Uses and Applications
- Vendor Configurations/ Implementation
  - Cisco and Juniper
- Flow-tools
  - Architectural issues
  - Software, tools etc
- More Discussion / Lab Demonstration

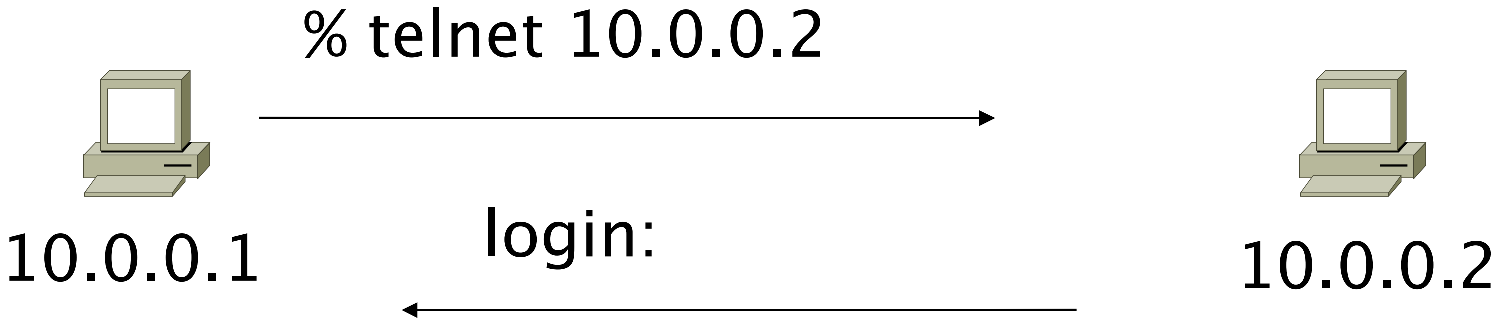
Network Flows

- Packets or frames that have a common attribute.
- Creation and expiration policy – what conditions start and stop a flow.
- Counters – packets,bytes,time.
- Routing information – AS, network mask, interfaces.

Network Flows

- Unidirectional or bidirectional.
- Bidirectional flows can contain other information such as round trip time, TCP behavior.
- Application flows look past the headers to classify packets by their contents.
- Aggregated flows – flows of flows.

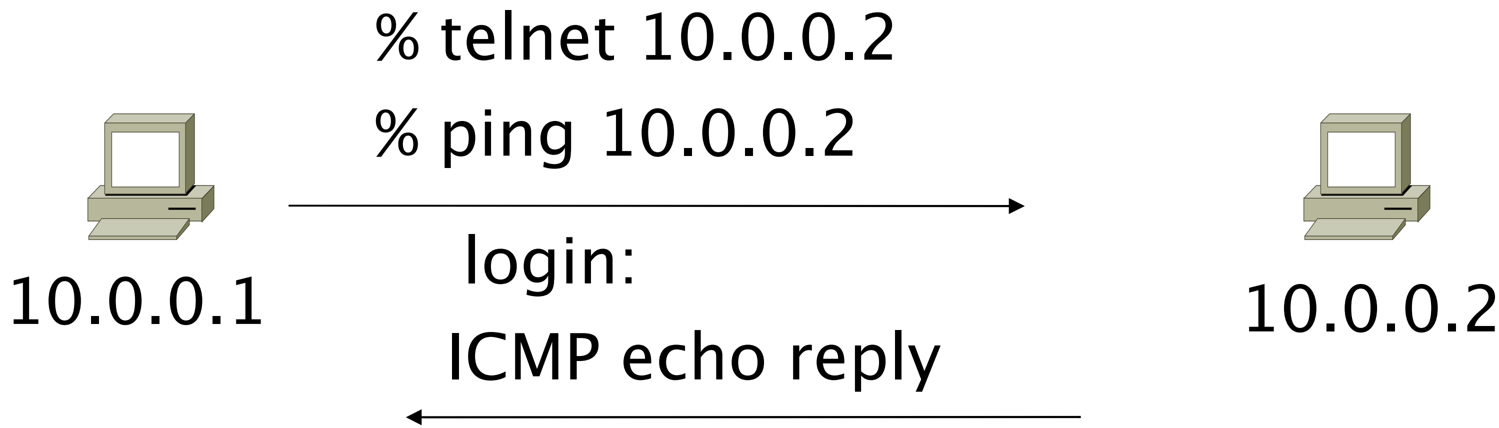
Unidirectional Flow with Source/Destination IP Key



Active Flows

Flow	Source IP	Destination IP
1	10.0.0.1	10.0.0.2
2	10.0.0.2	10.0.0.1

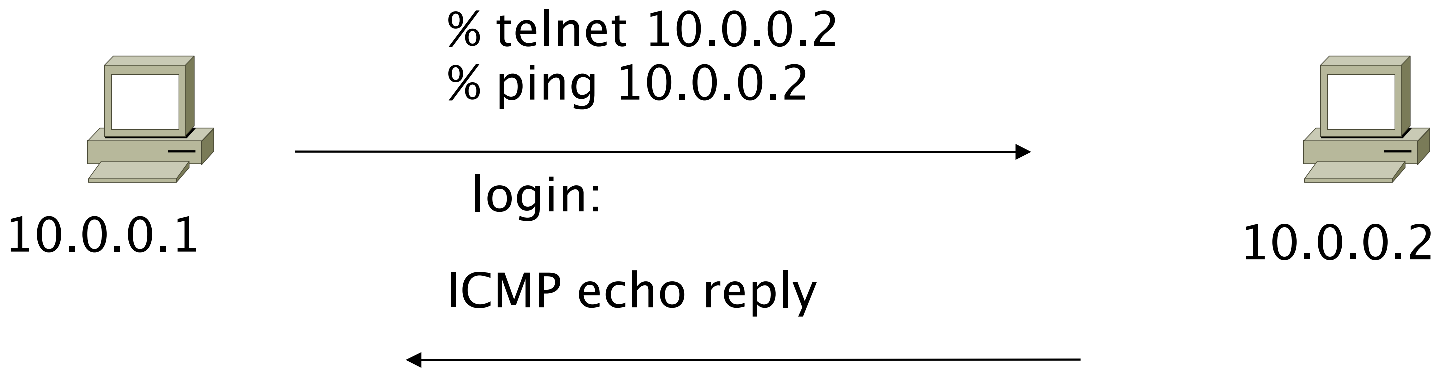
Unidirectional Flow with Source/Destination IP Key



Active Flows

Flow	Source IP	Destination IP
1	10.0.0.1	10.0.0.2
2	10.0.0.2	10.0.0.1

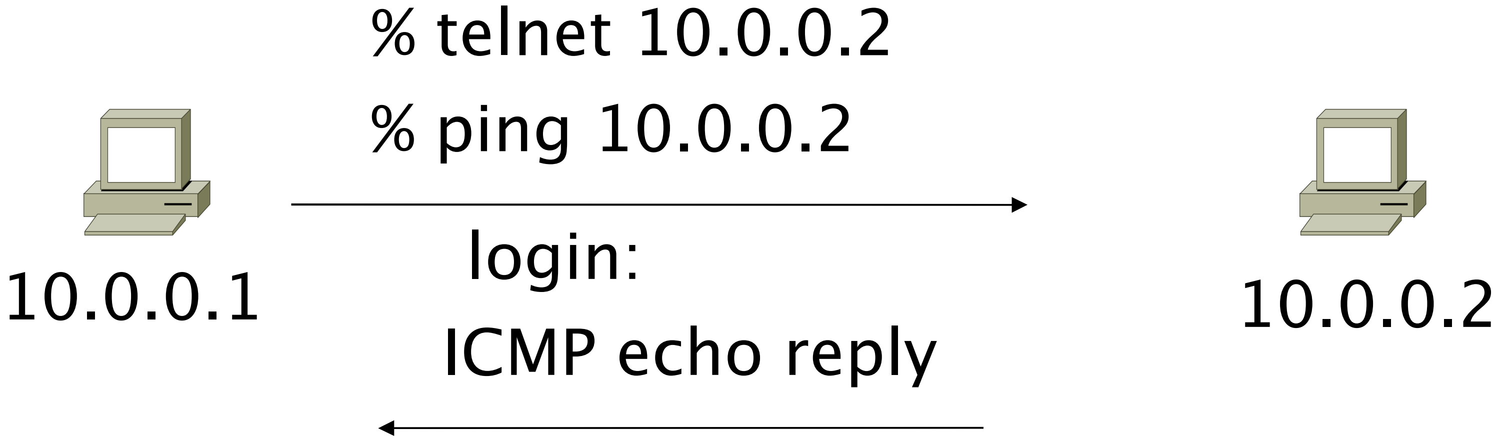
Unidirectional Flow with IP, Port,Protocol Key



