



# Network Monitoring and Management

## NetFlow Overview



These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc/3.0/>)

# Agenda

## 1. Netflow

- What it is and how it works
- Uses and applications

## 2. Generating and exporting flow records

## 3. Nfdump and Nfsen

- Architecture
- Usage

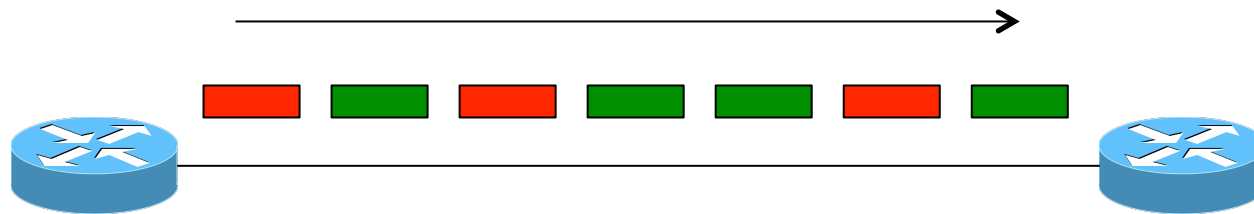
## 4. Lab



# What is a Network Flow?

- A set of related packets
- Packets that belong to the same transport connection. e.g.
  - TCP, same src IP, src port, dst IP, dst port
  - UDP, same src IP, src port, dst IP, dst port
  - Some tools consider "bidirectional flows", i.e. A->B and B->A as part of the same flow

[http://en.wikipedia.org/wiki/Traffic\\_flow\\_\(computer\\_networking\)](http://en.wikipedia.org/wiki/Traffic_flow_(computer_networking))

# Simple flows



-  = Packet belonging to flow X
-  = Packet belonging to flow Y

# Cisco IOS Definition of a Flow

## **Unidirectional sequence of packets sharing:**

1. Source IP address
2. Destination IP address
3. Source port for UDP or TCP, 0 for other protocols
4. Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
5. IP protocol
6. Ingress interface (SNMP ifIndex)
7. IP Type of Service

# IOS: which of these six packets are in the same flows?

	<i>Src IP</i>	<i>Dst IP</i>	<i>Protocol</i>	<i>Src Port</i>	<i>Dst Port</i>
A	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
B	5.6.7.8	1.2.3.4	6 (TCP)	80	4001
C	1.2.3.4	5.6.7.8	6 (TCP)	4002	80
D	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
E	1.2.3.4	8.8.8.8	17 (UDP)	65432	53
F	8.8.8.8	1.2.3.4	17 (UDP)	53	65432

# IOS: which of these six packets are in the same flows?

	<i>Src IP</i>	<i>Dst IP</i>	<i>Protocol</i>	<i>Src Port</i>	<i>Dst Port</i>
<b>A</b>	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
<b>B</b>	5.6.7.8	1.2.3.4	6 (TCP)	80	4001
<b>C</b>	1.2.3.4	5.6.7.8	6 (TCP)	4002	80
<b>D</b>	1.2.3.4	5.6.7.8	6 (TCP)	4001	80
<b>E</b>	1.2.3.4	8.8.8.8	17 (UDP)	65432	53
<b>F</b>	8.8.8.8	1.2.3.4	17 (UDP)	53	65432

*What about packets “C” and “D”?*

# Flow Accounting

A summary of all the packets seen in a flow (so far):

- Flow identification: protocol, src/dst IP/port...
- Packet count
- Byte count
- Start and end times
- Maybe additional info, e.g. AS numbers, netmasks

Records traffic volume and type but  
*not content*



# Uses and Applications

## **You can answer questions like:**

- Which user / department has been uploading / downloading the most?
- Which are the most commonly-used protocols on my network?
- Which devices are sending the most SMTP traffic, and to where?
- Identification of anomalies and attacks
- More fine-grained visualisation (graphing) than can be done at the interface level

