

# INASP: Effective Network Management Workshops

## Unit 7: Network Monitoring

### About these workshops

Authors:

- Dick Elleray, AfriConnect
  - [delleray@africonnect.com](mailto:delleray@africonnect.com)
- Chris Wilson, Aptivate
  - [chris + inaspbmo2013@aptivate.org](mailto:chris+inaspbmo2013@aptivate.org)

Date: 2013-04-29

### Objectives

On completion of this session, we hope you will know about:

- Importance of monitoring in network management
- What problems do we want to solve?
- What questions do we want to answer?
- How do we answer them?

### Why Monitor?

Do you have the information you need:

- Are getting what you paid for?
- Is it being used for the purpose intended?
- Is it being used efficiently?
- What will you need in future?
- Can you troubleshoot problems quickly?
- Can you enforce and improve the Acceptable Use Policy?

### Are you getting what you paid for?

How can you monitor this long-term?

We covered how to measure speed instantaneously in Unit 6. But you want a long-term solution, and manual speed tests take a lot of time. You can use [Nagios](#) to check for contention at your ISP:

- packet loss
- latency
- available bandwidth (speed test)

You will get false alarms when your own connection is congested, so:

- either eliminate that (with careful bandwidth management), or
- subtract the percentage of time when the local connection is the problem.

You may want to measure other providers (collaborate with other universities?) to see if they offer a better service.

Creating the Nagios service to measure bandwidth will require:

- An `iperf` server hosted on the Internet (don't know of a public one); or
- A public bandwidth testing server (maybe [Web100](#) with adapted client?)
- A large file downloaded with `wget` or `ab` (limited results)

TODO measure some data or use the GPRS benchmark data to show how to monitor and interpret contention and congestion.

## How is it being used?

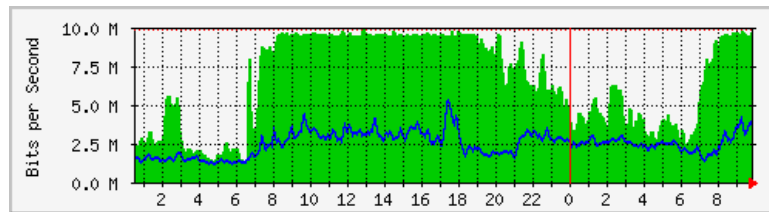
To answer these questions:

- Is it being used for the purpose intended?
- Is it being used efficiently?

We need to analyse the traffic on the link.

## Overall traffic level

A good indicator of network health is lack of congestion.



Is this link congested? When and for how long?

This graph was generated by [MRTG](#), collecting data from a router's SNMP counters or directly from the interfaces.

The characteristic *flat top* between 0800 and 1900 on the graph indicates that the network is fully utilised, which means it's very likely to be congested. The graph cannot prove this, because it doesn't measure latency, packet loss or available bandwidth (the signs of congestion; see *Unit 6/Is my network congested?*).

Why is it likely to be congested? Normally a flat top means that at least one TCP connection is running, since TCP tries to use all available bandwidth. TCP keeps increasing its bandwidth use until packets start dropping. This normally happens when the queue is full, so TCP normally causes congestion.

In practice there were probably hundreds or thousands of TCP flows active between these times. The total amount downloaded is 10 Mbps ( $\sim 1.25$  MBps) for 11 hours, which is  $\sim 50$  GB. It's unlikely that one person was downloading a single 50 GB file for the entire day!

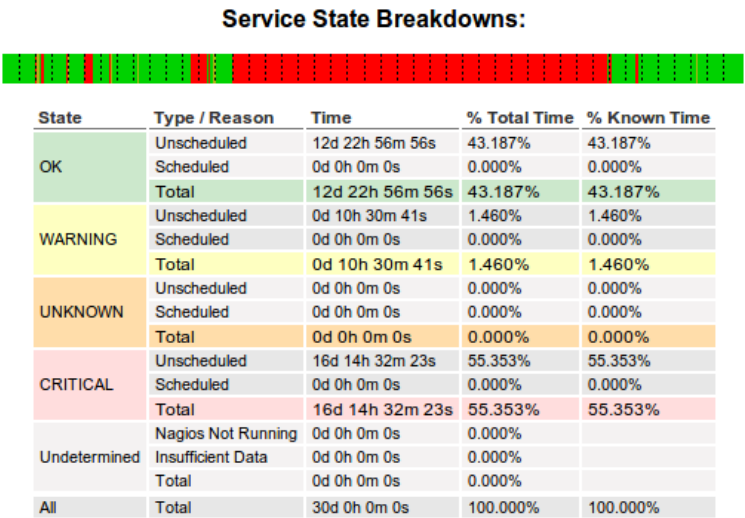
## Group Discussion

What is your experience of congestion?