Active Flows

Flow	Source IP	Destination IP	prot	srcPort	dstPort
1	10.0.0.1	10.0.0.2	TCP	32000	23
2	10.0.0.2	10.0.0.1	TCP	23	32000
3	10.0.0.1	10.0.0.2	ICMP	0	0
4	10.0.0.2	10.0.0.1	ICMP	0	0

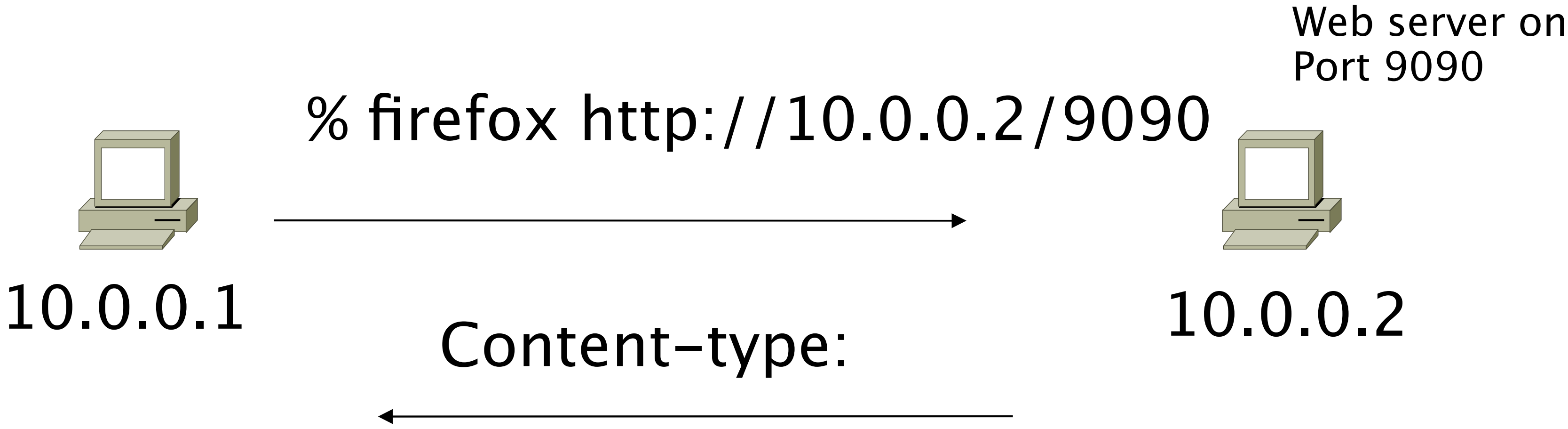
Bidirectional Flow with IP, Port,Protocol Key



Active Flows

Flow	Source IP	Destination IP	prot	srcPort	dstPort
1	10.0.0.1	10.0.0.2	TCP	32000	23
2	10.0.0.1	10.0.0.2	ICMP	0	0

Application Flow



Active Flows

Flow	Source IP	Destination IP	Application
1	10.0.0.1	10.0.0.2	HTTP

Aggregated Flow

Main Active flow table

Flow	Source IP	Destination IP	prot	srcPort	dstPort
1	10.0.0.1	10.0.0.2	TCP	32000	23
2	10.0.0.2	10.0.0.1	TCP	23	32000
3	10.0.0.1	10.0.0.2	ICMP	0	0
4	10.0.0.2	10.0.0.1	ICMP	0	0

Source/Destination IP Aggregate

Flow	Source IP	Destination IP
1	10.0.0.1	10.0.0.2
2	10.0.0.2	10.0.0.1

Working with Flows

- Generating and Viewing Flows
- Exporting Flows from devices
  - Types of flows
  - Sampling rates
- Collecting it
  - Tools to Collect Flows - Flow-tools
- Analyzing it
  - More tools available, can write your own

Flow Descriptors

- A Key with more elements will generate more flows.
- Greater number of flows leads to more post processing time to generate reports, more memory and CPU requirements for device generating flows.
- Depends on application. Traffic engineering vs. intrusion detection.

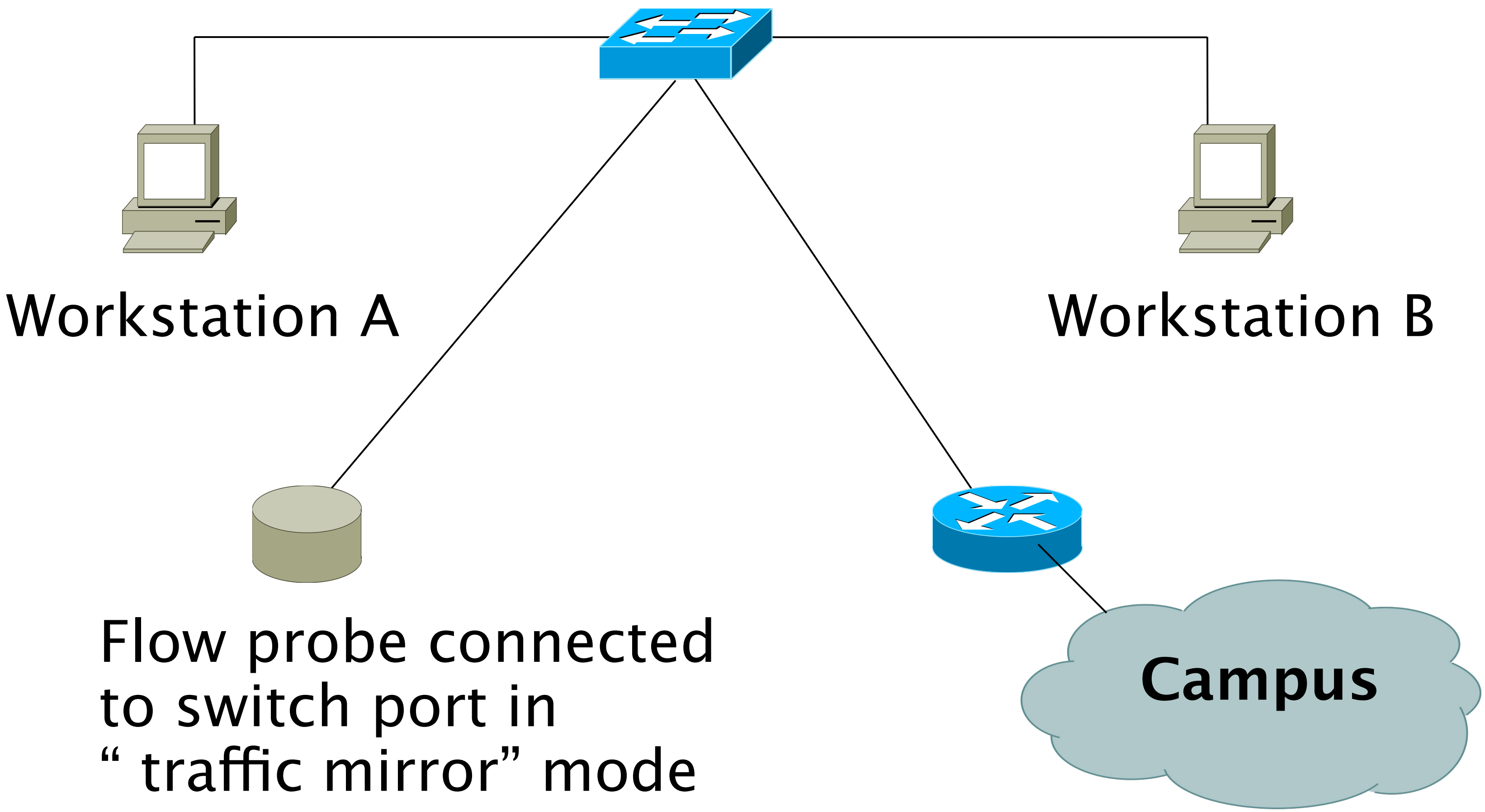
Flow Accounting

- Accounting information accumulated with flows.
- Packets, Bytes, Start Time, End Time.
- Network routing information – masks and autonomous system number.