# Working with flows

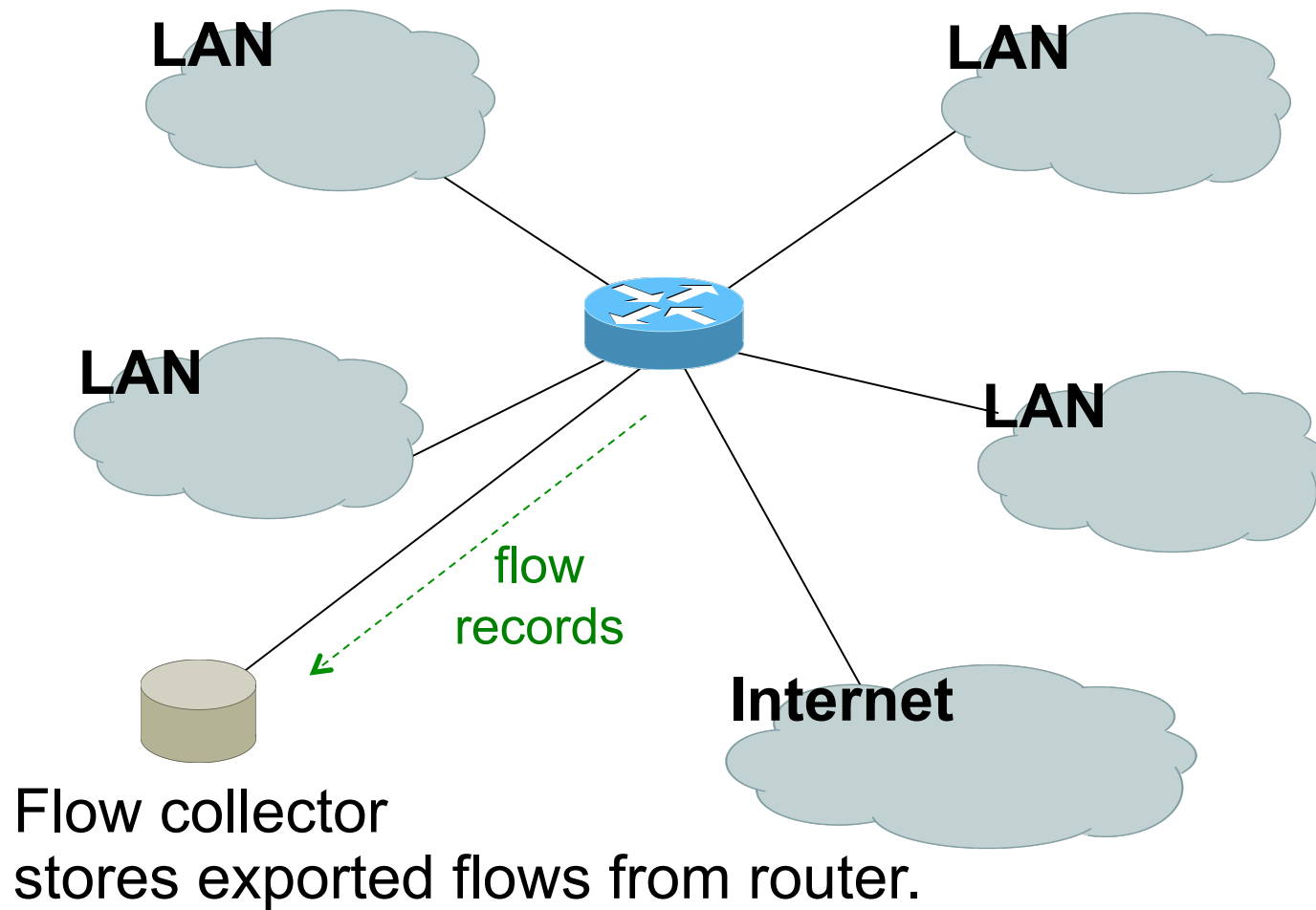
1. Configure device (e.g. router) to generate flow accounting records
2. Export the flows from the device (router) to a collector (PC)
  - Configure protocol version and destination
3. Receive the flows, write them to disk
4. Analyse the flows

Many tools available, both free and commercial

# Where to generate flow records

1. On a router or other network device
  - If the device supports it
  - No additional hardware required
  - Might have some impact on performance
2. Passive collector (usually a Unix host)
  - Receives a copy of every packet and generates flows
  - Requires a mirror port
  - Resource intensive

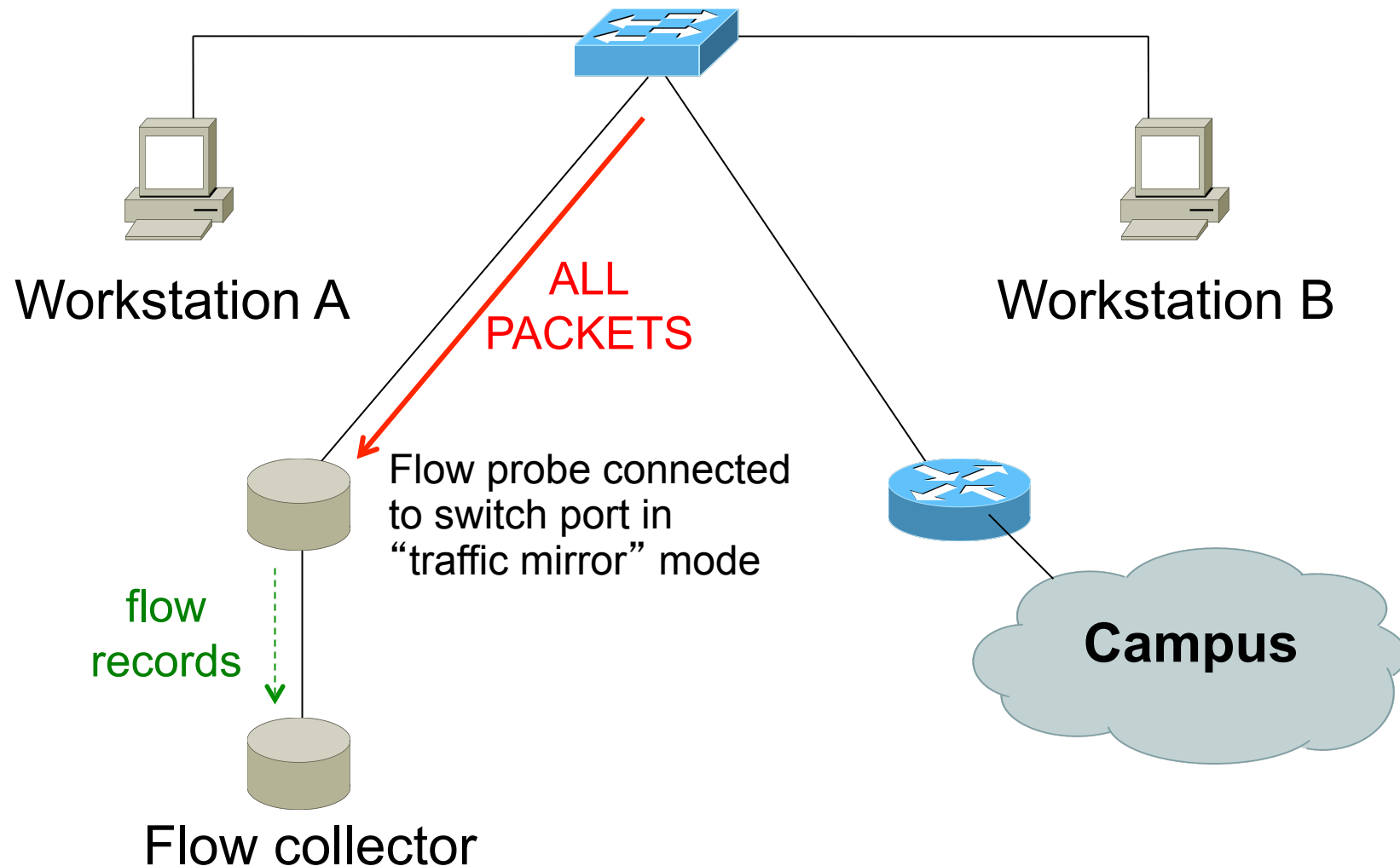
# Router Collection



# Router Collection

- All flows through router can be observed
- Router overhead to process & export flows
- Can select which interfaces Netflow collection is needed and not activate it on others
- If router on each LAN, Netflow can be activated on them to reduce load on core router

