

Bandwidth Management

Chris Wilson
Aptivate Ltd, UK
AfNOG 2010



Ingredients

- **What is bandwidth management**
 - When to manage bandwidth
 - Troubleshooting an Internet connection
 - Monitoring an Internet connection
 - Setting policy
 - Enforcing Policy
 - Social measures
 - Technical measures
 - Summary and resources

Specific Questions

- Divide bandwidth between different networks on CentOS
- Reserving bandwidth for specific services
- Strategic Plan for Buying Bandwidth based on value and need

What is Bandwidth Management?

- Network management of slow links, and the networks that use them?
 - Do you have a better definition?
- Particularly important to internet users
- Users often complain that “the internet is slow” or “the internet is down”
- You may need more bandwidth, but:
 - Usage always grows until resource is not worth using
 - Bandwidth is very expensive
 - Good management can save you a lot of money



Meeting Expectations

- Users have an expectation of network performance
 - Set by previous experience, e.g. cyber cafés, friends, other employers, connection at home
- Users will ask for more bandwidth than you can supply (if it doesn't cost them more money)
- Business and academia don't provide “neutral pipes”
 - Subsidised service for specific objectives, e.g. research
- Maximise utility for the intended purposes
 - Reduce, eliminate or move all other traffic
 - Make the most capacity available



Bandwidth Mis-management

- If an internet connection is not well managed:
 - PCs will become infected with viruses and worms
 - Virus and worm traffic will fill the connection
 - P2P users and download managers will fight for the rest
 - Ordinary web browsing will become impossible
 - Skype, VoIP and other interactive applications will be unusable
- Departments may demand a separate connection
 - Wastes resources that could be better pooled
 - Appears to work for a while, then suffers the same fate



Next

- ✓ What is bandwidth management
- **When to manage bandwidth**
 - Troubleshooting an Internet connection
 - Monitoring an Internet connection
 - Setting policy
 - Enforcing Policy
 - Social measures
 - Technical measures
 - Summary and resources



When to Manage Bandwidth

- Do we need bandwidth management?
 - Users complaining (and bandwidth is definitely the issue)
 - Billed by usage
 - Throttled by usage
 - Complaints from upstream provider
 - Improve quality of service
 - Downgrade connection to save money
- Not sure? Monitor!
 - Management will not help unless link is overloaded
 - Monitoring gives early warning of problems



Next

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- **Troubleshooting an Internet connection**
 - Monitoring an Internet connection
 - Setting policy
 - Enforcing Policy
 - Social measures
 - Technical measures
 - Summary and resources

The Internet is so slow!

- What do we mean by “slow”?
 - completely down?
 - packet loss (tcp backoff)
 - long ping times (round-trip times)
 - long DNS lookup times (or DNS failure)
 - jitter (mostly affects Skype and other VoIP)
- What doesn't work?
 - Access to ordinary web pages? (HTTP, DNS)
 - BitTorrent and P2P software?
 - Skype and other real-time network applications?

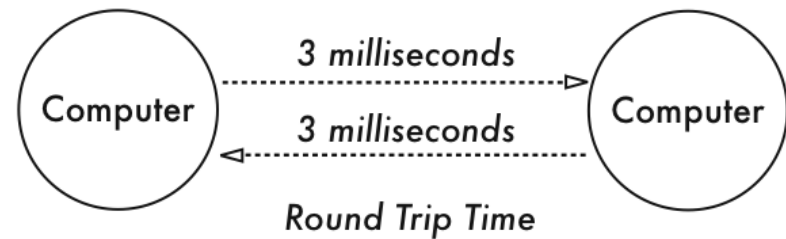
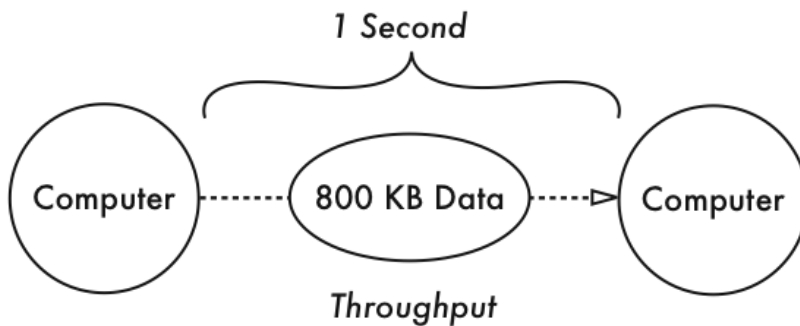
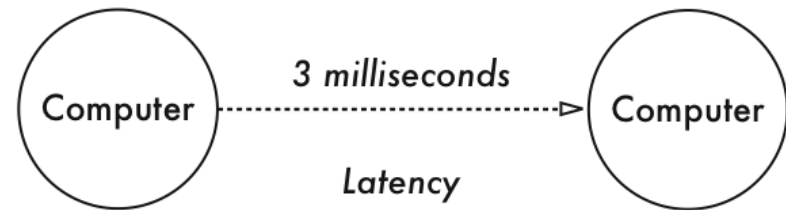
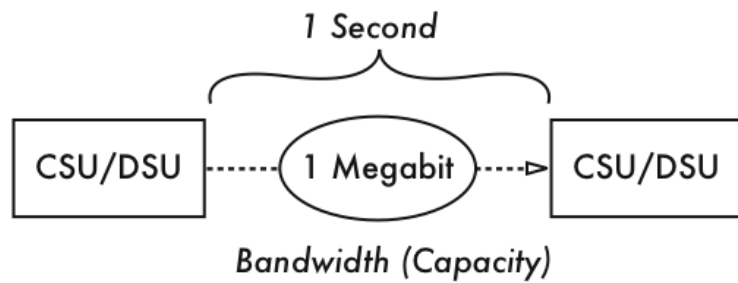