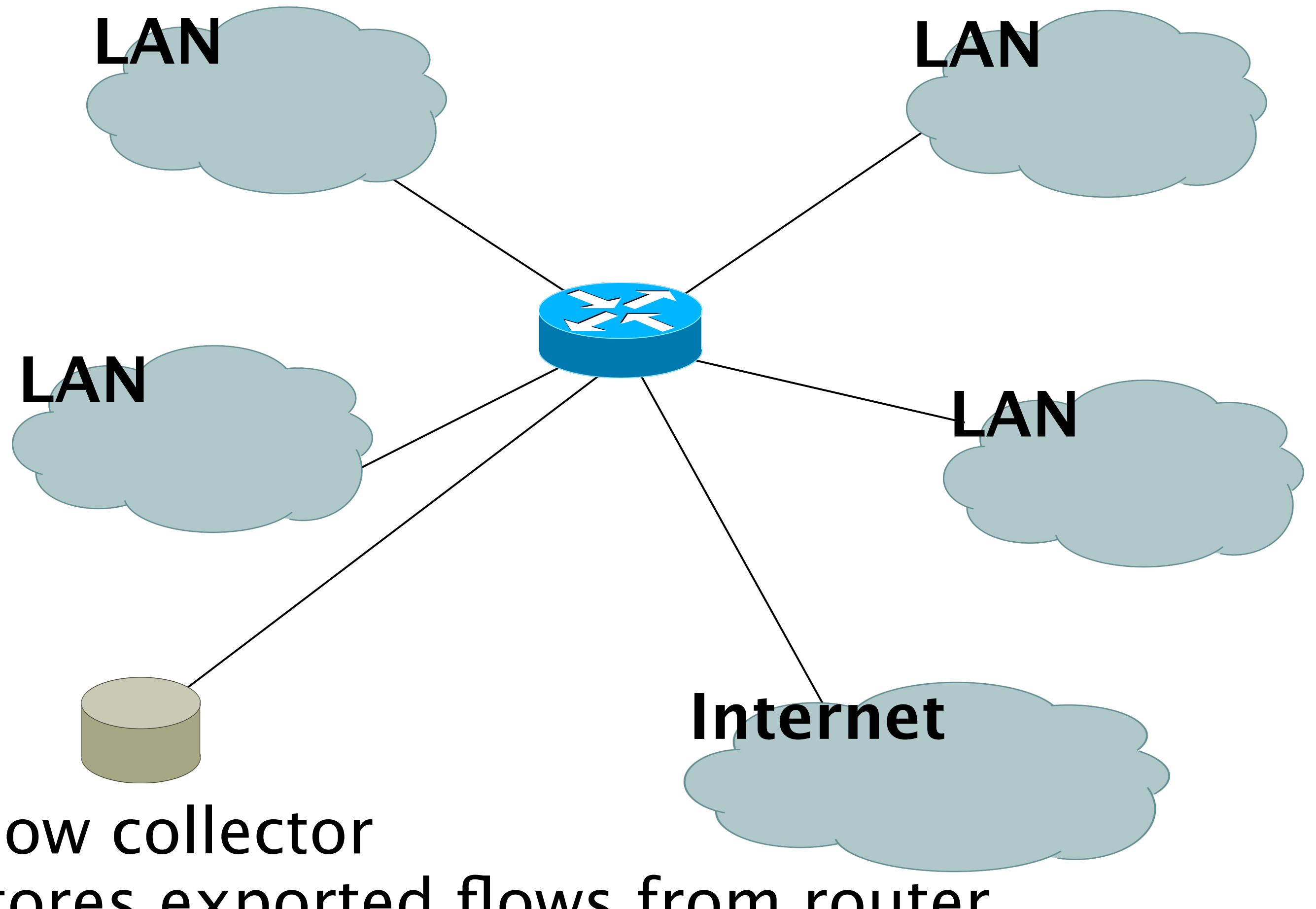
Flow Generation/Collection

- Passive monitor
  - A passive monitor (usually a unix host) receives all data and generates flows.
  - Resource intensive, newer investments needed
- Router or other existing network device.
  - Router or other existing devices like switch, generate flows.
  - Sampling is possible
  - Nothing new needed

Passive Monitor Collection



Router Collection



Flow collector
stores exported flows from router.

Passive Monitor

- Directly connected to a LAN segment via a switch port in “mirror” mode, optical splitter, or repeated segment.
- Generate flows for all local LAN traffic.
- Must have an interface or monitor deployed on each LAN segment.
- Support for more detailed flows – bidirectional and application.

Router Collection

- Router will generate flows for traffic that is directed to the router.
- Flows are not generated for local LAN traffic.
- Limited to “simple” flow criteria (packet headers).
- Generally easier to deploy – no new equipment.

Vendor implementations

Cisco NetFlow

- Unidirectional flows.
- IPv4 unicast and multicast.
- Aggregated and unaggregated.
- Flows exported via UDP.
- Supported on IOS and CatOS platforms.
- Catalyst NetFlow is different implementation.

Cisco NetFlow Versions

- 4 Unaggregated types (1,5,6,7).
- 14 Aggregated types (8.x, 9).
- Each version has its own packet format.
- Version 1 does not have sequence numbers – no way to detect lost flows.
- The “version” defines what type of data is in the flow.
- Some versions specific to Catalyst platform.

NetFlow v1

- Key fields: Source/Destination IP, Source/Destination Port, IP Protocol, ToS, Input interface.
- Accounting: Packets, Octets, Start/End time, Output interface
- Other: Bitwise OR of TCP flags.

NetFlow v5

- Key fields: Source/Destination IP, Source/Destination Port, IP Protocol, ToS, Input interface.
- Accounting: Packets, Octets, Start/End time, Output interface.
- Other: Bitwise OR of TCP flags, Source/Destination AS and IP Mask.
- Packet format adds sequence numbers for detecting lost exports.

NetFlow v8

- Aggregated v5 flows.
- Not all flow types available on all equipments
- Much less data to post process, but loses fine granularity of v5 – no IP addresses.

NetFlow v8

- AS
- Protocol/Port
- Source Prefix
- Destination Prefix
- Prefix
- Destination
- Source/Destination
- Full Flow

NetFlow v8

- ToS/AS
- ToS/Protocol/Port
- ToS/Source Prefix
- ToS/Destination Prefix
- Tos/Source/Destination Prefix
- ToS/Prefix/Port

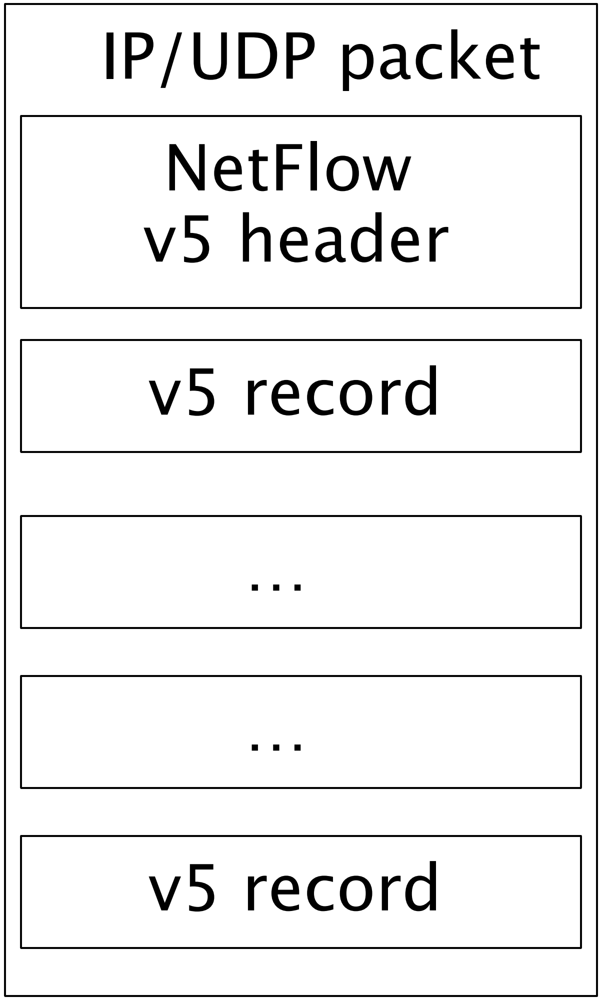
NetFlow v9

- Record formats are defined using templates.
- Template descriptions are communicated from the router to the NetFlow Collection Engine.
- Flow records are sent from the router to the NetFlow Collection Engine with minimal template information so that the NetFlow Collection Engine can relate the records to the appropriate template.
- Version 9 is independent of the underlying transport (UDP, TCP, SCTP, and so on).

