Netflow, Flow-tools tutorial

Gaurab Raj Upadhaya

Agenda

- Agenda bashing
  - Do you want to see the labs, or want to discuss issues
- 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

Net-flow

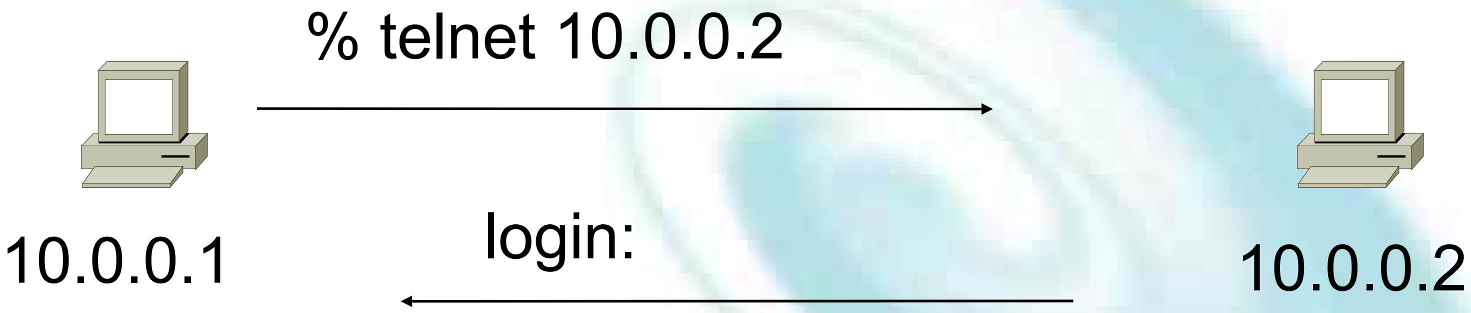
Network Flows

- 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



% telnet 10.0.0.2

10.0.0.1

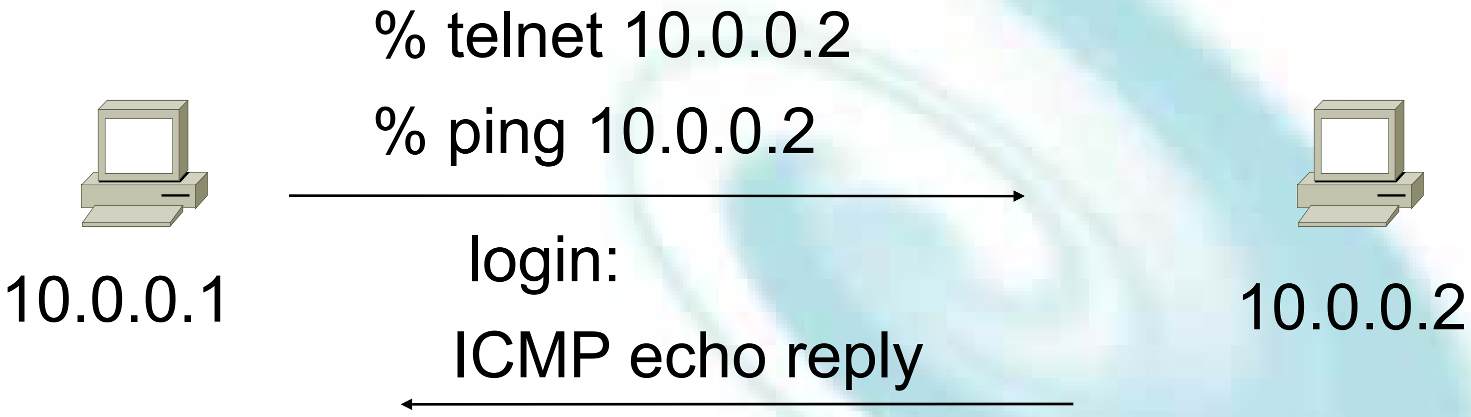
login:

10.0.0.2

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



% telnet 10.0.0.2

% ping 10.0.0.2

10.0.0.1

login:

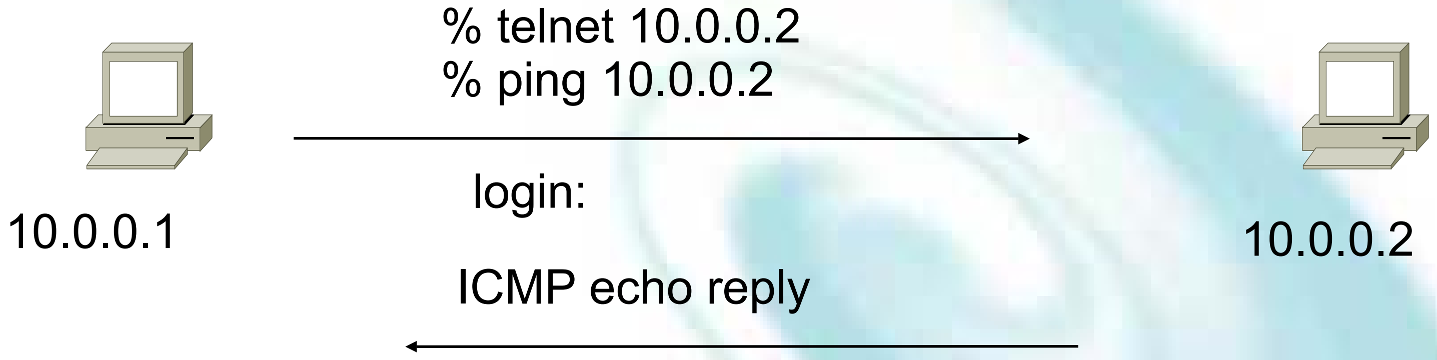
ICMP echo reply

10.0.0.2

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



% telnet 10.0.0.2
% ping 10.0.0.2

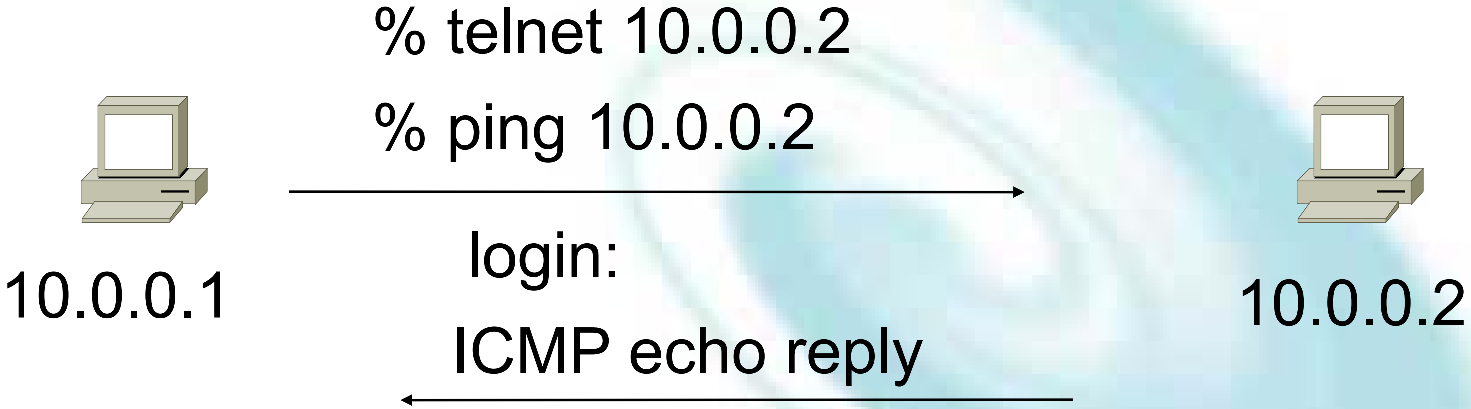
login:

ICMP echo reply

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



% telnet 10.0.0.2

% ping 10.0.0.2

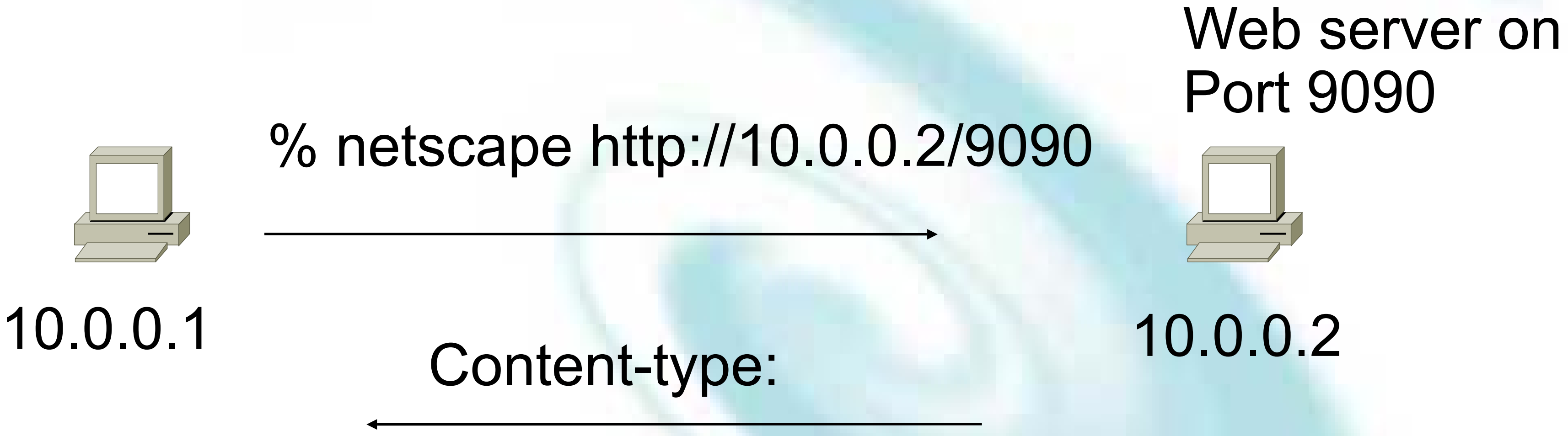
login:

ICMP echo reply

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



Web server on Port 9090

% netscape http://10.0.0.2/9090

10.0.0.1

10.0.0.2

Content-type:

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.