# Passive Monitor Collection



# Passive Collector

## Examples

- softflowd (Linux/BSD)
- pfflowd (BSD)
- ng\_netflow (BSD)
- Collector sees all traffic through the network point it is connected on and generates flows
- Relieves router from processing traffic, creating flows and exporting them

# Passive Collector cont.

## Useful on links:

- with only one entry into the network
- where only flows from one section of the network are needed

Can be deployed in conjunction with an IDS



## **A thought:**

Your network probably already has a device which is keeping track of IP addresses and port numbers of traffic flowing through it.

What is it?

# Flow Export Protocols

- Cisco Netflow, different versions
  - v5: widely deployed
  - v9: newer, extensible, includes IPv6 support
- IP Flow Information Export (**IPFIX**):
  - IETF standard, based on Netflow v9
- **sFlow**: Sampling-based, commonly found on switches
- **jFlow**: Juniper
- We use Netflow, but many tools support multiple protocols

# Cisco Netflow

- Unidirectional flows
- IPv4 unicast and multicast
  - (IPv6 in Netflow v9)
- Flows exported via UDP
  - Choose a port. No particular standard, although 2055 and 9996 are commonly used
- Supported on IOS, ASA and CatOS platforms - but with different implementations

# Cisco IOS Configuration

- Configured on each input interface
  - modern IOS allows both input *and* output
- Define the version
- Define the IP address and port 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

# Configuring Netflow: the old way

## Enable CEF

- `ip cef`
- `ipv6 cef`

## Enable flow on each interface

`ip route cache flow`

OR

`ip flow ingress`

`ip flow egress`

## Exporting Flows to a collector

```
ip flow-export version [5|9] [origin-as|peer-as]  
ip flow-export destination <x.x.x.x> <udp-port>
```

# **"Flexible Netflow": the new way**

- Only way to monitor IPv6 flows on modern IOS
- Start using it now – IPv6 is coming / here
- Many mind-boggling options available, but basic configuration is straightforward

# Flexible netflow configuration

- Define one or more exporters

```
flow exporter EXPORTER-1
  destination 192.0.2.99
  transport udp 9996
  source Loopback0
  template data timeout 300
```

- Define one or more flow monitors

```
flow monitor FLOW-MONITOR-V4
  exporter EXPORTER-1
  cache timeout active 300
  record netflow ipv4 original-input
flow monitor FLOW-MONITOR-V6
  exporter EXPORTER-1
  cache timeout active 300
  record netflow ipv6 original-input
```

# Flexible netflow configuration

## Apply flow monitors to interface

```
interface GigabitEthernet0/0/0
  ip flow monitor FLOW-MONITOR-V4 input
  ip flow monitor FLOW-MONITOR-V4 output
  ipv6 flow monitor FLOW-MONITOR-V6 input
  ipv6 flow monitor FLOW-MONITOR-V6 output
```



# "Top-talkers"

**You can summarize flows directly on the router, e.g.**

```
show flow monitor FLOW-MONITOR-V4  
cache aggregate ipv4 source address  
ipv4 destination address sort  
counter bytes top 20
```

**Yes, that's one long command!**

Old command "show ip flow top-talkers" sadly gone, but you could make an alias

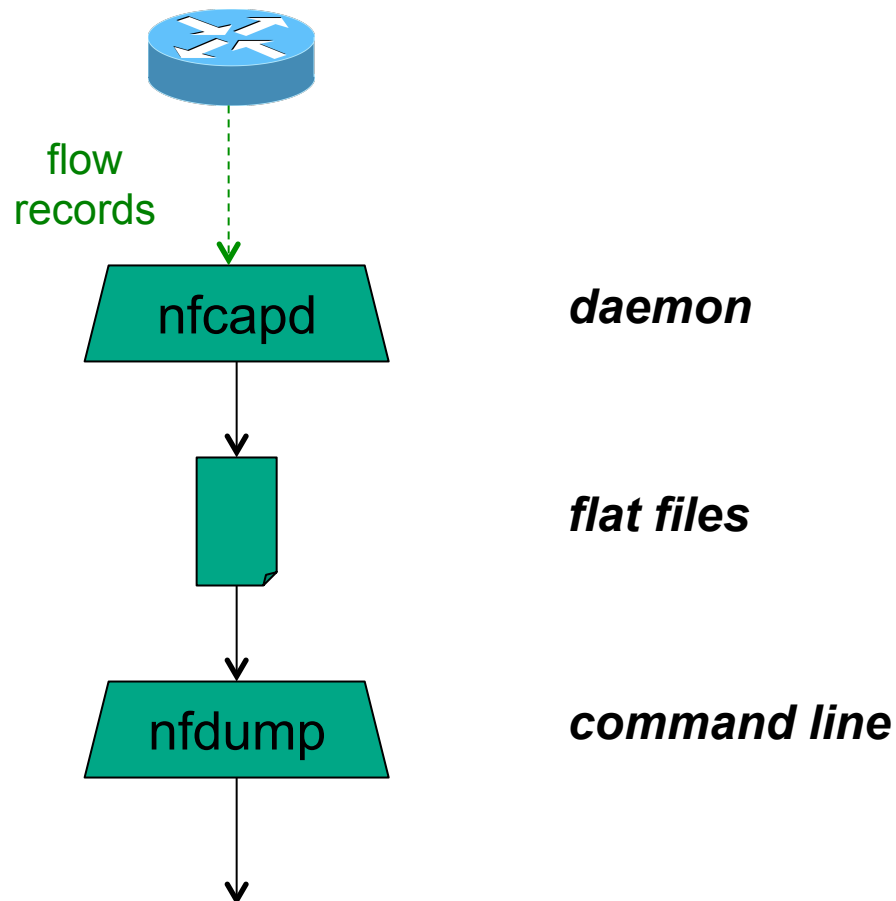
- `conf t`
- `alias exec top-talkers show flow..`