NetFlow Packet Format

- Common header among export versions.
- All but v1 have a sequence number.
- Version specific data field where N records of data type are exported.
- N is determined by the size of the flow definition. Packet size is kept under ~1480 bytes. No fragmentation on Ethernet.

NetFlow v5 Packet Example



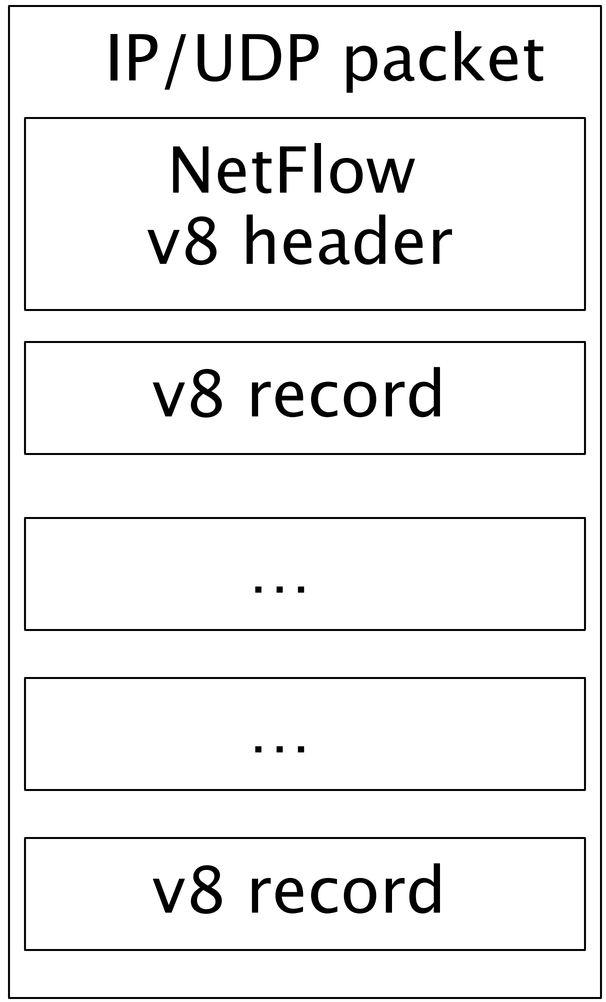
NetFlow v5 Packet (Header)

```
struct ftpdu_v5 {
  /* 24 byte header */
  u_int16 version;       /* 5 */
  u_int16 count;         /* The number of records in the PDU */
  u_int32 sysUpTime;     /* Current time in millisecs since router booted */
  u_int32 unix_secs;     /* Current seconds since 0000 UTC 1970 */
  u_int32 unix_nsecs;    /* Residual nanoseconds since 0000 UTC 1970 */
  u_int32 flow_sequence; /* Seq counter of total flows seen */
  u_int8  engine_type;   /* Type of flow switching engine (RP,VIP,etc.) */
  u_int8  engine_id;     /* Slot number of the flow switching engine */
  u_int16 reserved;
```

NetFlow v5 Packet (Records)

```
 /* 48 byte payload */
  struct ftrec_v5 {
    u_int32 srcaddr;    /* Source IP Address */
    u_int32 dstaddr;    /* Destination IP Address */
    u_int32 nexthop;    /* Next hop router's IP Address */
    u_int16 input;      /* Input interface index */
    u_int16 output;     /* Output interface index */
    u_int32 dPkts;      /* Packets sent in Duration */
    u_int32 dOctets;    /* Octets sent in Duration. */
    u_int32 First;      /* SysUptime at start of flow */
    u_int32 Last;       /* and of last packet of flow */
    u_int16 srcport;    /* TCP/UDP source port number or equivalent */
    u_int16 dstport;    /* TCP/UDP destination port number or equiv */
    u_int8  pad;
    u_int8  tcp_flags;  /* Cumulative OR of tcp flags */
    u_int8  prot;       /* IP protocol, e.g., 6=TCP, 17=UDP, ... */
    u_int8  tos;        /* IP Type-of-Service */
    u_int16 src_as;     /* originating AS of source address */
    u_int16 dst_as;     /* originating AS of destination address */
    u_int8  src_mask;   /* source address prefix mask bits */
    u_int8  dst_mask;   /* destination address prefix mask bits */
    u_int16 drops;
  } records[FT_PDU_V5_MAXFLOWS];
};
```

NetFlow v8 Packet Example (AS Aggregation)



NetFlow v8 AS agg. Packet

```
struct ftpdu_v8_1 {
  /* 28 byte header */
  u_int16 version;       /* 8 */
  u_int16 count;         /* The number of records in the PDU */
  u_int32 sysUpTime;     /* Current time in millisecs since router booted */
  u_int32 unix_secs;     /* Current seconds since 0000 UTC 1970 */
  u_int32 unix_nsecs;    /* Residual nanoseconds since 0000 UTC 1970 */
  u_int32 flow_sequence; /* Seq counter of total flows seen */
  u_int8  engine_type;   /* Type of flow switching engine (RP,VIP,etc.) */
  u_int8  engine_id;     /* Slot number of the flow switching engine */
  u_int8  aggregation;   /* Aggregation method being used */
  u_int8  agg_version;   /* Version of the aggregation export */
  u_int32 reserved;
  /* 28 byte payload */
  struct ftrec_v8_1 {
    u_int32 dFlows;      /* Number of flows */
    u_int32 dPkts;       /* Packets sent in duration */
    u_int32 dOctets;     /* Octets sent in duration */
    u_int32 First;       /* SysUpTime at start of flow */
    u_int32 Last;        /* and of last packet of flow */
    u_int16 src_as;      /* originating AS of source address */
    u_int16 dst_as;      /* originating AS of destination address */
    u_int16 input;       /* input interface index */
    u_int16 output;      /* output interface index */
  } records[FT_PDU_V8_1_MAXFLOWS];
};
```

Cisco IOS Configuration

- Configured on each input interface.
- Define the version.
- Define the IP address of the collector (where to send the flows).
- Optionally enable aggregation tables.
- Optionally configure flow timeout and main (v5) flow table size.
- Optionally configure sample rate.