- Scope
- Regularity
- Impact
- Recoverability
- Prevention

If you are working in a group, please share your experience with the others. If you are studying alone, please have a look at your network monitoring systems, and try to identify congestion. Is your network congested right now?

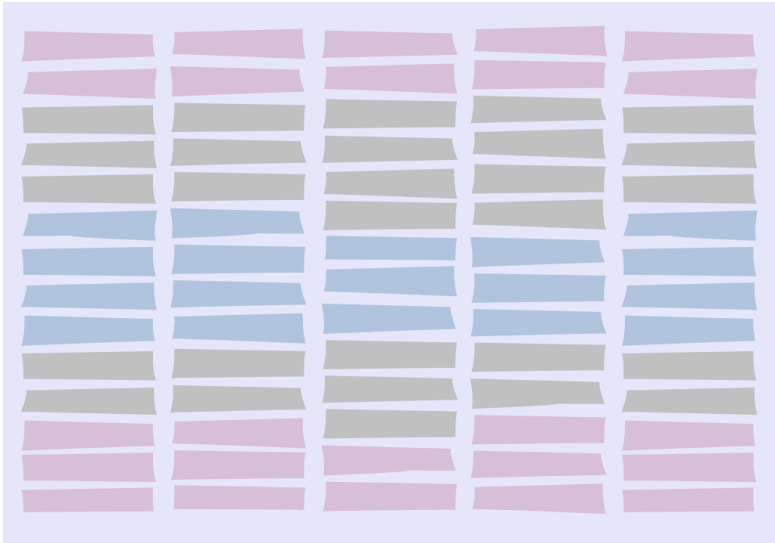
# Long-term congestion reporting



*Nagios can generate congestion reports if properly configured.*

If you configure Nagios to monitor your link for congestion, you can generate reports showing how much of the time the network was congested for. Some information on this is provided in the previous unit.

## Traffic patterns



- What sort of content?
- Is size important ?
- Is quantity important?
- Is time important?
- Is it user or system traffic?
- Is it desirable traffic?

## Levels of analysis

From least to most detail:

- Total traffic volumes
- Top talkers
- Applications (by port number)
- Applications (by deep packet inspection/DPI)
- Websites (by DPI or proxy server)
- Traffic flows (Netflow etc)
- Individual packets (pcap, Wireshark)

**Top talkers** are PCs, servers, and users generating the highest volumes of network traffic based on IP or MAC address. If they are end users, they are people you might want to have a word with. If they are servers, then you need to know how much traffic is local (where there is plenty of bandwidth) and how much is going over the scarce Internet link.

Some **applications** can be identified by port number. For example, web sites and web services use HTTP, which is almost always on port 80 or 443, and thus easy to identify. Some applications such as *Skype* and *Bittorrent* use many different ports or try hard not to be detected and classified.

Because web traffic is so varied, you may need to investigate deeper into which individual websites and types of content are being transferred, to differentiate between preferred, commodity and undesirable traffic.

## Desirability of traffic

According to your Acceptable Use Policy (AUP), you should be able to classify each stream as:

- Preferred/prioritised/institutionally important
- Politically necessary/expedient
- “Best effort” commodity traffic
- Undesirable
- Forbidden

Examples might include, depending on your AUP:

### **Preferred/prioritised/institutionally important traffic**

Journals, PDFs, external access to locally hosted web sites and email.

### **Politically necessary/expedient traffic**

Email, external webmail, search. May be important but not time critical; email can acceptably be delayed a few minutes.

### **"Best effort" commodity traffic**

Web sites that are not specifically academic.