# Questions?

# Collecting flows: nfdump

- Free and open source – Runs on collector
- *nfcapd* listens for incoming flow records and writes them to disk (flat files)
  - typically starts a new file every 5 minutes
- *nfdump* reads the files and turns them into human-readable output
- nfdump has command-line options to filter and aggregate the flows

# nfdump architecture

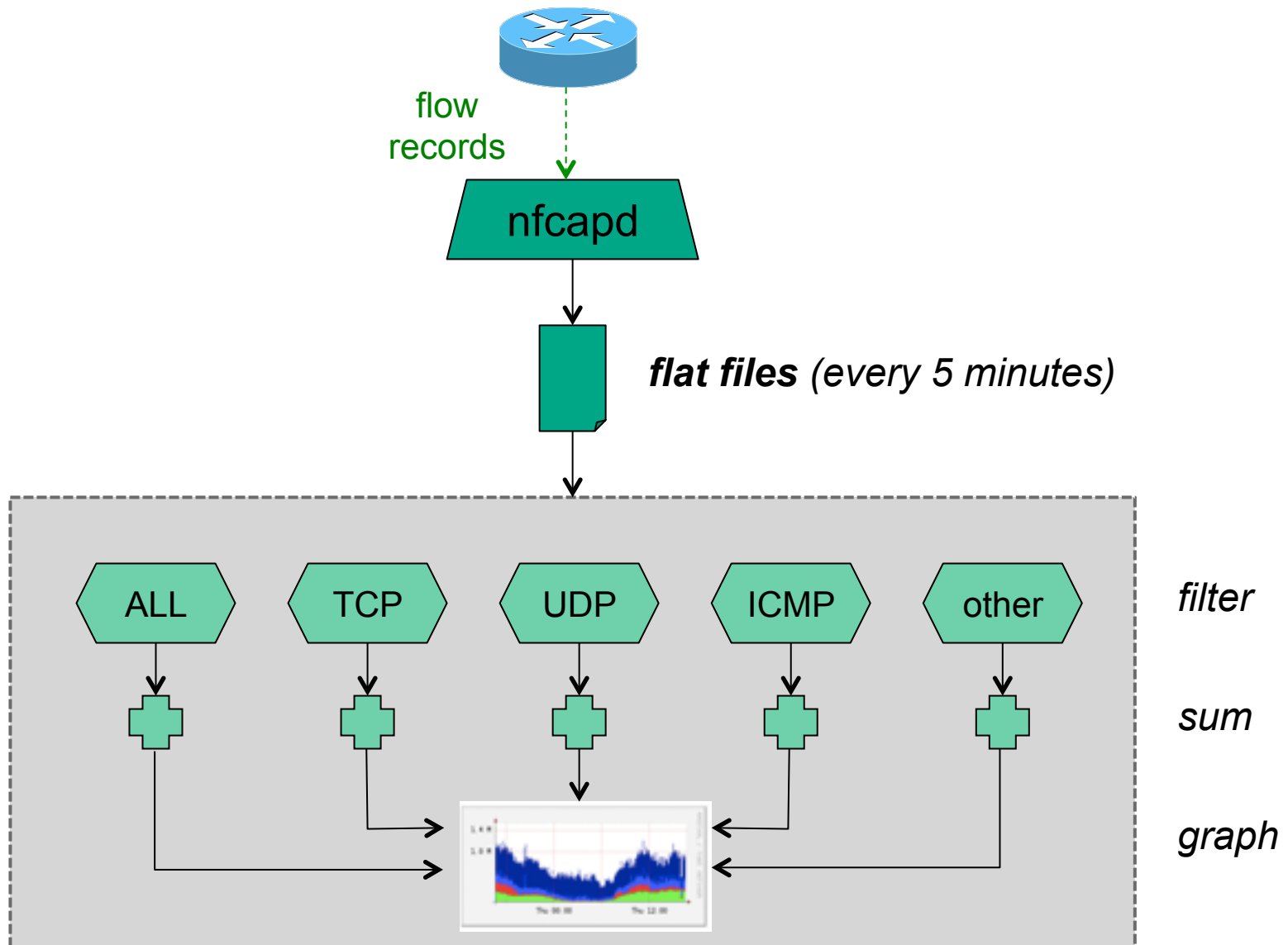


Date	flow start	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2013-04-18	13:35:23.353	1482.000	UDP	10.10.0.119:55555	->	190.83.150.177:54597	8683	445259	1
2013-04-18	13:35:23.353	1482.000	UDP	190.83.150.177:54597	->	10.10.0.119:55555	8012	11.1 M	1
2013-04-18	13:48:21.353	704.000	TCP	196.38.180.96:6112	->	10.10.0.119:62099	83	20326	1
2013-04-18	13:48:21.353	704.000	TCP	10.10.0.119:62099	->	196.38.180.96:6112	105	5085	1

# Analysing flows: nfsen

- Companion to nfdump
- Web GUI
- Creates RRD graphs of traffic totals
- Lets you zoom in to a time of interest and do nfdump analysis
- Manages nfcapd instances for you
  - Can run multiple nfcapd instances for listening to flows from multiple routers
- Plugins available like port tracker, surfmap

# nfsen architecture



## nfsen: points to note

- Every 5 minutes nfcapd starts a new file, and nfsen processes the previous one
- Hence each graph point covers 5 minutes
- The graph shows you the *total* of selected traffic in that 5-minute period
- To get more detailed information on the individual flows in that period, the GUI lets you drill down using nfdump