Cisco IOS Configuration

```
interface FastEthernet0/0
 description Access to backbone
 ip address 169.223.11.194 255.255.252.0
 ip route-cache flow
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description Access to local net
 ip address 169.223.2.1 255.255.255.128
 ip route-cache flow
 duplex auto
 speed auto


ip flow-export version 5
ip flow-export destination 169.223.2.2 5004
```

Cisco IOS Configuration

- Change in command in newer IOS

```
interface FastEthernet0/0
 ip route-cache flow      ! Prior to IOS 12.4
 ip flow [ingress|egress] ! From IOS 12.4
```

- If CEF is not configured on the router, this turns off the existing switching path on the router and enables NetFlow switching (basically modified optimum switching).
- If CEF is configured on the router, NetFlow simply becomes a "flow information gatherer" and feature accelerator—CEF remains operational as the underlying switching process

Cisco IOS Configuration

```
gw-169-223-2-0#sh ip flow export
Flow export v5 is enabled for main cache
  Export source and destination details :
  VRF ID : Default
    Destination(1)  169.223.2.2 (5004)
  Version 5 flow records
  55074 flows exported in 3348 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
```

Cisco IOS Configuration

```
gw-169-223-2-0#sh ip cache flow
IP packet size distribution (3689551 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .483 .189 .014 .002 .003 .001 .000 .000 .000 .000 .000 .000 .000 .001

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .001 .000 .008 .002 .288 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  26 active, 4070 inactive, 55206 added
  1430681 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 25800 bytes
  26 active, 998 inactive, 55154 added, 55154 added to flow
  0 alloc failures, 0 force free
  1 chunk, 2 chunks added
  last clearing of statistics never
```

Cisco IOS Configuration

Protocol --------	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active(Sec) /Flow	Idle(Sec) /Flow
TCP-Telnet	3357	0.0	35	92	1.3	0.5	11.5
TCP-FTP	128	0.0	19	97	0.0	0.6	1.5
TCP-FTPD	128	0.0	105	771	0.1	0.2	1.5
TCP-WWW	13462	0.1	125	962	19.3	7.0	5.9
TCP-X	269	0.0	1	40	0.0	0.0	14.3
TCP-other	9107	0.1	154	62	16.1	6.9	8.2
UDP-DNS	2248	0.0	1	73	0.0	0.8	15.4
UDP-NTP	3132	0.0	1	76	0.0	0.0	15.4
UDP-TFTP	24	0.0	6	49	0.0	30.0	15.3
UDP-Frag	6	0.0	1	32	0.0	0.0	15.5
UDP-other	6700	0.0	9	104	0.7	2.2	15.5
ICMP	16661	0.1	23	87	4.5	18.5	15.4
Total:	55222	0.6	66	480	42.3	8.8	11.6

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	SrcP	DstP	Pkts
Fa0/1	169.223.2.195	Fa0/0	202.128.0.7	01	0000	0800	4
Fa0/1	169.223.2.195	Fa0/0	218.185.127.204	01	0000	0800	4
Fa0/1	169.223.2.2	Fa0/0	169.223.15.102	06	0016	C917	89
Fa0/1	169.223.2.2	Local	169.223.2.1	06	DB27	0016	120
Fa0/1	169.223.2.195	Fa0/0	202.128.31.179	01	0000	0800	4
Fa0/0	208.81.191.133	Fa0/1	169.223.2.194	06	0050	8452	3

Cisco IOS Configuration

```
ip flow-top-talkers
 top 10
 sort-by bytes
```

```
gw-169-223-2-0#sh ip flow top-talkers

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP Bytes
Fa0/1         169.223.2.2     Fa0/0         169.223.11.33   06 0050 0B64  3444K
Fa0/1         169.223.2.2     Fa0/0         169.223.11.33   06 0050 0B12  3181K
Fa0/0         169.223.11.33   Fa0/1         169.223.2.2     06 0B12 0050    56K
Fa0/0         169.223.11.33   Fa0/1         169.223.2.2     06 0B64 0050    55K
Fa0/1         169.223.2.2     Local         169.223.2.1     01 0000 0303    18K
Fa0/1         169.223.2.130   Fa0/0         64.18.197.134   06 9C45 0050    15K
Fa0/1         169.223.2.130   Fa0/0         64.18.197.134   06 9C44 0050    12K
Fa0/0         213.144.138.195 Fa0/1         169.223.2.130   06 01BB DC31  7167
Fa0/0         169.223.15.102  Fa0/1         169.223.2.2     06 C917 0016  2736
Fa0/1         169.223.2.2     Local         169.223.2.1     06 DB27 0016  2304
10 of 10 top talkers shown. 49 flows processed.
```

Cisco command summary

- Enable CEF
  - **`ip cef`**
- Enable flow on each interface

```
ip route cache flow OR
ip flow ingress
ip flow egress
```

- View flows
  - **`show ip cache flow`**
  - **`show ip flow top-talkers`**

Cisco Command Summary

- Exporting Flows to a collector

```
ip flow-export version 5 [origin-as|peer-as]
ip flow-export destination x.x.x.x <udp-port>
```

- Exporting aggregated flows

```
ip flow-aggregation cache as|prefix|dest|source|proto
  enabled
  export destination x.x.x.x <udp-port>
```

Flows and Applications

Uses for Flow

- Problem identification / solving
  - Traffic classification
  - DoS Traceback (some slides by Danny McPherson)
- Traffic Analysis
  - Inter-AS traffic analysis
  - Reporting on application proxies
- Accounting
  - Cross verification from other sources
  - Can cross-check with SNMP data

Traffic Classification

- Based on Protocol, source and destination ports
  - Protocol identification (TCP, UDP, ICMP)
  - Can define well known ports
  - Can identify well known P2P ports
  - Most common use
    - Proxy measurement - http , ftp
    - Rate limiting P2P traffic

Traceback: Flow-based*

- Trace attack by matching fingerprint/signature at each interface via passive monitoring:
  - Flow data (e.g., NetFlow, cflowd, sFlow, IPFIX)
  - Span Data
  - PSAMP (Packet Sampling, IETF PSAMP WG)
- Number of open source and commercial products evolving in market
- Non-intrusive, widely supported

Flow–based Detection*

- Monitor flows (i.e., Network and Transport Layer transactions) on the network and build baselines for what normal behavior looks like:
  - Per interface
  - Per prefix
  - Per Transport Layer protocol & ports
  - Build time-based buckets (e.g., 5 minutes, 30 minutes, 1 hours, 12 hours, day of week, day of month, day of year)

Detect Anomalous Events: SQL "Slammer" Worm*

Flow-based Detection (cont)*

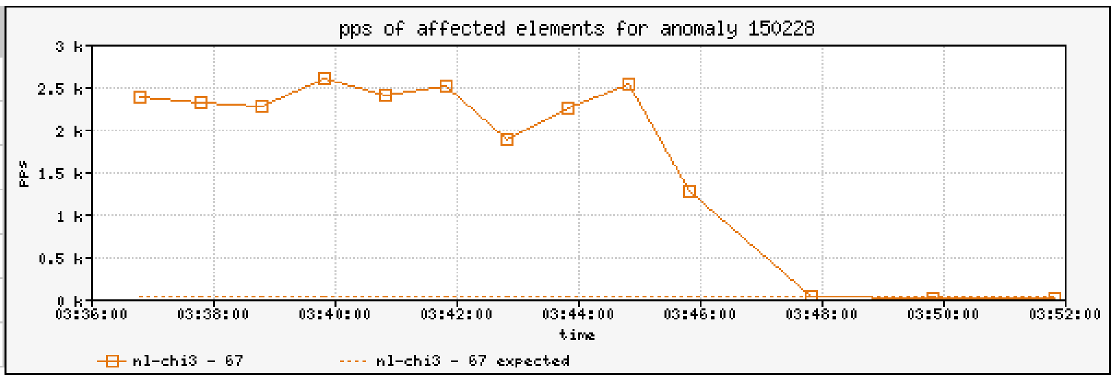
- Once baselines are built anomalous activity can be detected
  - Pure **rate-based** (pps or bps) anomalies may be legitimate or malicious
  - Many **misuse** attacks can be immediately recognized, even **without** baselines (e.g., TCP SYN or RST floods)
  - **Signatures** can also be defined to identify “interesting” transactional data (e.g., proto udp and port 1434 and 404 octets(376 payload) == slammer!)
  - Temporal compound signatures can be defined to detect with higher precision

Flow-based Commercial Tools...*

Anomaly 150228	Get Report: PDF XML

ID	Importance	Duration	Start Time	Direction	Type	Resource
150228	**High** 130.0% of 2 Kpps	17 mins	03:34, Aug 16	Incoming	Bandwidth (Profiled)	Microsoft 207.46.0.0/16 windowsupdate.com