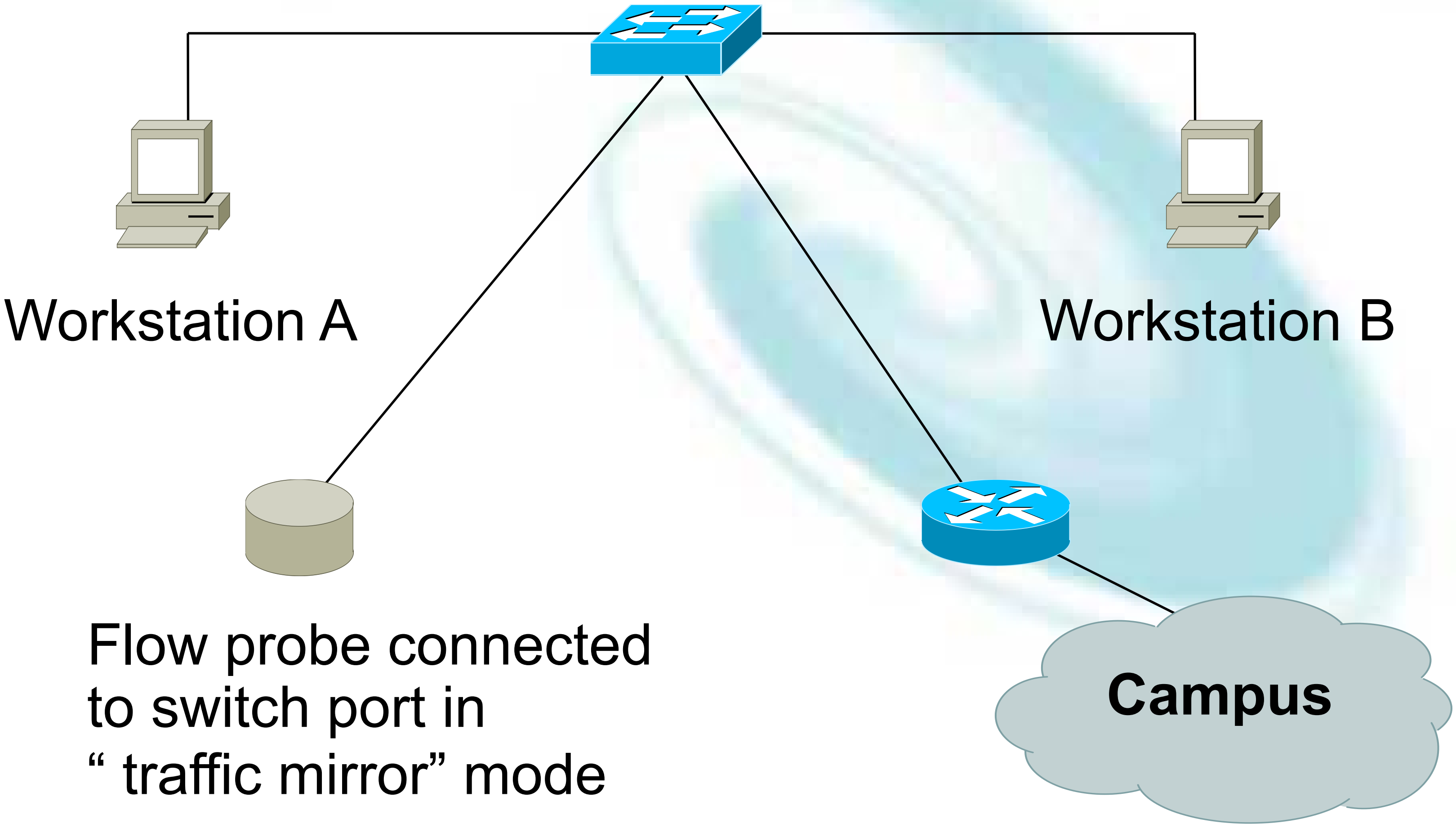
Flow Accounting

- 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



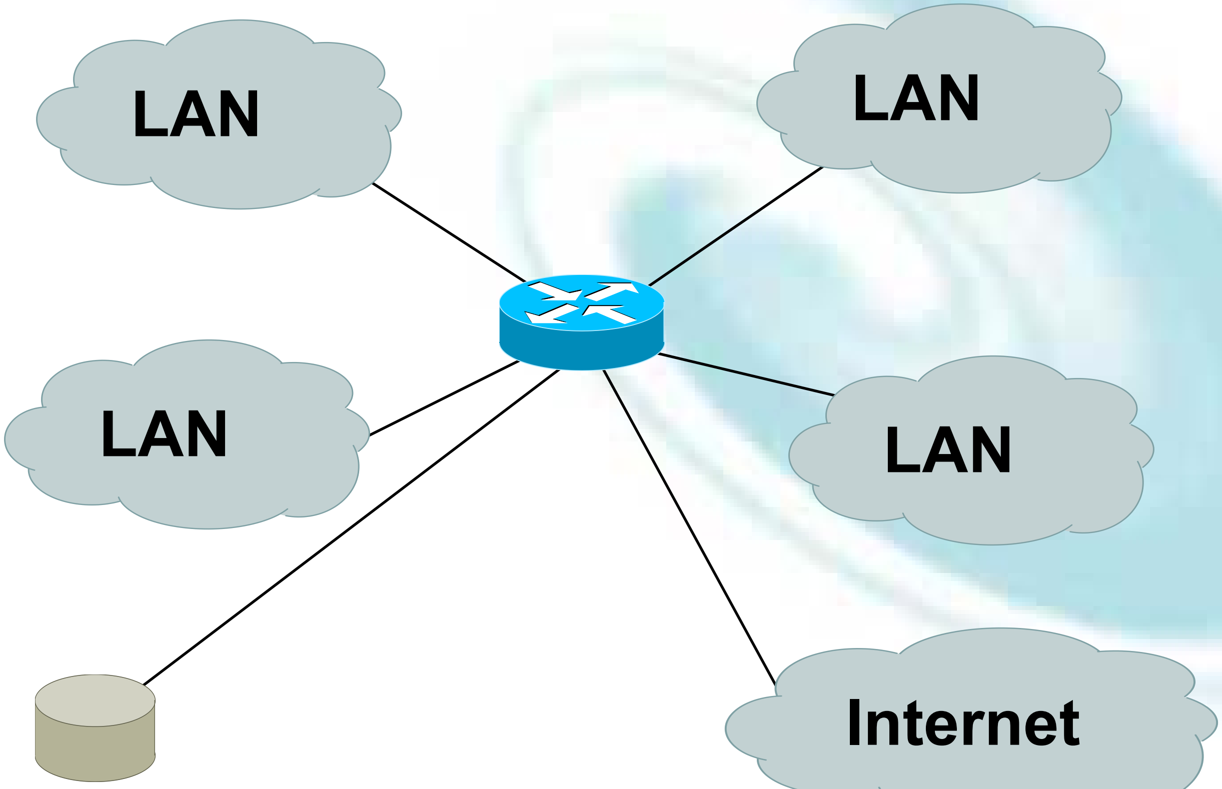
Workstation A

Workstation B

Flow probe connected
to switch port in
“ traffic mirror” mode

Campus

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 CatIOS platforms.
- Catalyst NetFlow is different implementation.

Cisco NetFlow Versions

- 4 Unaggregated types (1,5,6,7).
- 14 Aggregated types (8.x).
- 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.
- 3 Catalyst 65xx specific that correspond to the configurable flow mask.
- Much less data to post process, but lose fine granularity of v5 – no IP addresses.