# Demonstration

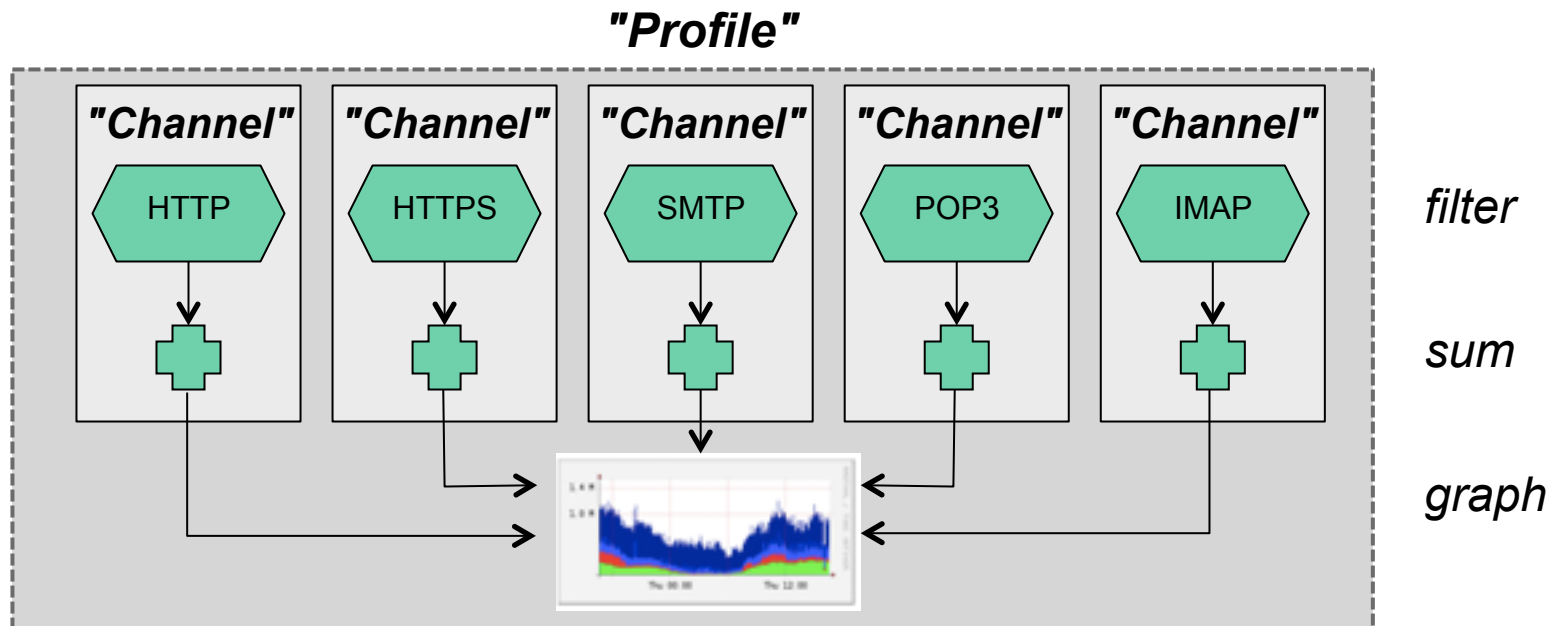
Now we will use nfsen to find biggest users of bandwidth



# Profiles and Channels

- A "channel" identifies a type of traffic to graph, and a "profile" is a collection of channels which can be shown together
- You can create your own profiles and channels, and hence graphs. e.g.
  - Total HTTP, HTTPS, SMTP traffic (etc)
  - Traffic to and from the Science department
  - ...
- Use filters to define the traffic of interest

# Profiles and Channels



# References – Tools

nfdump and nfsen:

<http://nfdump.sourceforge.net/>

<http://nfsen.sourceforge.net/>

<http://nfsen-plugins.sourceforge.net/>

pmacct and pmgraph:

<http://www.pmacct.net/>

<http://www.aplivate.org/pmgraph/>

flow-tools:

<http://www.splintered.net/sw/flow-tools>

# References – Further Info

- Wikipedia:  
<http://en.wikipedia.org/wiki/Netflow>
- IETF standards effort:  
<http://www.ietf.org/html.charters/ipfix-charter.html>
- Abilene NetFlow page  
<http://abilene-netflow.itec.oar.net/>
- Cisco Centric Open Source Community  
<http://cosi-nms.sourceforge.net/related.html>
- Cisco NetFlow Collector User Guide  
[http://www.cisco.com/en/US/docs/net\\_mgmt/netflow\\_collection\\_engine/6.0/tier\\_one/user/guide/user.html](http://www.cisco.com/en/US/docs/net_mgmt/netflow_collection_engine/6.0/tier_one/user/guide/user.html)

# **The end**

(Additional reference materials follow)