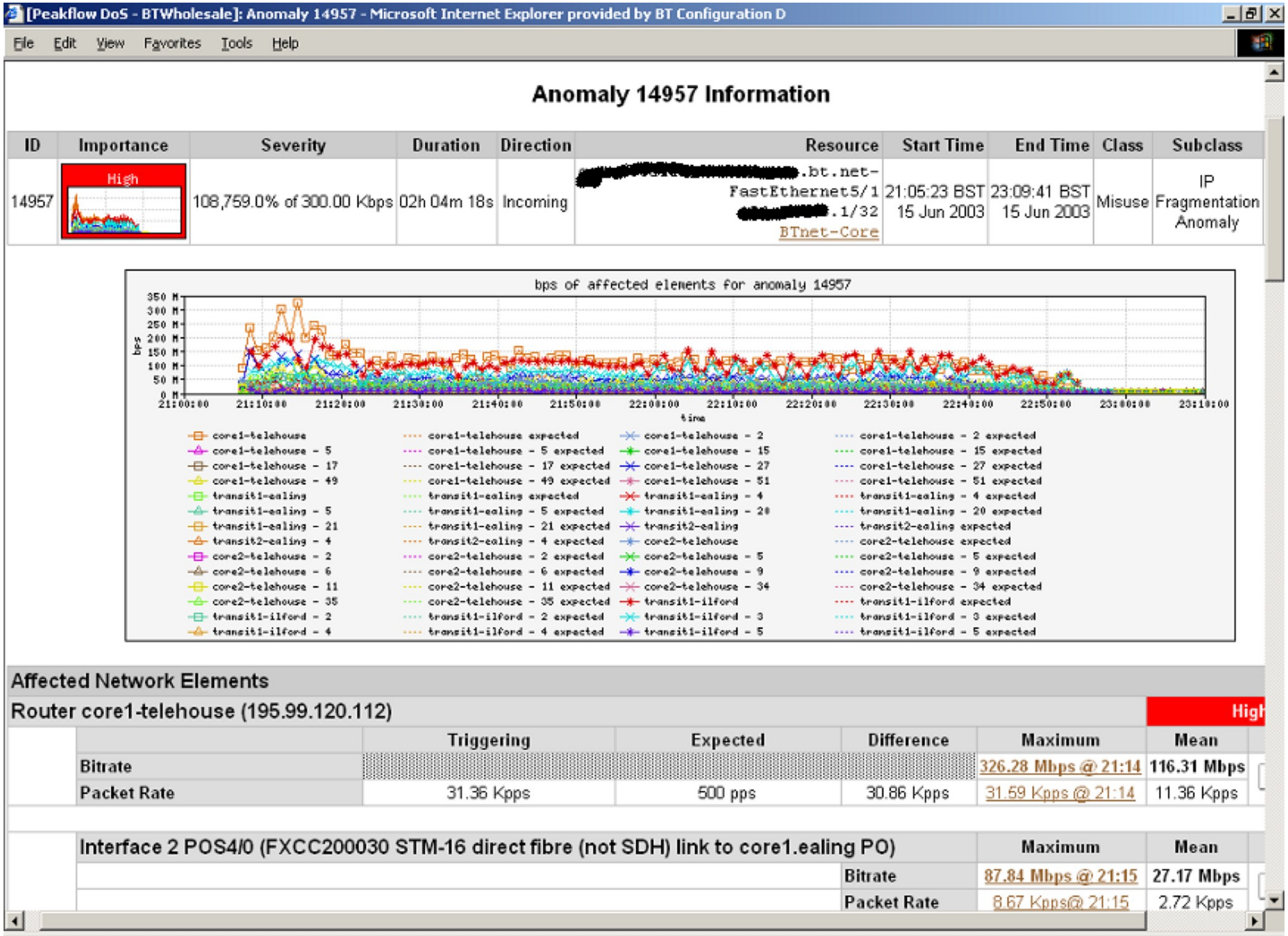
Traffic Characterization	
Sources	204.38.130.0/24
	204.38.130.192/26
	1024 - 1791
Destination	207.46.248.234/32
	80 (http)
Protocols	tcp (6)
TCP Flags	S (0x02)



Affected Network Elements		Expected	Observed bps		Observed pps		
	Importance	pps	Max	Mean	Max	Mean	
Router nl-chi3 198.110.131.125	**High**						
Interface 67 at-1/1/0.14 *pvc to WMU*		26	832 K	563.1 K	2.6 K	1.7 K	Details

Anomaly Comments

Commercial Detection
A Large Scale DOS attack*



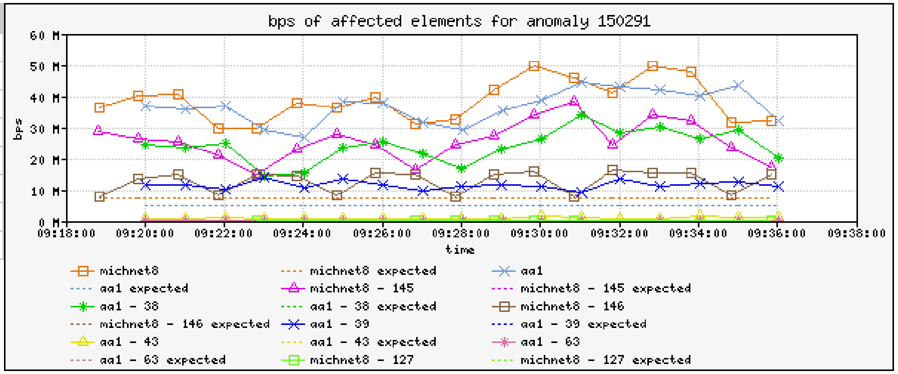
Traceback: Commercial*

Anomaly 150291 Get Report: PDF XML

ID	Importance	Duration	Start Time	Direction	Type	Resource
150291	High 124.6% of 40 Mbps	19 mins	09:16, Aug 17	Incoming	Protocol TCP (Profiled)	[redacted]

Traffic Characterization

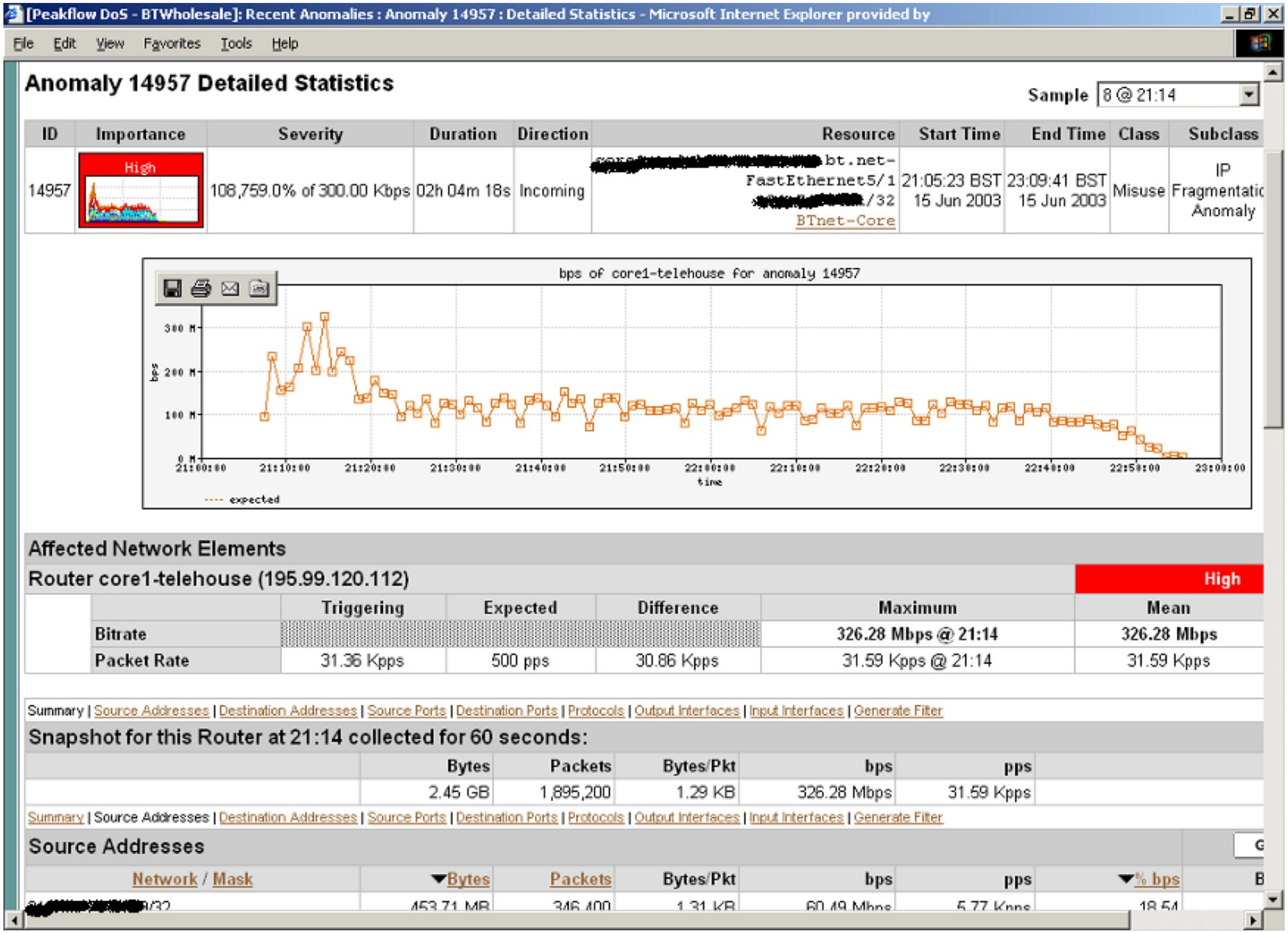
Sources	136.165.56.151/32
	69.1.194.74/32
	0 - 4095
Destination	[redacted]
	1409 (here-lm)
Protocols	tcp (6)
TCP Flags	AP (0x18) A (0x10)