NetFlow v8

- AS
- Protocol/Port
- Source Prefix
- Destination Prefix
- Prefix
- Destination (Catalyst 65xx)
- Source/Destination (Catalyst 65xx)
- Full Flow (Catalyst 65xx)

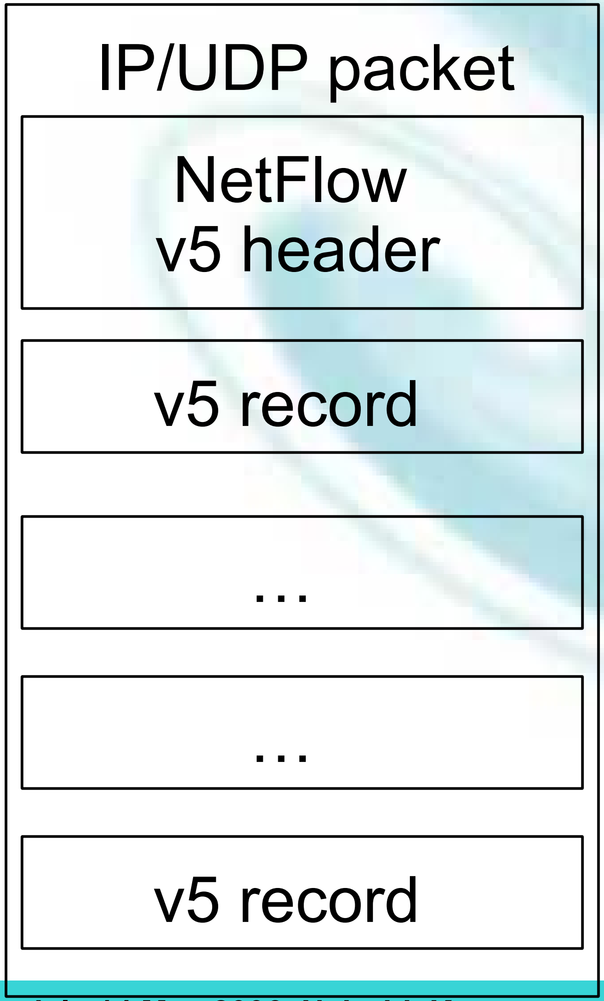
NetFlow v8

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

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



IP/UDP packet

NetFlow
v5 header

v5 record

…

…

v5 record

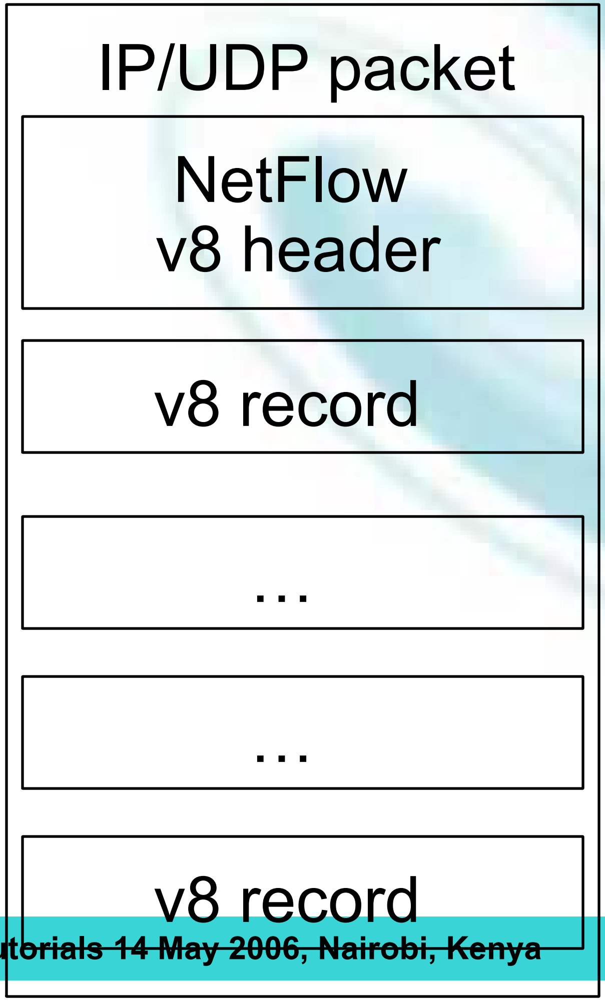
NetFlow v5 Packet (Header)

```
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)



IP/UDP packet

NetFlow v8 header

v8 record

…

…

v8 record

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 FastEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 ip route-cache flow

interface ATM1/0/0
 no ip address
 no ip directed-broadcast
 ip route-cache flow

interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 no ip directed-broadcast

ip flow-export version 5 origin-as
ip flow-export destination 10.0.0.10 5004
ip flow-export source loopback 0

ip flow-aggregation cache prefix
 export destination 10.0.0.10 5555
 enabled
```

Cisco IOS Configuration

```
krc4#sh ip flow export
Flow export is enabled
  Exporting flows to 10.0.0.10 (5004)
  Exporting using source IP address 10.10.10.10
  Version 5 flow records, origin-as
  Cache for prefix aggregation:
    Exporting flows to 10.0.0.10 (5555)
    Exporting using source IP address 10.10.10.10
  3176848179 flows exported in 105898459 udp datagrams
  0 flows failed due to lack of export packet
  45 export packets were sent up to process level
  0 export packets were punted to the RP
  5 export packets were dropped due to no fib
  31 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
  0 export packets were dropped enqueuing for the RP
  0 export packets were dropped due to IPC rate limiting
  0 export packets were dropped due to output drops
```

Cisco IOS Configuration

```
krc4#sho ip ca fl
IP packet size distribution (106519M total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .002 .405 .076 .017 .011 .010 .007 .005 .004 .005 .004 .004 .003 .002 .002

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .002 .006 .024 .032 .368 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  36418 active, 29118 inactive, 3141073565 added
  3132256745 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet     2951815      0.6        61   216     42.2      26.6      21.4
TCP-FTP       24128311      5.6        71   748    402.3      15.0      26.3
TCP-FTPD       2865416      0.6       916   843    611.6      34.7      19.8
TCP-WWW      467748914    108.9        15   566   1675.8       4.9      21.6
TCP-SMTP      46697428     10.8        14   370    159.6       4.0      20.1
TCP-X           521071      0.1       203   608     24.7      24.5      24.2
TCP-BGP        2835505      0.6         5    94      3.3      16.2      20.7
```

Cisco IOS Configuration

```
krc4#sho ip ca fl

TCP-other    1620253066    377.2          47    631  18001.6        27.3        23.4
UDP-DNS       125622144     29.2           2     78     82.5         4.6        24.7
UDP-NTP        67332976     15.6           1     76     22.0         2.7        23.4
UDP-TFTP          37173      0.0           2     76      0.0         4.1        24.6
UDP-Frag          68421      0.0         474    900      7.5       111.7        21.6
UDP-other     493337764    114.8          17    479   1990.3         3.8        20.2
ICMP          243659509     56.7           3    166    179.7         3.3        23.3
IGMP              18601      0.0          96     35      0.4       941.4         8.1
IPINIP            12246      0.0          69     52      0.1       548.4        15.2
GRE              125763      0.0         235    156      6.9        50.3        21.1
IP-other       75976755     17.6           2     78     45.4         3.9        22.8
Total:       3176854246    739.6          33    619  24797.4        16.2        22.6

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
AT5/0/0.4     206.21.162.150  AT1/0/0.1     141.219.73.45   06 0E4B A029   507
AT4/0/0.10    132.235.174.9   AT1/0/0.1     137.99.166.126  06 04BE 074C     3
AT4/0/0.12    131.123.59.33   AT1/0/0.1     137.229.58.168  06 04BE 09BB   646
AT1/0/0.1     137.99.166.126  AT4/0/0.10    132.235.174.9   06 074C 04BE     3
```

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 route cache flow
  - show ip route flow top-talkers

Juniper Configuration

- Sample packets with firewall filter and forward to routing engine.
- Sampling rate is limited to 7000pps. Fine for traffic engineering, but restrictive for DoS and intrusion detection.
- Juniper calls NetFlow cflowd.

Juniper Configration

Firewall filter

```
firewall {
    filter all {
        term all {
            then {
                sample;
                accept;
            }
        }
    }
}
```

Enable sampling / flows

```
forwarding-options {
    sampling {
        input {
            family inet {
                rate 100;
            }
        }
        output {
            cflowd 10.0.0.16{
                port 2055;
                version 5;
            }
        }
    }
}
```

Juniper Configration

Apply firewall filter to each interface.

```
interfaces {
    ge-0/3/0 {
        unit 0 {
            family inet {
              filter {
                    input all;
                    output all;
                }
                address 192.148.244.1/24;
            }
        }
    }
```