# Filter examples

any

*all traffic*

proto tcp

*only TCP traffic*

dst host 1.2.3.4

*only traffic to 1.2.3.4*

dst net 10.10.1.0/24

*only traffic to that range*

not dst net 10.10.1.0/24

*only traffic not to that range*

proto tcp and src port 80

*only TCP with source port 80*

dst net 10.10.1.0/24 or dst net 10.10.2.0/24

*only traffic to those nets*

dst net 10.10.1.0/24 and proto tcp and src port 80

*only HTTP response traffic to that net*

(dst net 10.10.1.0/24 or dst net 10.10.2.0/24) and proto tcp and src port 80

*...more complex combinations possible*



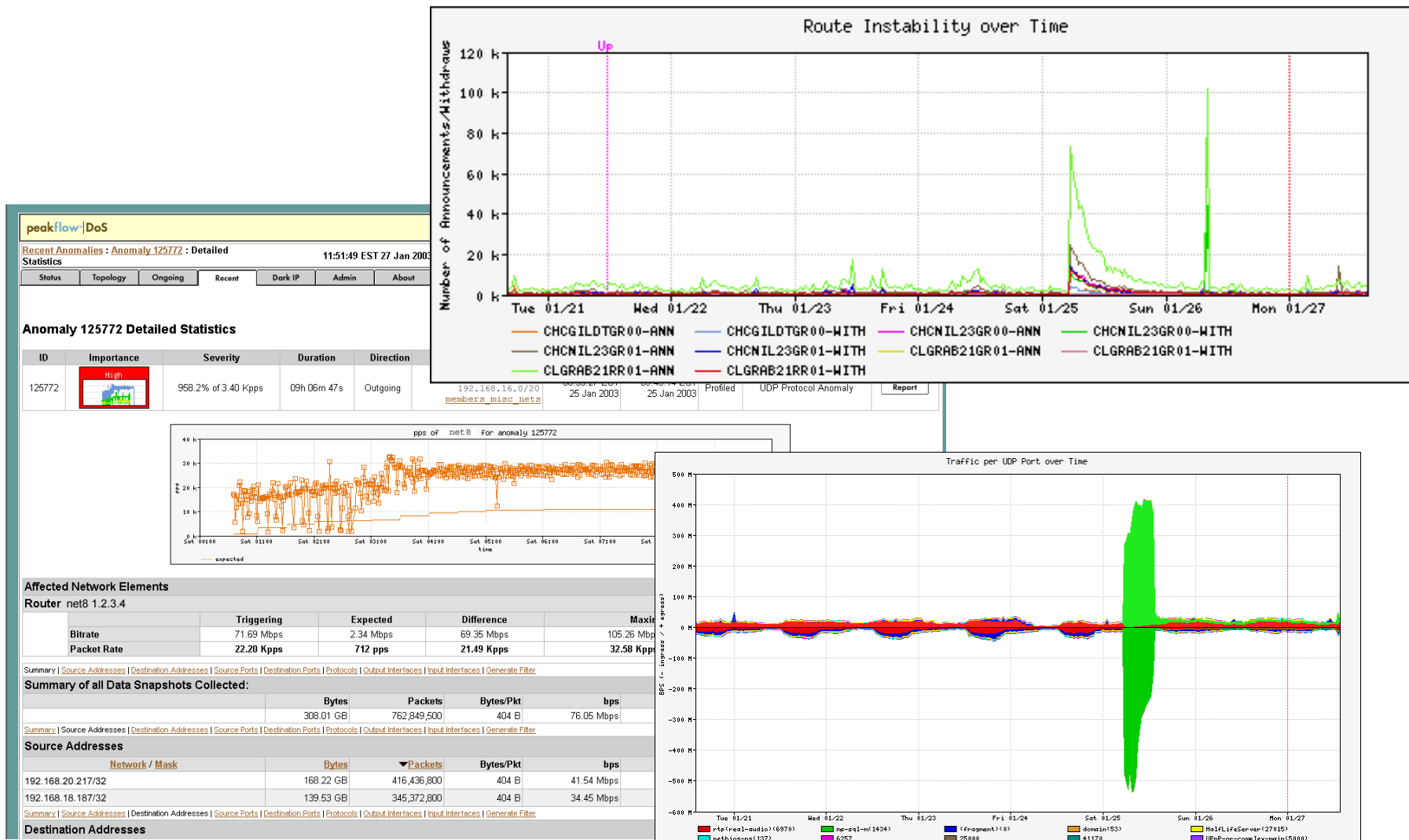
# Flows and Applications: More Examples

# Uses for NetFlow

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



# Detect Anomalous Events: SQL “Slammer” Worm\*

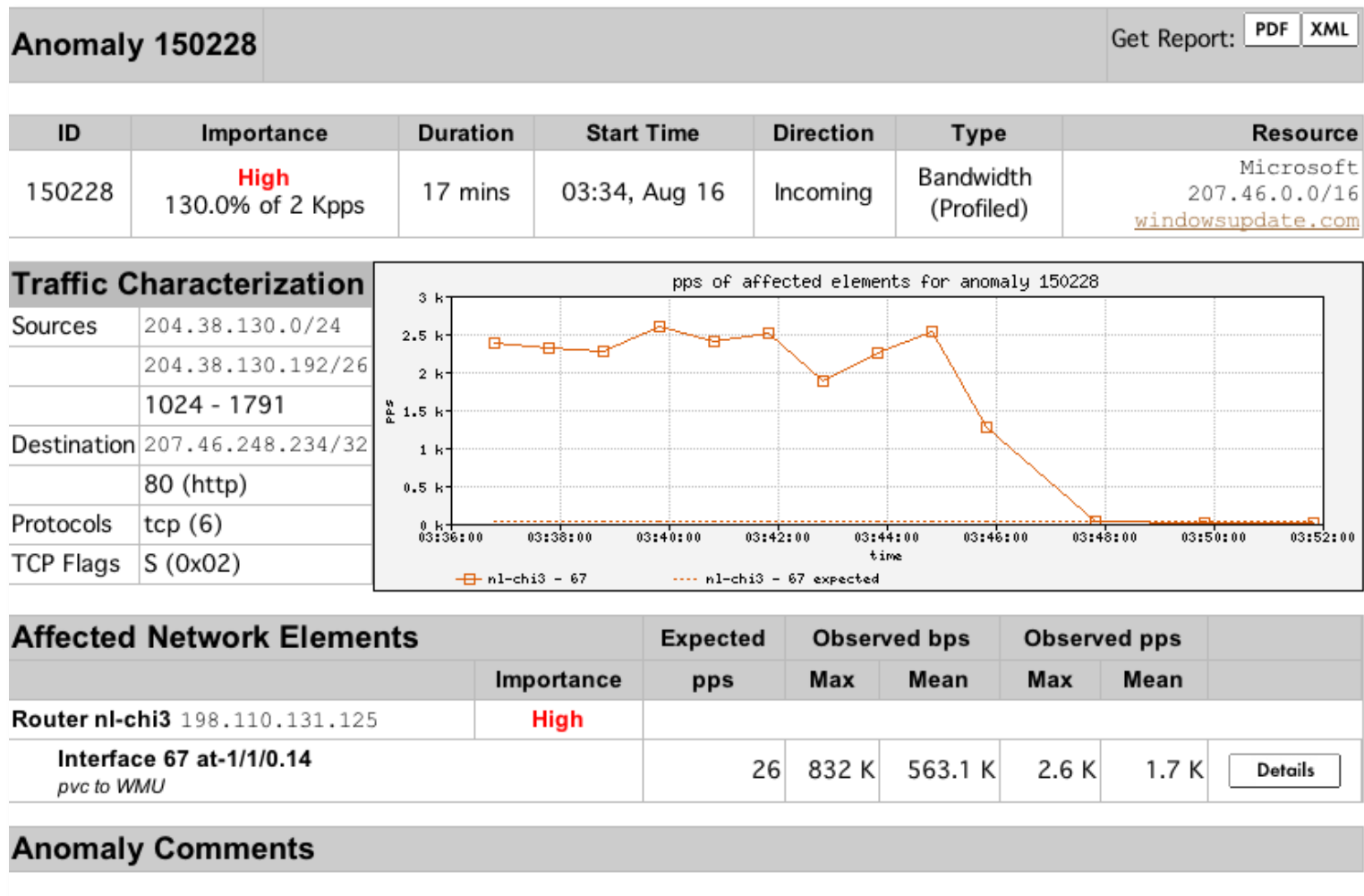


# Flow-based Detection (cont)\*

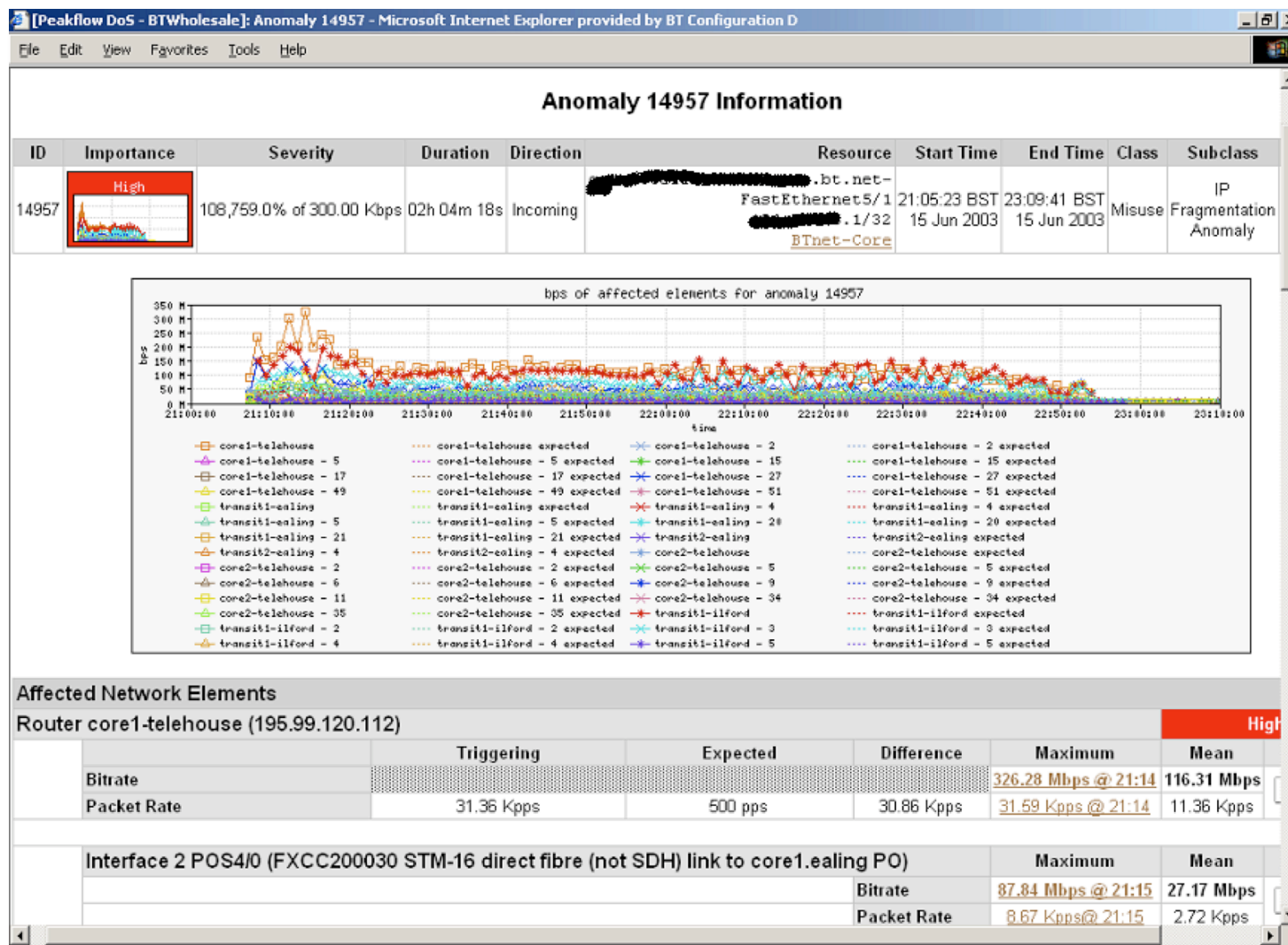
Once baselines are built anomalous activity can be detected