Affected Network Elements

	Importance	Expected bps	Observed bps Max	Observed bps Mean	Observed pps Max	Observed pps Mean	
Router michnet8 198.108.90.125	High	7.2 M	49.9 M	38.7 M	5.3 K	4.2 K	Details
Interface 127 ATM1/0.27-aal5 layer 198.108.22.181 *pvc to NL-PORT1*		-	3.3 K	1.2 K	5	2.4	Details
Interface 145 GigabitEthernet5/0.22 - 802.1q vlan subinterface 198.108.23.153 *vlan to MichBin*		-	38.4 M	25.8 M	3.7 K	2.6 K	Details
Interface 146 GigabitEthernet5/0.24 - 802.1q vlan subinterface 198.108.23.245 *CHI-ANN_Bin*		-	16.6 M	12.8 M	1.9 K	1.6 K	Details
Router aa1 198.108.90.21	High	5.1 M	44.4 M	36.6 M	4.5 K	3.8 K	Details
Interface 38 so-0/2/0.1 192.122.183.9 *pvc to Abilene Indianapolis*		-	34.0 M	24.0 M	3.0 K	2.2 K	Details
Interface 39 so-0/2/0.2 63.149.0.186 *pvc to Qwest Chicago*		-	13.9 M	11.6 M	1.3 K	1.1 K	Details
Interface 43 so-1/0/0.0 208.172.10.138 *OC3 to C&W (Chicago)*		-	1.6 M	959.6 K	600	408.8	Details
Interface 63 ge-0/1/0.11 198.108.90.17 *;vlan to Comcast*		-	411.5 K	56.9 K	83.3	41.7	Details

Commercial Traceback: More Detail*



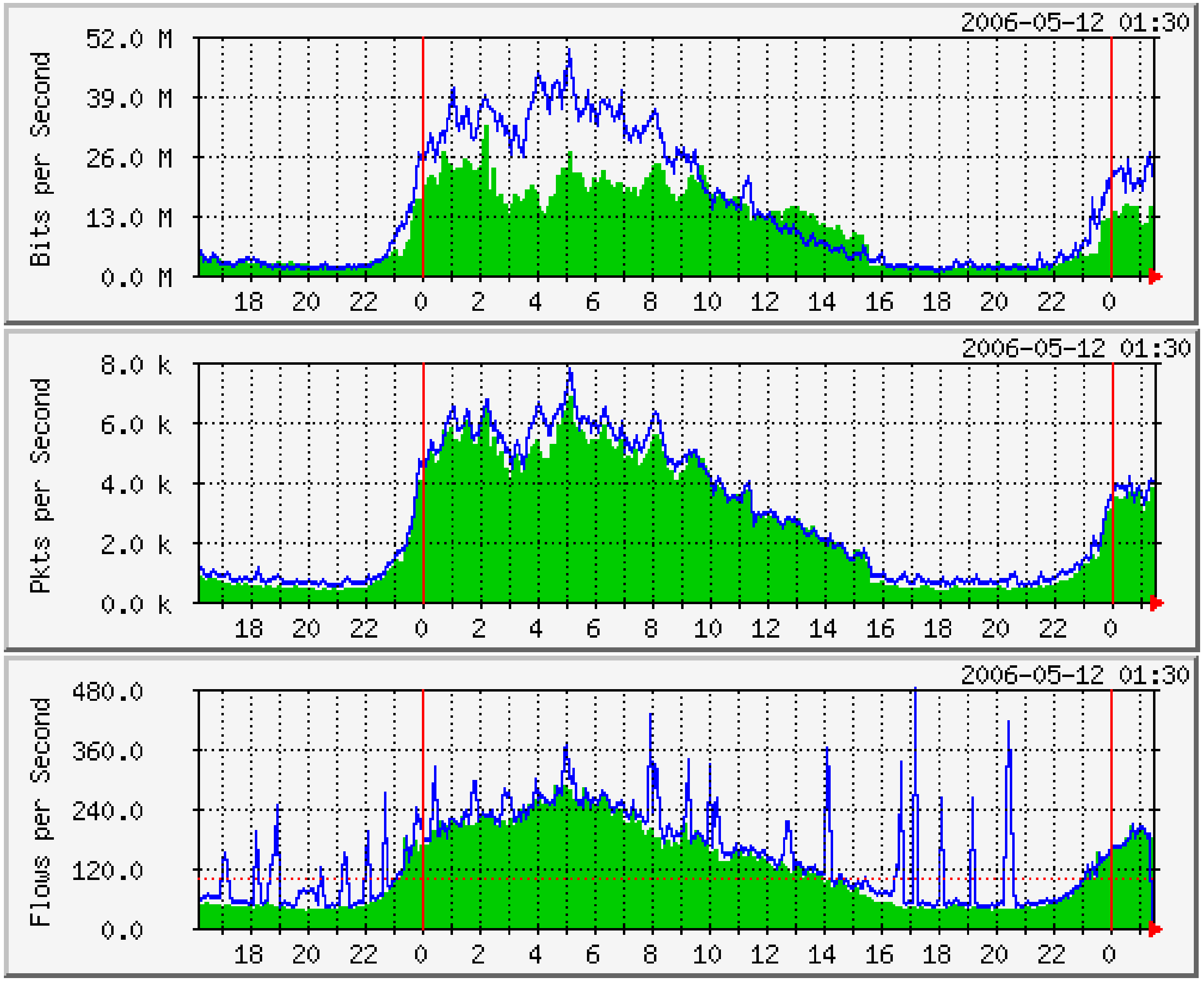
Traffic Analysis

- Can see traffic based on source and destination AS
  - Source and destination AS derived through the routing table on the router
  - Introduces the need to run full mesh BGP at IXPs as well as transit and peering
  - Source and destination prefix based flows can be collected and plotted against external prefix to ASN data