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

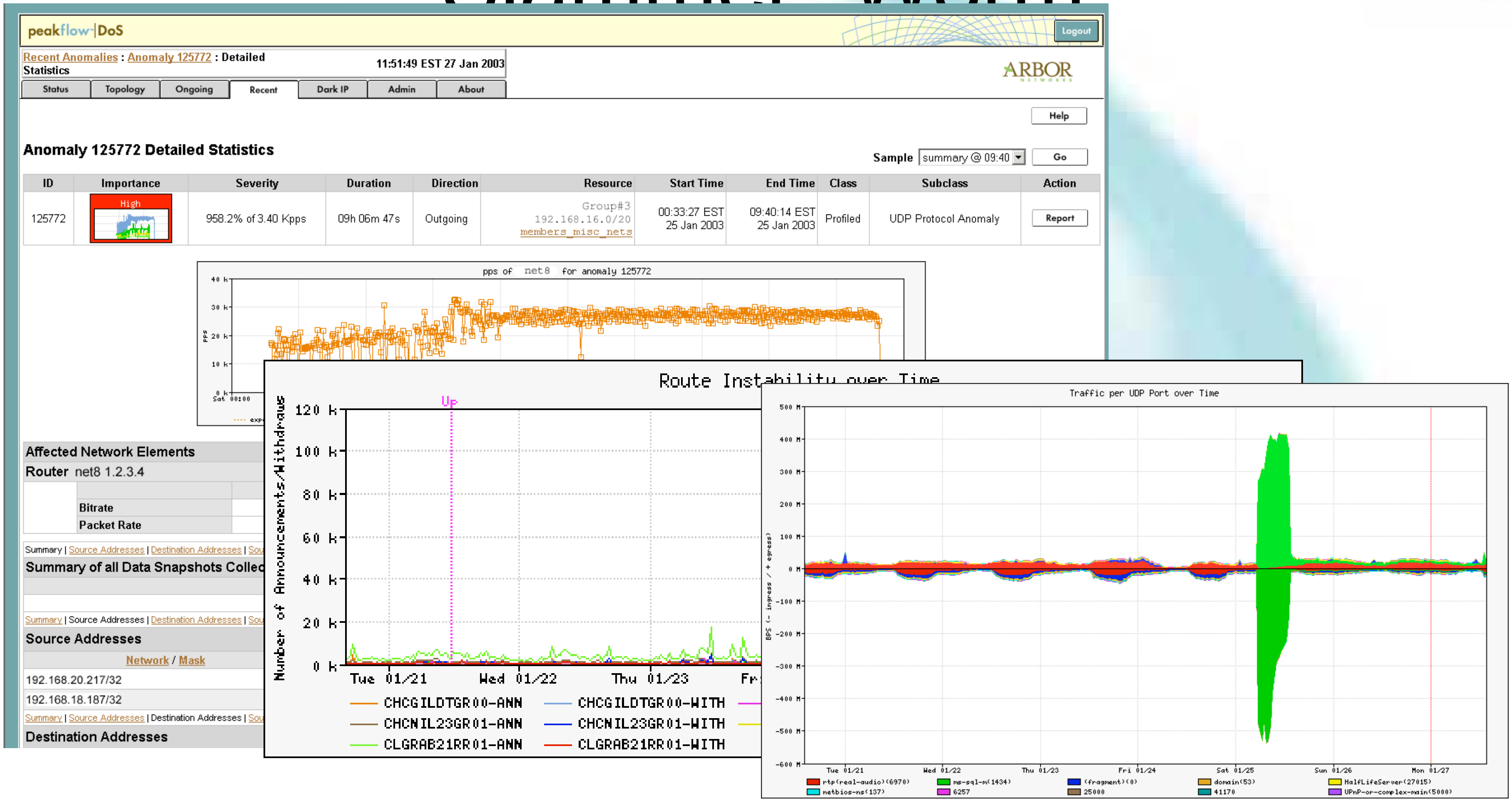
Traceback: Flow-based*

- 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*



peakflow | DoS

Logout

Recent Anomalies : Anomaly 125772 : Detailed Statistics

11:51:49 EST 27 Jan 2003

ARBOR NETWORKS

Status | Topology | Ongoing | Recent | Dark IP | Admin | About

Help

Anomaly 125772 Detailed Statistics

Sample: summary @ 09:40 Go

ID	Importance	Severity	Duration	Direction	Resource	Start Time	End Time	Class	Subclass	Action
125772	High	958.2% of 3.40 Kpps	09h 06m 47s	Outgoing	Group#3 192.168.16.0/20 members_misc_nets	00:33:27 EST 25 Jan 2003	09:40:14 EST 25 Jan 2003	Profiled	UDP Protocol Anomaly	Report

Affected Network Elements

Router net8 1.2.3.4

Bitrate

Packet Rate

Summary | Source Addresses | Destination Addresses | Sou

Summary of all Data Snapshots Collec

Summary | Source Addresses | Destination Addresses | Sou

Source Addresses

Network / Mask
192.168.20.217/32
192.168.18.187/32

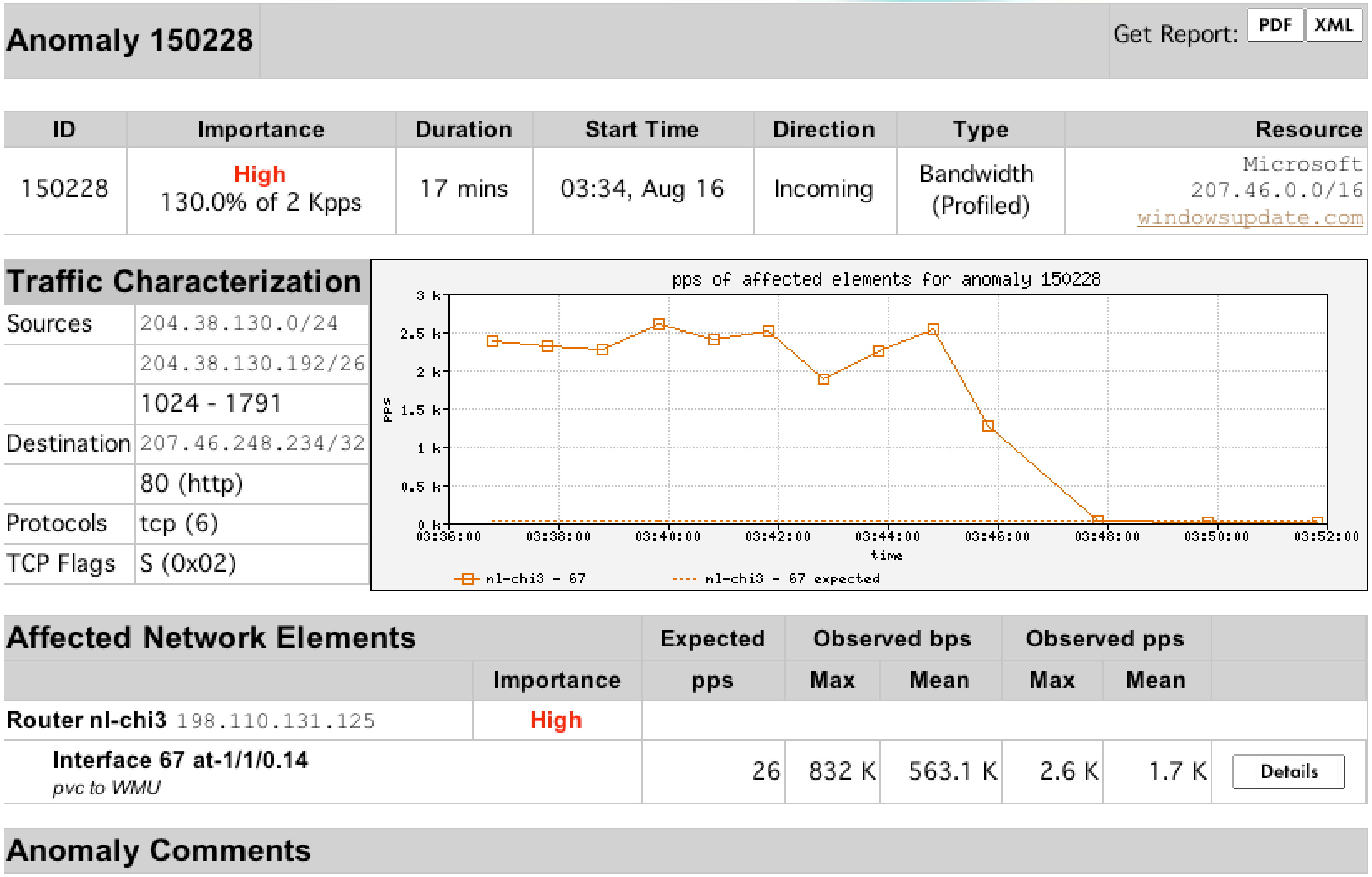
Summary | Source Addresses | Destination Addresses | Sou