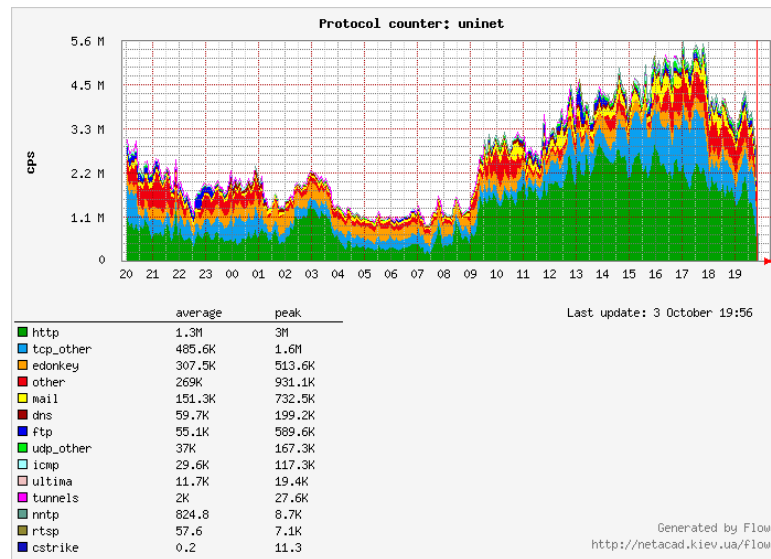
### **Undesirable traffic**

Videos, media downloads.

### **Forbidden traffic**

P2P, viruses, spam, pornography.

# Monitoring traffic types



In some cases you can identify whether the traffic is desirable just by protocol.

Which of these traffic classes are desirable?

This graph was collected by the [Flowc netflow collector](#), analysing data collected by a Cisco router and sent to a monitoring station using the [Netflow](#) protocol.

Flowc provides an overview of the traffic, but cannot help you to break down traffic by destination website (e.g. journals vs. YouTube) or local client. Skype and HTTP traffic are difficult to identify using Netflow. It also requires that you have a Cisco router, or another that can export Netflow data.

[Argus](#) and [NfSen](#) allow you to perform more detailed investigation into your network traffic, including *Top talkers* on your local network. However they are difficult to use.

[pmGraph](#) also allows such detailed analysis, and can collect data from a monitoring station that is not a Cisco router. It's also designed to be easy to use and powerful.

## Some commercial monitoring tools

There are [many tools](#) to be aware of, these are just a few:

- Agilent FireHunter
- Apparent Networks
- ixia IxChariot
- NetMon.ca
- Netscout Sniffer
- OPNET ACE
- PRTG
- Solar Winds
- Spirent SmartBits
- Various CISCO / 3COM / HP NMS tools

These are here just to show the range available, and to allow you after the course to research their capabilities. You can then compare these against your choice of no/low cost tools.

## Some free monitoring tools

- Argus

- BandwidthD
- Cacti
- Etherape
- Flowscan
- ifTop
- Iperf
- Microsoft Network Monitor (netmon)
- MRTG
- Munin
- Nagios
- NeDi
- nfSen
- Ngrep
- NMAP
- Ntop
- OpenNMS
- Snort
- tcpdump
- Wireshark

## Free vs. commercial tools

Advantages of commercial tools:

- usually more features
- usually easier to use

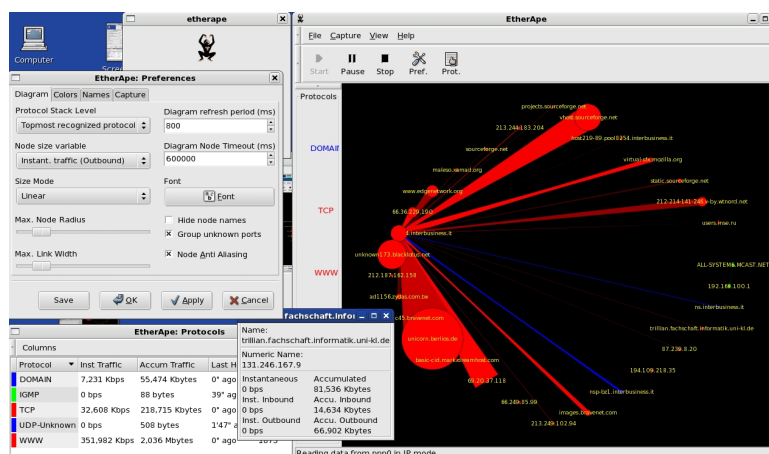
Disadvantages:

- (more) expensive
- proprietary lock-in

Some shops refuse to use free tools; some refuse to use commercial.