Accounting

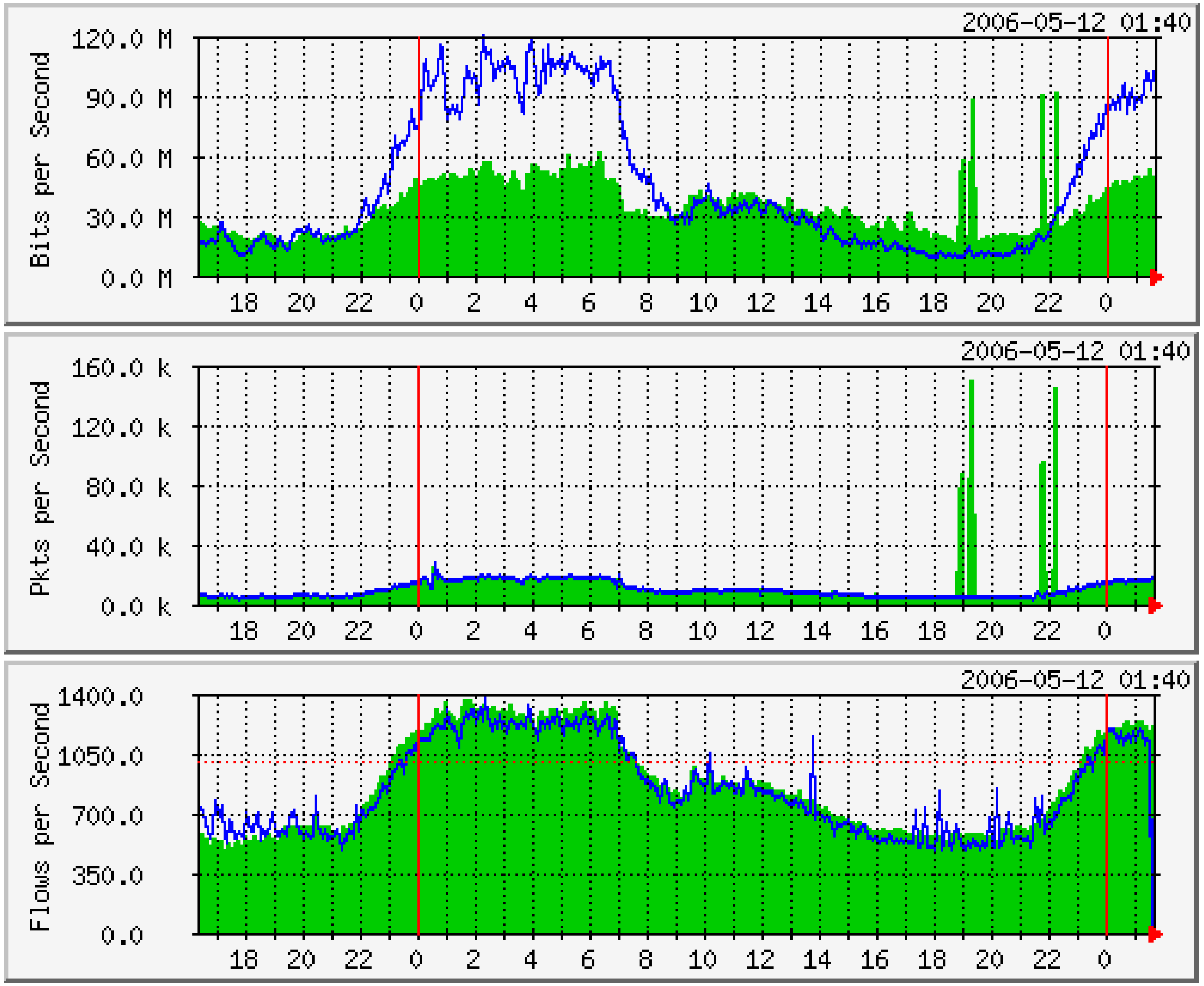
- Flow based accounting can be a good supplement to SNMP based accounting.

SNMP and Flows



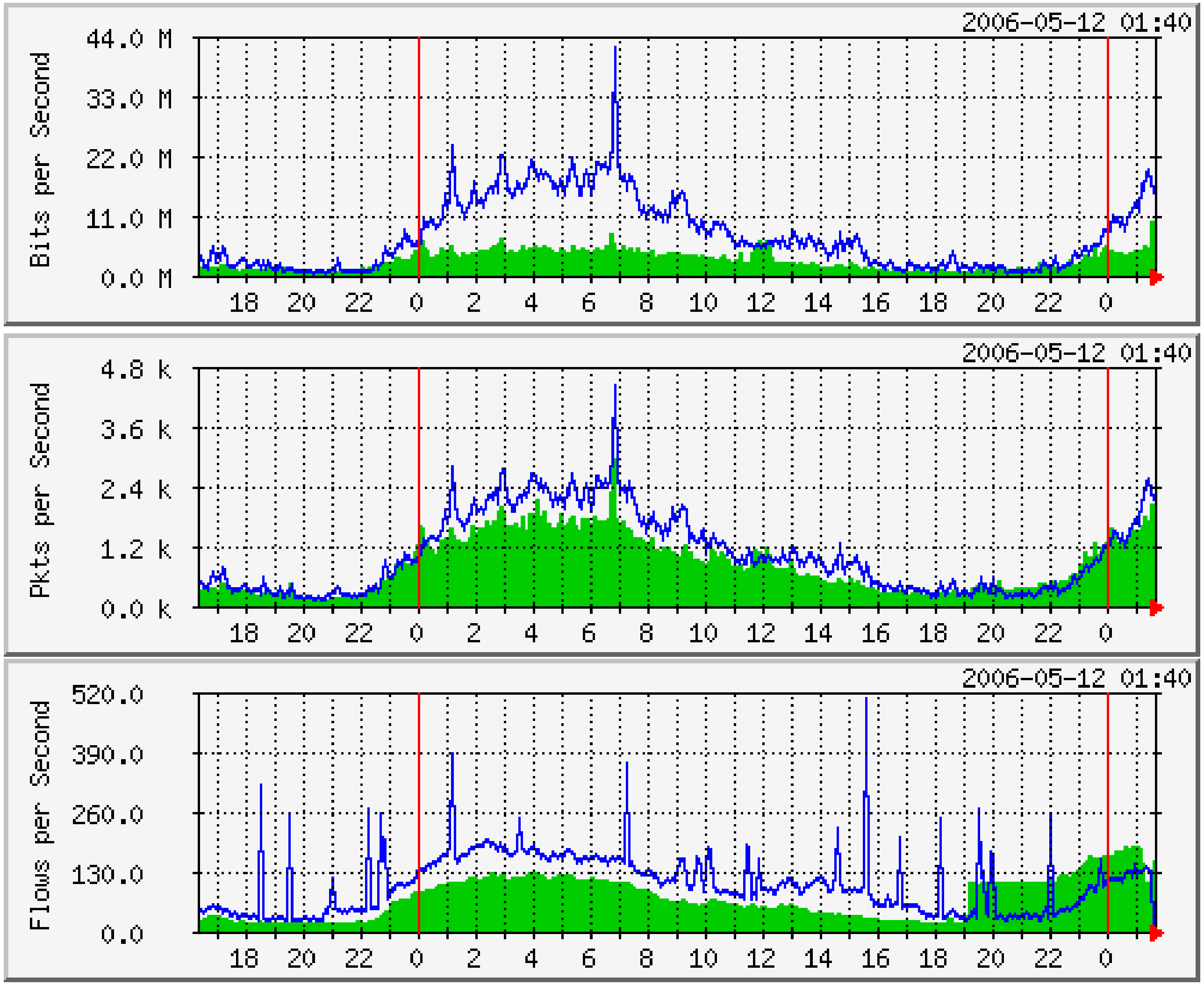
Data Courtesy AARNET, Australia and Bruce Morgan

See the fine lines..



Data Courtesy AARNET, Australia and Bruce Morgan

SNMP and Flows



Data Courtesy AARNET, Australia and Bruce Morgan

What Next

- IPFIX (IP Flow Information Exchange)
  - To make the flow format uniform and make it easier to write analysis tools
  - http://www1.ietf.org/html.charters/ipfix-charter.ht
  - Requirements for IP Flow Information Export (RFC 39

  - Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX) (RFC 3955)

References

- flow-tools:
  http://www.splintered.net/sw/flow-tools
- NetFlow Applications
  http://www.inmon.com/technology/netflowap
  ps.php
- Netflow HOW-TO
  http://www.linuxgeek.org/netflow-howto.php
- IETF standards effort:
  http://www.ietf.org/html.charters/ipfix-
  charter.html

References

- Abilene NetFlow page
  http://abilene-netflow.itec.oar.net/
- Flow-tools mailing list:
  flow-tools@splintered.net
- Cisco Centric Open Source Community
  http://cosi-nms.sourceforge.net/related.html