Destination Addresses

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

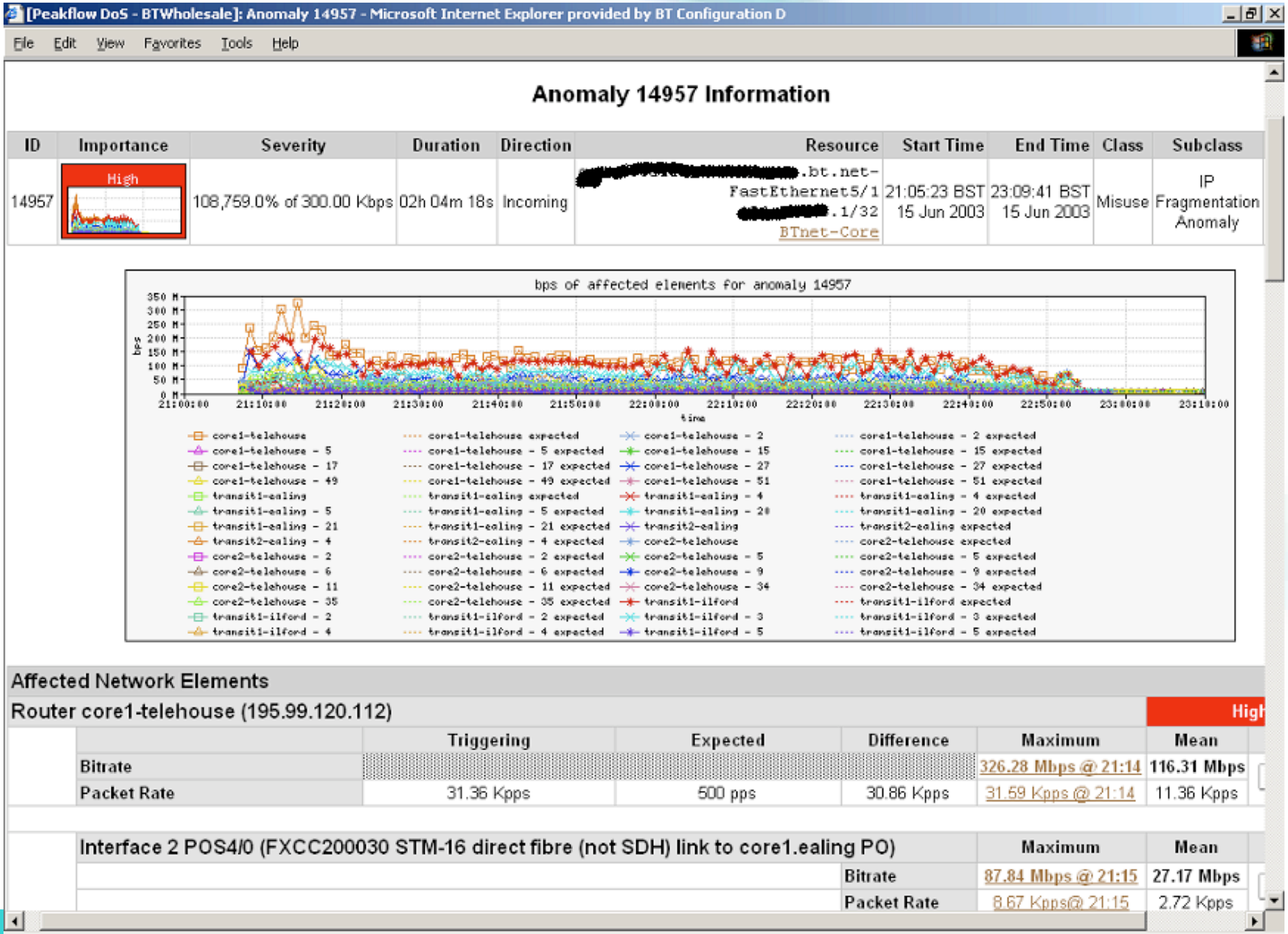
ID	Importance	Duration	Start Time	Direction	Type	Resource
150228	High 130.0% of 2 Kpps	17 mins	03:34, Aug 16	Incoming	Bandwidth (Profiled)	Microsoft 207.46.0.0/16 windowsupdate.com

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*



[Peakflow DoS - BTWholesale]: Anomaly 14957 - Microsoft Internet Explorer provided by BT Configuration D

Anomaly 14957 Information

ID	Importance	Severity	Duration	Direction	Resource	Start Time	End Time	Class	Subclass
14957	High	108,759.0% of 300.00 Kbps	02h 04m 18s	Incoming	[redacted].bt.net-FastEthernet5/1 [redacted].1/32 BTnet-Core	21:05:23 BST 15 Jun 2003	23:09:41 BST 15 Jun 2003	Misuse	IP Fragmentation Anomaly

bps of affected elements for anomaly 14957

Affected Network Elements

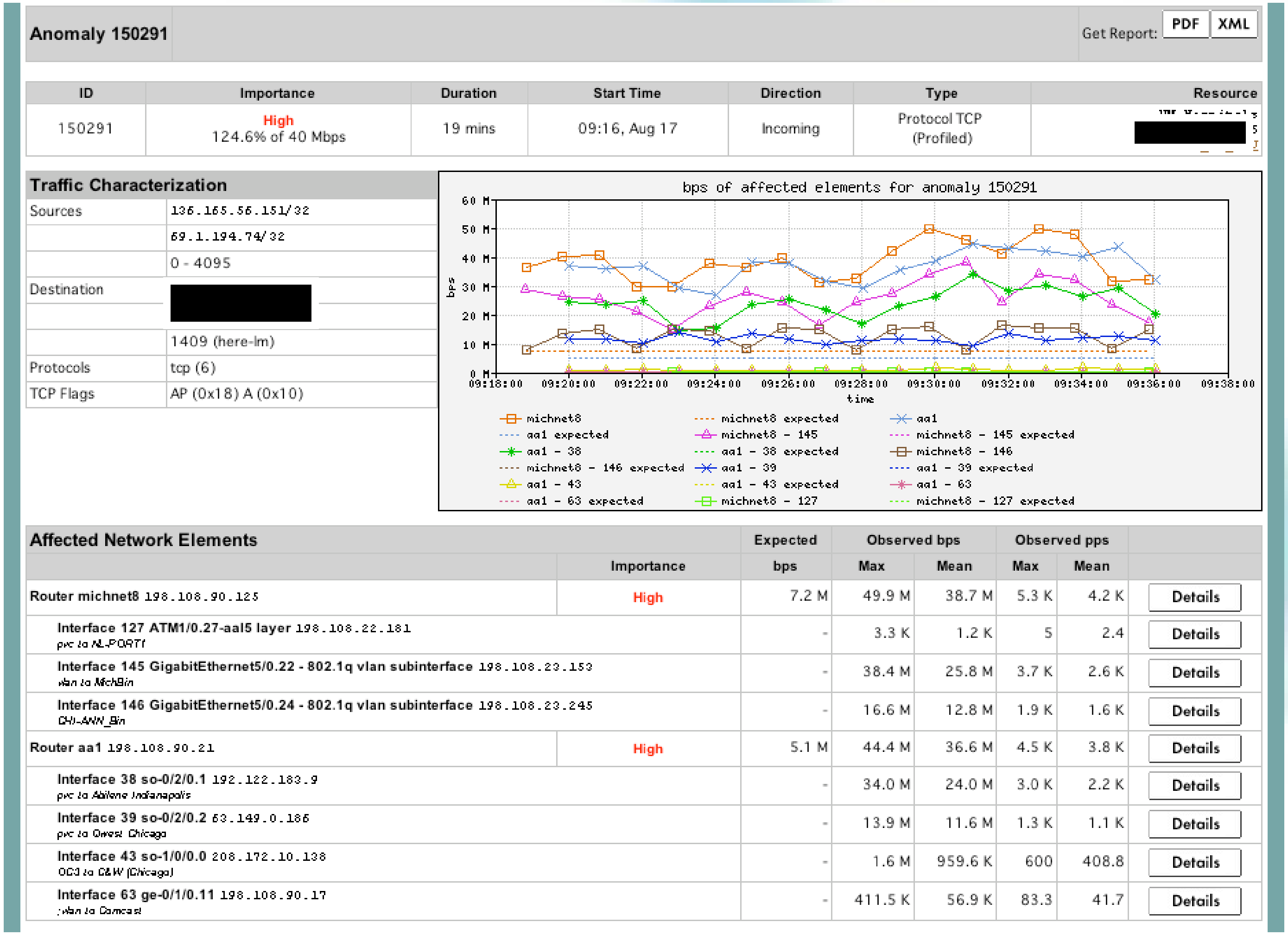
Router core1-telehouse (195.99.120.112) High

	Triggering	Expected	Difference	Maximum	Mean
Bitrate				326.28 Mbps @ 21:14	116.31 Mbps
Packet Rate	31.36 Kpps	500 pps	30.86 Kpps	31.59 Kpps @ 21:14	11.36 Kpps

Interface 2 POS4/0 (FXCC200030 STM-16 direct fibre (not SDH) link to core1.ealing PO)

	Maximum	Mean
Bitrate	87.84 Mbps @ 21:15	27.17 Mbps
Packet Rate	8.67 Kpps@ 21:15	2.72 Kpps