## Monitoring network hosts

[Etherape](#) is a graphical network monitor for Unix:

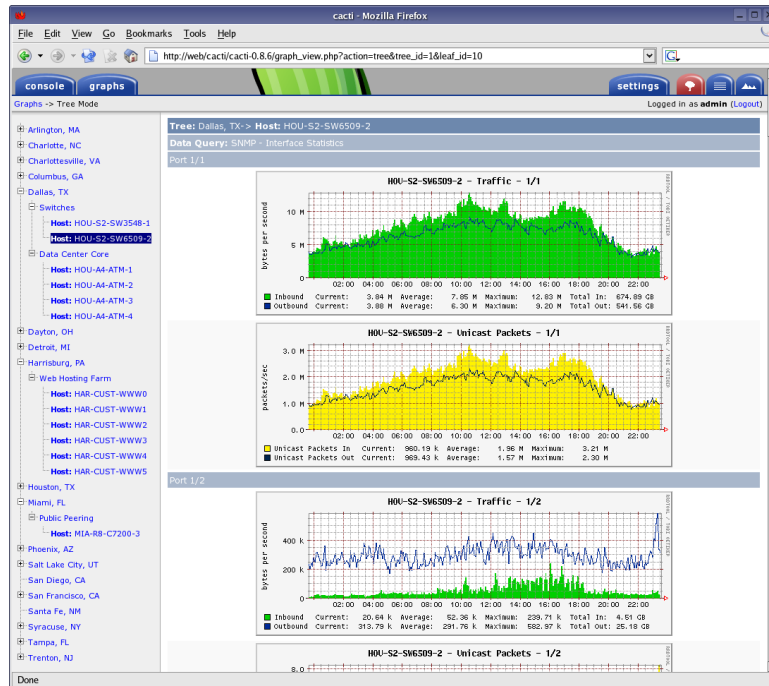


- Network traffic is displayed graphically
- 'Top Talkers' indicated visually
- Select protocol stack of focus
- Network filters
- View internal traffic, end to end IP, or port to port TCP

- Can read saved tcpdump file
- Many protocols supported

## Monitoring routers and switches

Cacti is an open source tool to monitor devices on the network via web browser.

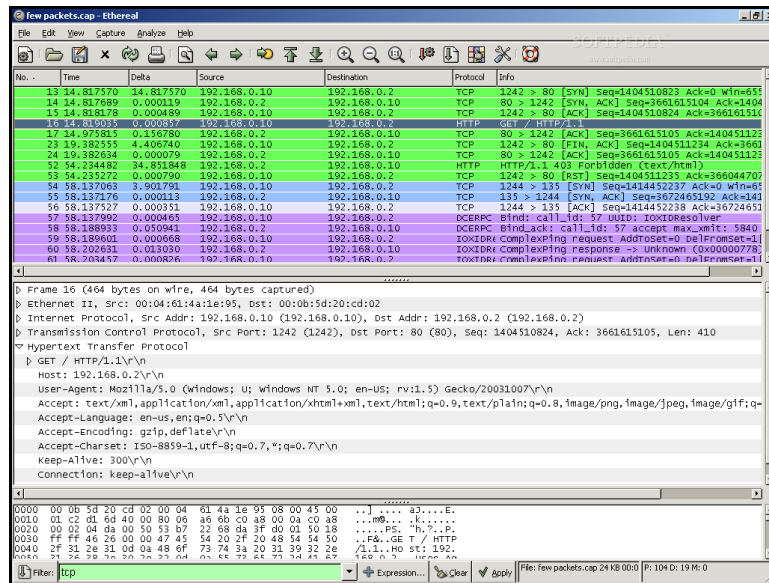


- Generates HTML with PNG reports
- Provides a live visual representation of historic traffic
- Allows monitoring and analysis of many data centre functions
- Collect network port, CPU, latency, utilization, temperature, etc. using SNMP or scripts
- On the fly ability to magnify interesting graphs

You can download more information, exercises and worksheets for installing and configuring Cacti on the [AfNOG SSE](#) course website, and many other places on the Internet.

## Packet level analysis

Wireshark is a great debugging tool.



It shows exactly what is happening on your network, packet by packet.

Wireshark is a *network protocol analyzer* (sniffer):

- Examine data from a live network
- Examine saved capture file
- Supports many capture formats
- Reasonably intuitive interface
- View reconstructed TCP sessions
- Filters and graphs (not very easy to use!)

Wireshark captures, displays and analyses every packet that it's configured to capture. Unfortunately this level of information can be overwhelming. On a busy network (say 10 Mbps, fully utilised) there can be 10,000 or more packets per second!