- 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...\*

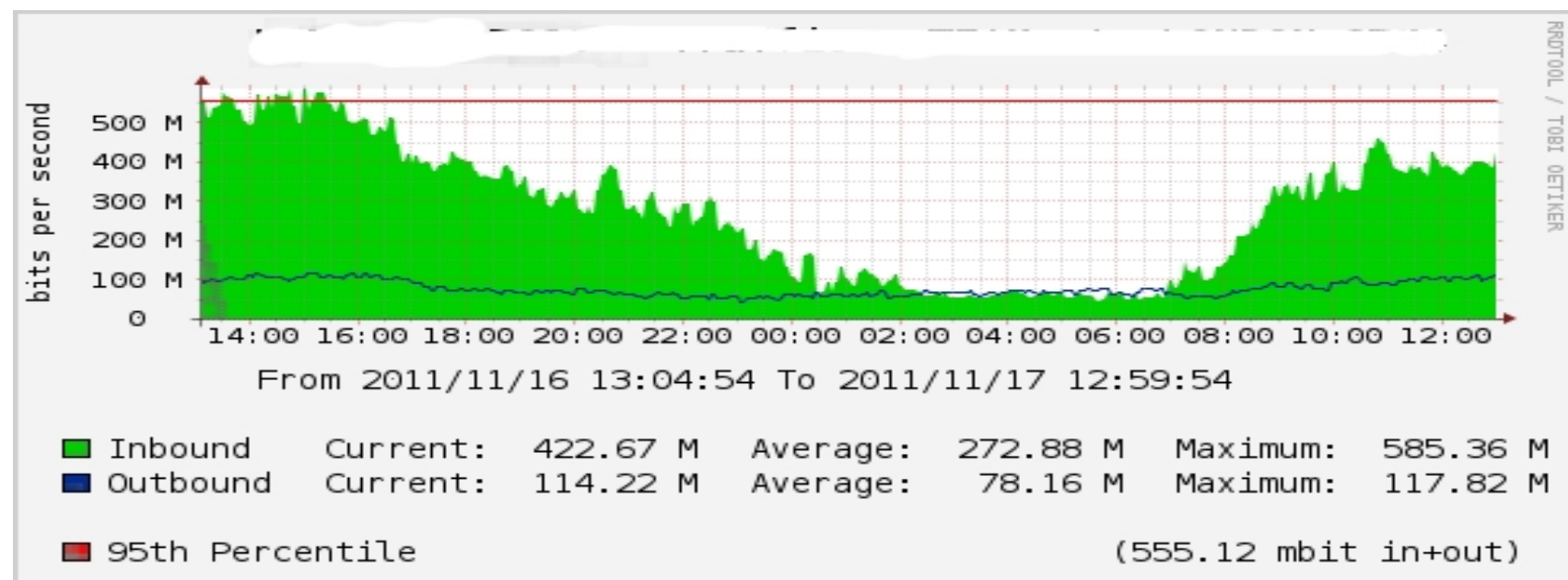
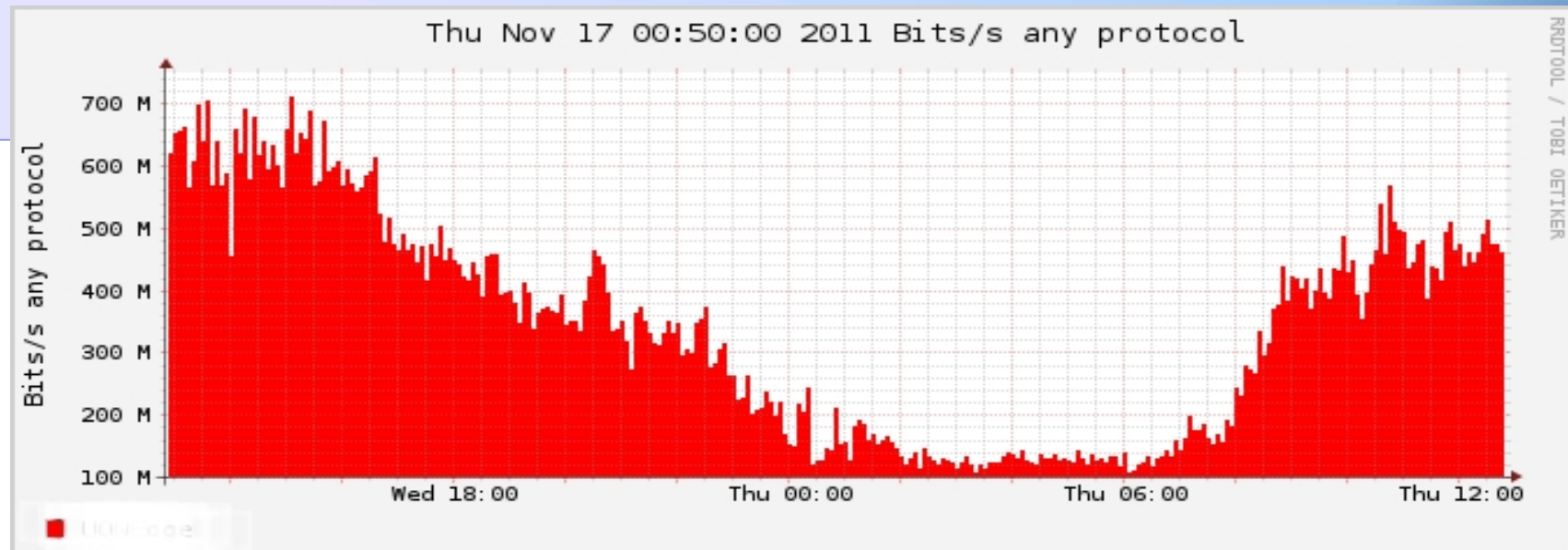


# Commercial Detection: A Large Scale DOS Attack



# Accounting

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



A decorative graphic consisting of a vertical blue bar on the left and a horizontal blue bar at the top, both with a gradient from light to dark blue.

# Cisco Netflow Versions

# NetFlow Version 1

- 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.
- Does not have sequence numbers – no way to detect lost flows
- Obsolete



# NetFlow Versions 2-4

- Cisco internal
- Were never released

# 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.
- IPv4 only

# 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 v9

- IPv6 support
- Additional fields like MPLS labels
- Builds on earlier versions
- Periodically sends "template" packet, all flow data fields reference the template