Traceback: Commercial*



Anomaly 150291 Get Report: PDF XML

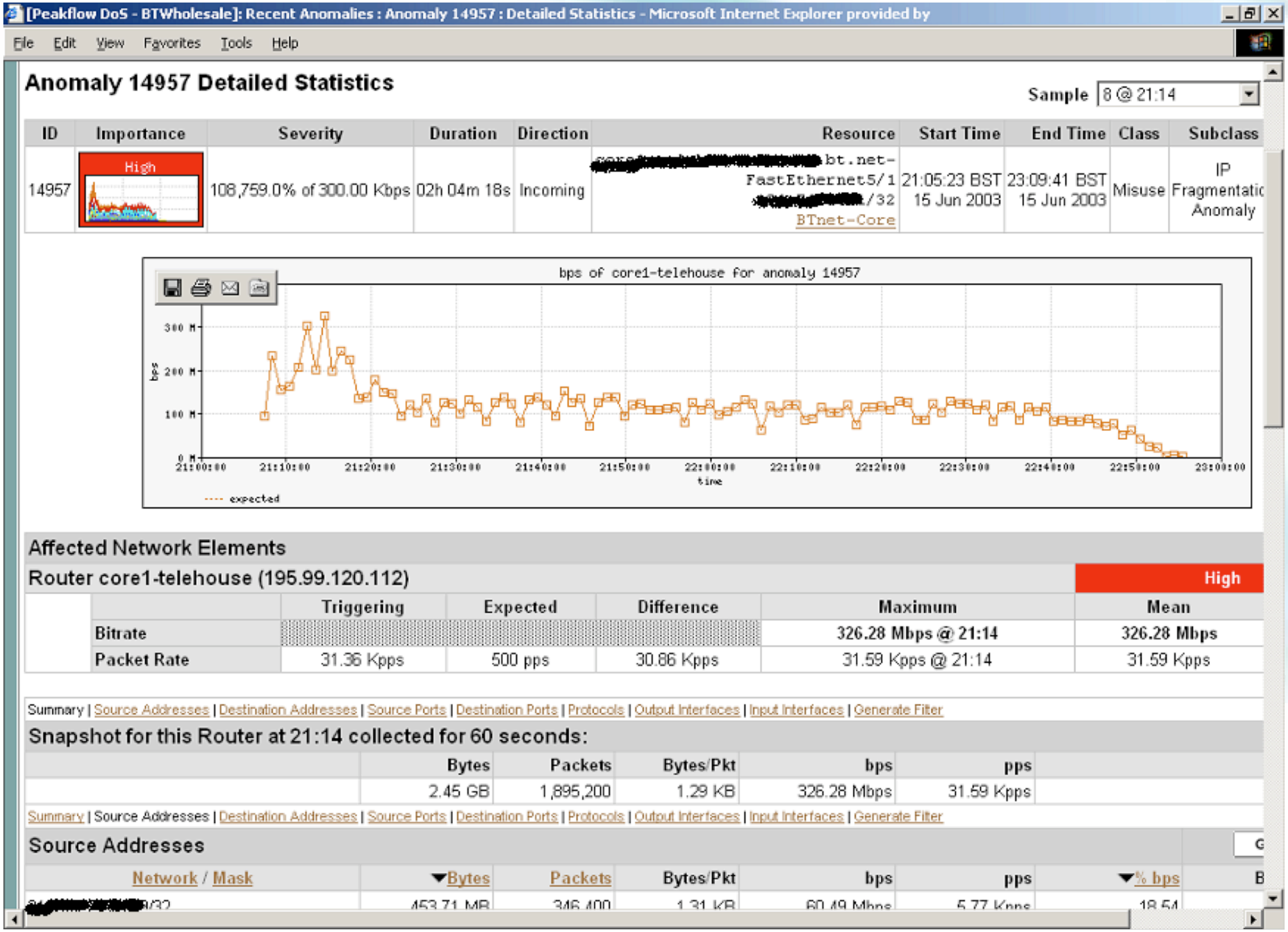
ID	Importance	Duration	Start Time	Direction	Type	Resource
150291	High 124.6% of 40 Mbps	19 mins	09:16, Aug 17	Incoming	Protocol TCP (Profiled)	[redacted]

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		Expected	Observed bps		Observed pps		
	Importance	bps	Max	Mean	Max	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-PORTf*		-	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*



Anomaly 14957 Detailed Statistics

Sample 8 @ 21:14

ID	Importance	Severity	Duration	Direction	Resource	Start Time	End Time	Class	Subclass
14957	High	108,759.0% of 300.00 Kbps	02h 04m 18s	Incoming	[redacted]bt.net- FastEthernet5/1 [redacted]/32 BTnet-Core	21:05:23 BST 15 Jun 2003	23:09:41 BST 15 Jun 2003	Misuse	IP Fragmentatic Anomaly

Affected Network Elements

Router core1-telehouse (195.99.120.112) High

		Triggering	Expected	Difference	Maximum	Mean
	Bitrate				326.28 Mbps @ 21:14	326.28 Mbps
	Packet Rate	31.36 Kpps	500 pps	30.86 Kpps	31.59 Kpps @ 21:14	31.59 Kpps

Summary | Source Addresses | Destination Addresses | Source Ports | Destination Ports | Protocols | Output Interfaces | Input Interfaces | Generate Filter

Snapshot for this Router at 21:14 collected for 60 seconds:

	Bytes	Packets	Bytes/Pkt	bps	pps
	2.45 GB	1,895,200	1.29 KB	326.28 Mbps	31.59 Kpps

Summary | Source Addresses | Destination Addresses | Source Ports | Destination Ports | Protocols | Output Interfaces | Input Interfaces | Generate Filter

Source Addresses

Network / Mask	▼Bytes	Packets	Bytes/Pkt	bps	pps	▼% bps	B
[redacted]/32	453.71 MB	346,400	1.31 KB	60.49 Mbps	5.77 Kpps	18.54	