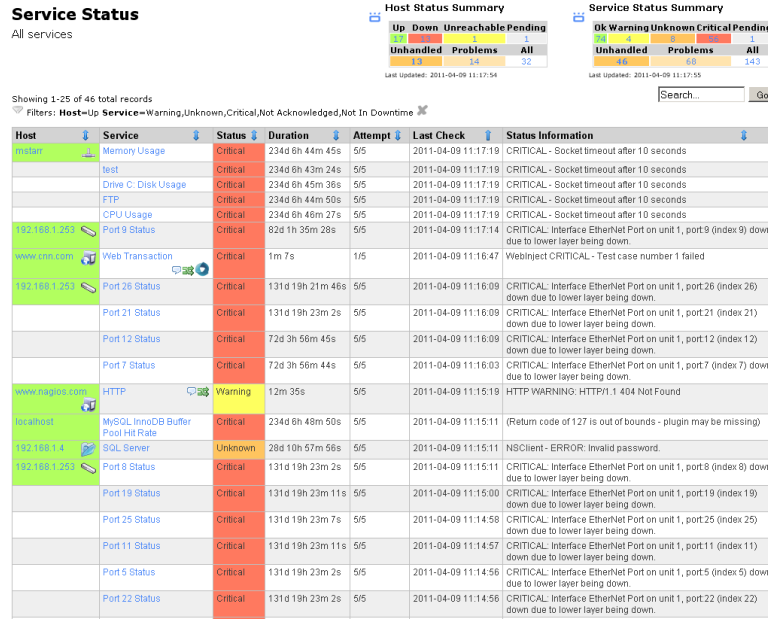
Wireshark includes filters, so you can restrict the capture to certain hosts or traffic types that interest you, and tools to break down traffic by protocol, although they can be slow. You can also investigate which website an individual connection is accessing, if you intercept the first packets of the conversation.

This volume of data (~1 MB per second, 84 GB per day) is too much to store and process in reasonable time, so this is only really useful as a real-time diagnostic tool.

## Service monitoring

[Nagios](#) is a network host and service monitor.





- Accessed via web browser
- Services (POP, PING, HTTP, etc)
- Host resources, Environmental factors
- Option of distributed monitoring
- Acknowledge issues via web interface
- Notification / event handlers
- Modular, allows for plug-ins

Nagios monitors hosts and services by regularly running *plugins* that check the status. There are hundreds (perhaps thousands) of plugins for common services, and it's easy to create new ones.

Nagios has a commercial version and several competitors, such as Zenoss, Zabbix, OpenNMS and Hyperic. The competitors may be easier to use, but Nagios is infinitely flexible, powerful and lightweight.

You can download more information, exercises and worksheets for installing and configuring Nagios on the [AfNOG SSE](#) course website, and many other places on the Internet.

## Network management framework

OpenNMS is a Network Management System framework.

Integrates “everything you need” for network management in one place.

Some people prefer a “one stop shop” that integrates many features in one place. (This is counter to the Unix philosophy, where you have many small, independent and flexible tools, making it easier to upgrade and replace just one of them).

OpenNMS includes:

### Action daemon

Automated actions (work flows)

### Collection daemon

Collects data from various sources (SNMP, plugins, etc)

### Capability daemon

Capability check on nodes (for uptime reporting and alerting)

### Discovery daemon

Initial and ongoing discovery of services on the network (reduces need for manual configuration)

**Events manager daemon**

Manages/stores events (changes in network and service status)

**Notification daemon**

External notification of users (sends emails, SMS, electric shocks, etc)

**Outage manager daemon**

Consolidates events (tries to avoid email storms during outages, when many services go down at one time)

**Poller daemon**

Polls managed nodes/services

**SNMP trap daemon**

Listens for SNMP traps (one of the biggest limitations of Nagios is that it doesn't have an SNMP trap listener)

**Threshold daemon**

Monitor for threshold values and generate events/alerts

OpenNMS is big, and therefore heavy, and takes a while to learn your way around.

## Group Discussion

If you are participating in a workshop, please discuss in groups:

- What sorts/aspects of traffic could be monitored?
- Why are those sorts/aspects of traffic monitoring important?
- Which does your institution monitor?
- What tools do you use? What works well or not?
- Have you found it of use? How and why?

When you have finished, please summarise your results to the other groups.

## Monitor/measure – Example

- Protocols
- Dropped packets
- IPs in use on LAN
- P2P
- Virus traffic
- Hackers spoofing
- SMTP (illegitimate mail)
- Usage (who, what)
- Applications using high bandwidth
- Movies
- Music
- telnet
- Voip/sip
- Microsoft ds
- Non business browsing
- Amount of bandwidth (per user if poss)