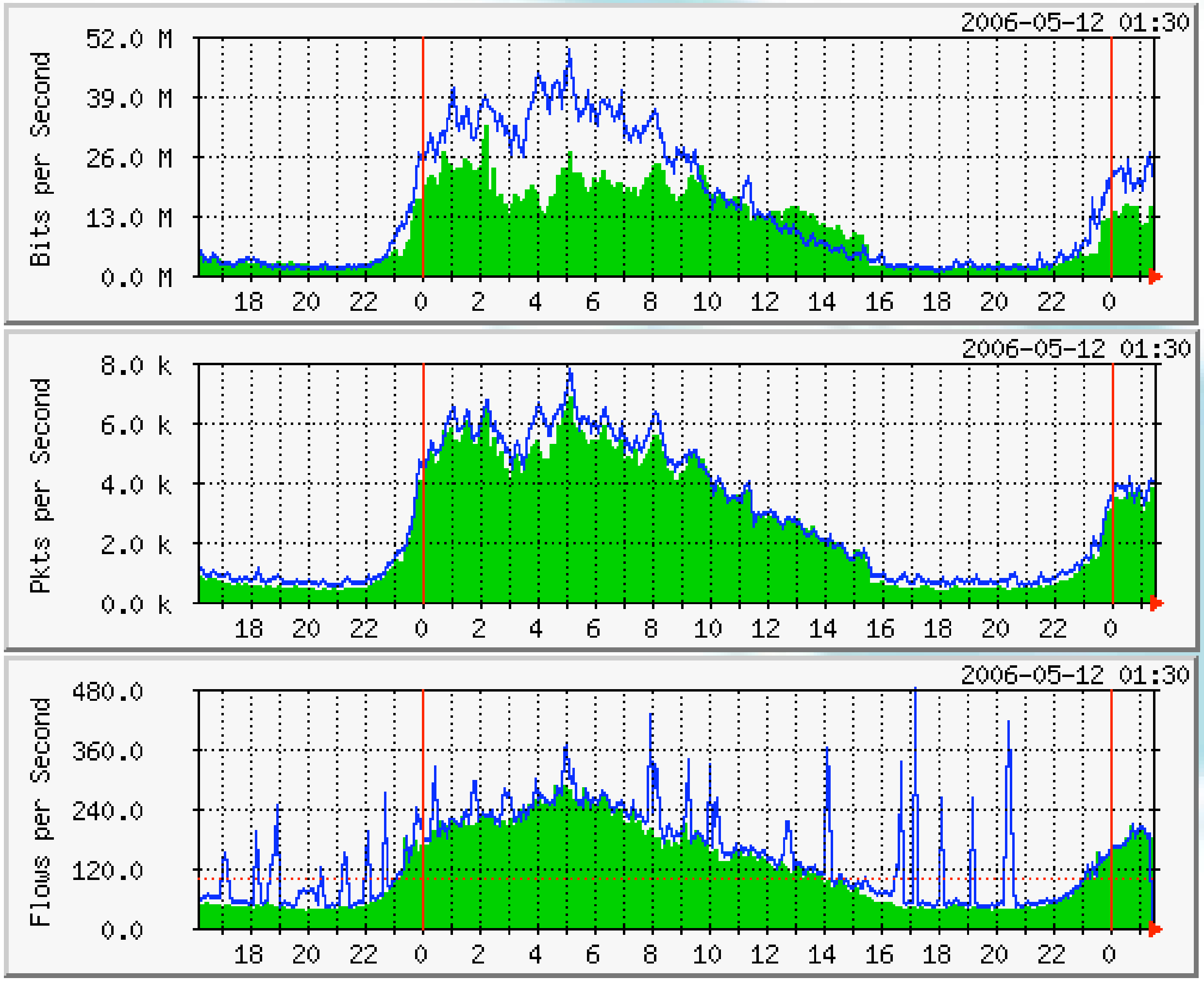
Traffic Analysis

- 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

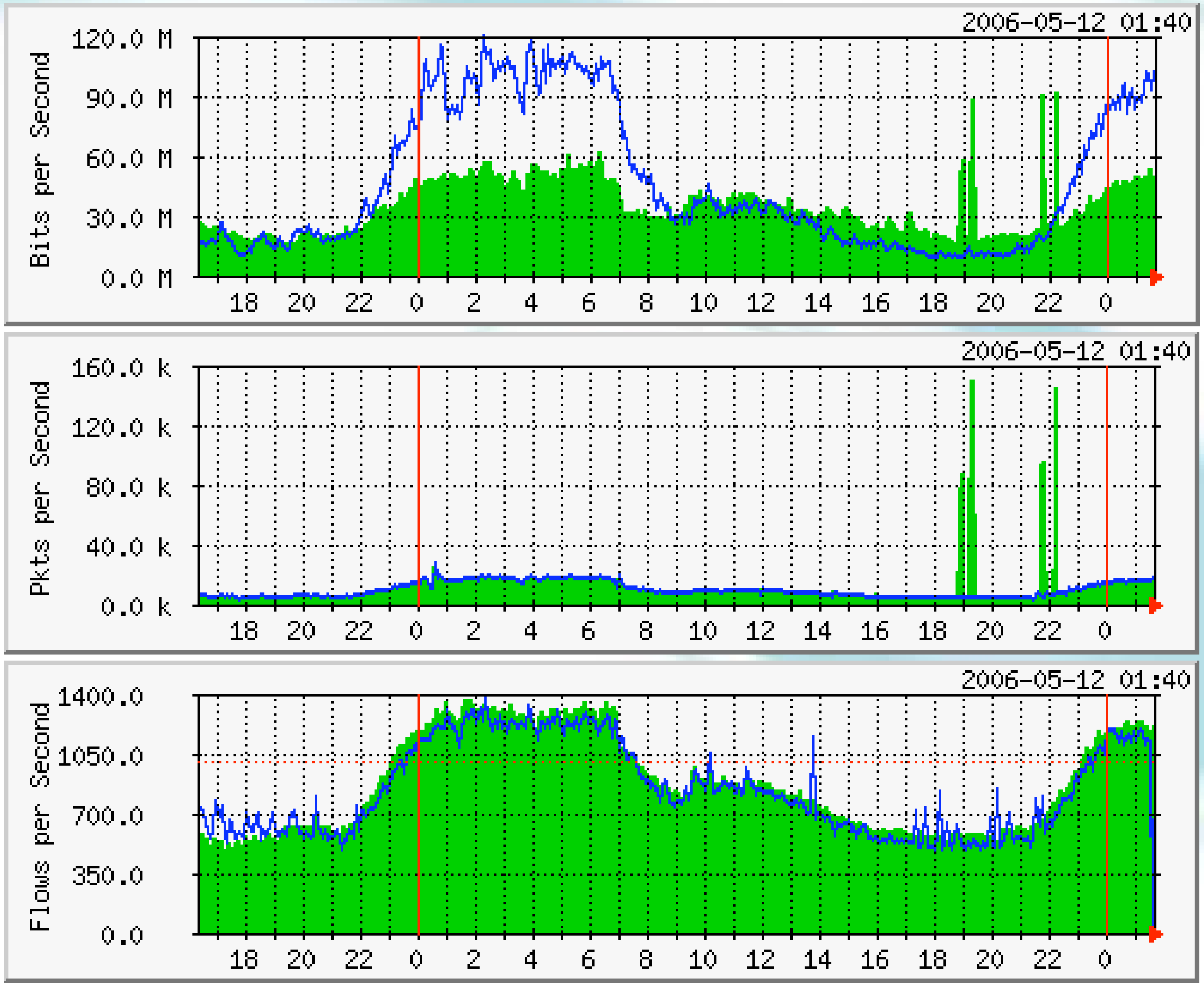
- Flow based accounting can be a good supplement to SNMP based accounting.

SNMP and Flows



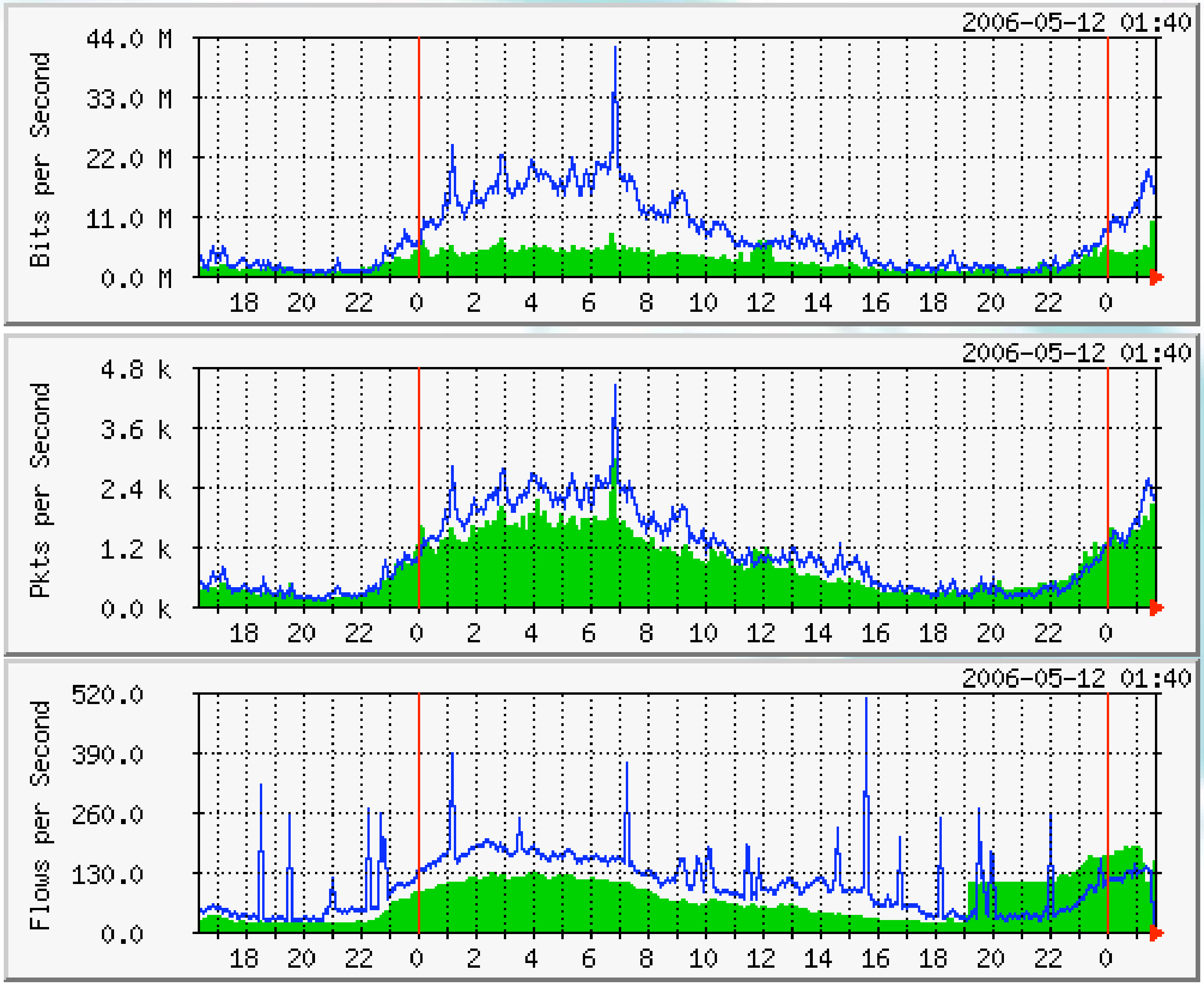
Data Courtesy AARNET, Australia and Bruce Morgan

See the fine lines..



Data Courtesy AARNET, Australia and Bruce Morgan

SNMP and Flows



Data Courtesy AARNET, Australia and Bruce Morgan

Flow-tools

- Collection of programs to post process Cisco NetFlow compatible flows.
- Written in C, designed to be fast (scales to large installations).
- Includes library (ftlib) for custom applications.
- Installation with configure;make;make install on most platforms (FreeBSD, Linux, Solaris, BSDi, NetBSD).

flow-capture

- Collect NetFlow exports and stores to disk.
- Built in compression.
- Manages disk space by expiring older flow files at configurable limits.
- Detects lost flows by missing sequence numbers and stores with flow metadata.

flow-fanout

- Replicate NetFlow UDP streams from one source to many destinations.
- Destination may be a multicast address.

flow-expire

- Expire (remove) old flow files based on disk usage.
- Same functionality built in to flow-capture.
- Used when managing disk space in a distributed environment.

Collector Placement and configuration

- NetFlow is UDP so the collector should ideally be directly connected to the router to minimize packet loss and IP spoofing risks.
- No flow control. Undersized collector will drop flows. Monitor netstat –s | grep buf and configure syslog so dropped flows will be logged.

flow-print

- Formatted output of flow files.

```
eng1:% flow-print < ft-v05.2002-01-21.093345-0500 | head -15
srcIP            dstIP             prot  srcPort  dstPort  octets packets
131.238.205.199  194.210.13.1      6     6346     40355    221          5
192.5.110.20     128.195.186.5     17    57040    33468    40           1
128.146.1.7      194.85.127.69     17    53       53       64           1
193.170.62.114   132.235.156.242   6     1453     1214     192          4
134.243.5.160    192.129.25.10     6     80       3360     654          7
132.235.156.242  193.170.62.114    6     1214     1453     160          4
130.206.43.51    130.101.99.107    6     3226     80       96           2
206.244.141.3    128.163.62.17     6     35593    80       739         10
206.244.141.3    128.163.62.17     6     35594    80       577          6
212.33.84.160    132.235.152.47    6     1447     1214     192          4
132.235.157.187  164.58.150.166    6     1214     56938    81           2
129.1.246.97     152.94.20.214     6     4541     6346     912         10
132.235.152.47   212.33.84.160     6     1214     1447     160          4
130.237.131.52   130.101.9.20      6     1246     80       902         15
```

flow-cat

- Concat many flow files or directories of files.

```
eng1:% ls
ft-v05.2002-01-21.160001-0500   ft-v05.2002-01-21.170001-0500
ft-v05.2002-01-21.161501-0500   ft-v05.2002-01-21.171501-0500
ft-v05.2002-01-21.163001-0500   ft-v05.2002-01-21.173001-0500
ft-v05.2002-01-21.164501-0500   tmp-v05.2002-01-21.174501-0500

eng1:% flow-cat . | flow-print

srcIP            dstIP            prot  srcPort  dstPort  octets packets
138.26.220.46    192.5.110.20     17    62242    33456    40           1
143.105.55.23    18.123.66.15     17    41794    41794    40           1
129.15.134.66    164.107.69.33    6     1214     2222     4500         3
132.235.170.19   152.30.96.188    6     6346     1475     128          3
```

flow-merge

- Flow-merge is similar to flow-cat except it maintains relative ordering of flows when combining the files.
- Typically used when combining flows from multiple collectors.

flow-filter

- Filter flows based on port, protocol, ASN, IP address, ToS bits, TCP bits, and tags.

```
eng1% flow-cat . | flow-filter -P119 | flow-print | head -10

srcIP            dstIP            prot  srcPort  dstPort  octets      packets
155.52.46.50     164.107.115.4    6     33225    119      114         2
128.223.220.29   129.137.4.135    6     52745    119      1438382     1022
155.52.46.50     164.107.115.4    6     33225    119      374         6
164.107.115.4    192.58.107.160   6     60141    119      5147961     8876
128.223.220.29   129.137.4.135    6     52745    119      1356325     965
128.223.220.29   129.137.4.135    6     52714    119      561016      398
130.207.244.18   129.22.8.64      6     36033    119      30194       121
155.52.46.50     164.107.115.4    6     33225    119      130         2
198.108.1.146    129.137.4.135    6     17800    119      210720652   216072
```

flow-split

- Split flow files into smaller files.
- Typically used with flow-stat and graphing. For example if flow files are 1 hour and want 5 minute data points in graph, flow-split can take the 1 hour flow files and generate 5 minute files.

flow-tag

- Adds a tag field to flows based on IP exporter, IP prefix, Autonomous System, or next hop.
- Like flow-filter used with other tools.
- Used to manage groups of prefixes or ASN’s.

flow-header

- Display meta information in flow file.

```
eng1:% flow-header < ft-v05.2002-01-21.093345-0500
#
# mode:                    normal
# capture hostname:        eng1.oar.net
# exporter IP address:     0.0.0.0
# capture start:           Mon Jan 21 09:33:45 2002
# capture end:             Mon Jan 21 09:45:01 2002
# capture period:          676 seconds
# compress:                on
# byte order:              little
# stream version:          3
# export version:          5
# lost flows:              0
# corrupt packets:         0
# sequencer resets:        0
# capture flows:           341370
#
```

flow-stat

- Generates reports from flow files.
- Output is readable and easily imported into graphing programs (gnuplot, etc).
- IP Address, IP address pairs, ports, packets, bytes, interfaces, next hop, Autonomous System, ToS bits, exporter, and tags.

flow-stat - summary

```
Total Flows                     : 24236730
Total Octets                    : 71266806610
Total Packets                   : 109298006
Total Time (1/1000 secs) (flows): 289031186084
Duration of data  (realtime)    : 86400
Duration of data (1/1000 secs)  : 88352112
Average flow time (1/1000 secs) : 11925.0000
Average packet size (octets)    : 652.0000
Average flow size (octets)      : 2940.0000
Average packets per flow        : 4.0000
Average flows / second (flow)   : 274.3201
Average flows / second (real)   : 280.5177
Average Kbits / second (flow)   : 6452.9880
Average Kbits / second (real)   : 6598.7781
```

flow-stat – Source AS % Total

```
#
# src AS         flows    octets   packets  duration
#
NSFNETTEST14-AS  6.430    6.582    7.019    5.693
ONENET-AS-1      2.914    4.417    3.529    3.566
UONET            0.600    4.052    2.484    1.979
UPITT-AS         1.847    3.816    2.697    2.552
CONCERT          1.786    2.931    2.391    1.955
OHIOU            3.961    2.601    2.140    1.655
CMU-ROUTER       1.962    2.577    2.349    2.075
BOSTONU-AS       1.503    2.126    1.665    1.914
PURDUE           2.185    1.994    2.157    2.507
STANFORD         2.124    1.950    2.270    2.636
UR               1.809    1.919    1.652    1.532
UMN-AGS-NET-AS   1.612    1.895    1.788    1.938
RISQ-AS          1.086    1.849    1.378    1.367
PENN-STATE       2.845    1.641    2.666    2.190
RIT-ASN          0.796    1.601    1.414    0.830
```

flow-stat – Dest AS % Total

```
#
# dst AS          flows    octets   packets  duration
#
NSFNETTEST14-AS   6.202    9.564    8.005    6.762
PENN-STATE        2.037    3.774    2.712    2.153
CONCERT           2.628    3.133    2.888    2.326
ONENET-AS-1       2.818    2.434    2.906    3.000
STANFORD          1.915    2.360    2.122    2.195
JANET             2.508    2.319    2.150    2.485
0                 0.831    2.187    2.431    2.910
DFN-WIN-AS        2.349    2.099    1.938    2.359
CMU-ROUTER        1.383    2.090    1.972    1.960
UONET             0.537    2.067    1.699    1.397
PURDUE            2.029    1.934    1.983    2.177
UMN-AGS-NET-AS    1.608    1.784    1.664    1.681
UPITT-AS          1.507    1.707    2.067    2.288
MIT-GATEWAYS      0.677    1.425    1.175    0.806
RIT-ASN           0.644    1.313    1.243    0.868
INDIANA-AS        0.899    1.285    0.996    0.781
```

flow-stat – Src/Dest AS % Total

```
#
# src AS            dst AS             flows    octets   packets  duration
#
GEORGIA-TECH        PENN-STATE         0.030    0.965    0.459    0.071
NWU-AS              0                  0.008    0.734    0.379    0.170
UONET               CONCERT            0.064    0.698    0.438    0.290
UCLA                NSFNETTEST14-AS    0.037    0.568    0.269    0.111
CONCERT             UONET              0.052    0.543    0.364    0.221
BCNET-AS            MIT-GATEWAYS       0.019    0.538    0.274    0.134
UONET               0                  0.015    0.536    0.318    0.200
MIT-GATEWAYS        STANFORD           0.032    0.477    0.245    0.073
ONENET-AS-1         NSFNETTEST14-AS    0.140    0.451    0.263    0.159
UONET               PENN-STATE         0.019    0.439    0.200    0.063
NOAA-AS             NOAA-FSL           0.018    0.438    0.255    0.031
DENET               UONET              0.032    0.410    0.189    0.188
NSFNETTEST14-AS     UC-DOM             0.022    0.365    0.244    0.081
ITALY-AS            UONET              0.016    0.358    0.228    0.117
NSFNETTEST14-AS     CONCERT            0.322    0.349    0.335    0.228
UONET               ITALY-AS           0.022    0.349    0.210    0.130
```

flow-dscan

- DoS detection / network scanning tool.
- Flag hosts which have flows to many other hosts.
- Flag hosts which are using a large number of TCP/UDP ports.
- Works better on smaller networks or with flow-filter to limit traffic. For example filter TCP port 25 to detect hosts infected with e-mail worm.

flow-gen

- Debugging tool to generate flows.

```
eng1:% flow-gen -V8.1 | flow-print | head -10

srcAS  dstAS  in     out    flows       octets      packets   duration
0      65280  0      65280  2           1           1         4294901760
1      65281  1      65281  4           2           2         4294901760
2      65282  2      65282  6           3           3         4294901760
3      65283  3      65283  8           4           4         4294901760
4      65284  4      65284  10          5           5         4294901760
5      65285  5      65285  12          6           6         4294901760
6      65286  6      65286  14          7           7         4294901760
7      65287  7      65287  16          8           8         4294901760
8      65288  8      65288  18          9           9         4294901760
```

flow-send

- Transmit flow files with NetFlow protocol to another collector.
- Can be used to take flow-tools files and send them to other NetFlow compatible collector.

flow-receive

- Like flow-capture but does not manage disk space. Output is to standard out and can be used directly with other flow-tools programs.
- Typically used for debugging.

```
eng1:% flow-receive 0/0/5555 | flow-print
flow-receive: New exporter: time=1011652474 src_ip=199.18.112.114
 dst_ip=199.18.97.102 d_version=8
srcPrefix           srcAS  dstPrefix           dstAS  input  output flows
143.105/16          600    128.9/16            4      48     25     1
140.141/16          600    150.216/16          81     48     25     4
132.235/16          17135  130.49/17           4130   38     25     25
131.123/16          11050  129.59/16           7212   42     25     1
206.21/16           600    128.239/16          11975  48     25     2
199.218/16          600    128.255/16          3676   48     25     1
```

flow-import

- Import flows from other formats into flow-tools.
- Currently supports ASCII and cflowd formats.

flow-export

- Export flows from flow-tools files to other formats.
- Currently supports ASCII and cflowd formats.
- ASCII output can be used with perl or other scripting languages (with a performance penalty).

flow-xlate

- Translate flows among NetFlow versions.
- Originally intended for use with Catalyst switches since they export some flows in version 7 and others in version 5 format.

Front End applications

- Flow-tools is good at collecting raw flows
- You may need additional tools to generate customized reports
- Perl applications are very popular.
  - flowscan.pm
  - Cflow.pm
  - CuGrapher.pl
- Integration with RRDTool, MRTG etc. makes it more useful

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.html
  - Requirements for IP Flow Information Export (RFC 3917)
  - 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/netflowapps.php
- Netflow HOW-TO http://www.linuxgeek.org/netflow-howto.php
- IETF standards effort: http://ipfix.doit.wisc.edu

References

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

More Info

- e-mail : gaurab @ lahai.com
- Labs and instruction on configuration how to configure Flow-tools, and a few more front end applications are available at

- On the web : http://lahai.com/netmgmt/

Acknowledgements

- Danny McPherson, Arbor
- Bruce Morgan, AARNet