In Case of Repeated Fires

- Sometimes (not always!) the problem will be that your connection is too often full (used to capacity)
- You can ping the router on your side without problems, but pinging your ISP's router shows:
 - very high latency (over 1 second) to your ISP
 - Windows reports latency over 4 seconds as “request timed out”
 - packet loss over 1% to your ISP
 - DNS timeouts or slow replies from your ISP (not cached)
 - high jitter (subjective, maybe over 20 ms stdev?)
- Could also be a faulty link or router on either end



Definitions



Diagnosing the Problem

- Check that your connection works
- Check that your DNS works
- Traceroute to the remote server, looking for:
 - sudden increase in ping times or packet loss
 - jitter (standard deviation changes)
 - identify between which hops this occurs
- Ping the remote server
- *telnet www.youtube.com 80*
 - *GET / HTTP/1.0*
Host: www.youtube.com
- Monitor intermittent problems with trending tools



Ping

- Useful for spot checking:
 - reachability (try `www.google.com`, 4.2.2.2)
 - round trip time (RTT), also known as latency
 - packet loss (`ping -f`, `ping -c 1000 -s 1400` may help)
 - jitter (`ping -c 1000` and check *mdev/stddev*)
 - fragmentation (`ping -s 1483`)

Matt's Traceroute (MTR)

- Interactive, repeating version of Traceroute

- *sudo pkg_add -r mtr*

- *mtr -t download.java.sun.com*

- | HOST: | rocio.int.aidworld.org | Loss% | Snt | Last | Avg | Best | Wrst | StDev |
|-------|------------------------------|-------|-----|-------------|-------|-------|------|-------|
| 1. | 196.200.217.254 | 0.0% | 10 | 1.6 1.7 | 1.6 | 1.8 | 0.1 | |
| 2. | rtr-tedata.mtg.afnog.org | 0.0% | 10 | 2.0 2.2 | 2.0 | 3.2 | 0.4 | |
| 3. | host-196.219.220.81-static.t | 0.0% | 10 | 5.5 8.4 | 4.0 | 45.0 | 12.9 | |
| 4. | host-163.121.160.229.tedata. | 0.0% | 10 | 6.7 4.8 | 4.3 | 6.7 | 0.8 | |
| 5. | host-163.121.189.73.tedata.n | 0.0% | 10 | 4.4 11.3 | 4.4 | 63.4 | 18.4 | |
| 6. | host-163.121.186.253.tedata. | 0.0% | 10 | 4.5 5.1 | 4.5 | 7.4 | 0.9 | |
| 7. | host-163.121.184.61.tedata.n | 0.0% | 10 | 5.0 5.7 | 4.6 | 13.5 | 2.8 | |
| 8. | pal6-telecom-egypt-1-eg.pal. | 0.0% | 10 | 72.3 66.4 | 54.5 | 100.7 | 15.4 | |
| 9. | ash1-new11-racc1.ash.seabone | 0.0% | 10 | 150.3 154.2 | 150.3 | 175.9 | 7.8 | |
| 10. | ntt-1-ash1.ash.seabone.net | 40.0% | 10 | 153.7 152.7 | 146.7 | 154.5 | 3.0 | |
| 11. | as-3.r20.snjsca04.us.bb.gin. | 0.0% | 10 | 153.7 182.7 | 146.1 | 219.0 | 36.8 | |
| 12. | as-3.r20.snjsca04.us.bb.gin. | 10.0% | 10 | 215.9 255.3 | 214.3 | 370.0 | 54.4 | |
| 13. | ge-3-3.r03.snjsca04.us.ce.gi | 10.0% | 10 | 216.9 253.5 | 216.2 | 402.0 | 63.7 | |
| 14. | border2.te8-1-bbnet2.sfo002. | 10.0% | 10 | 216.9 218.7 | 215.8 | 230.7 | 5.0 | |
| 15. | border2.te8-1-bbnet2.sfo002. | 50.0% | 10 | 215.2 215.6 | 214.9 | 216.9 | 0.8 | |
| 16. | ??? | 100.0 | 10 | 0.0 0.0 | 0.0 | 0.0 | 0.0 | |



Who Controls the Broken Link

- Every link is between two hops
- May be able to identify them from reverse DNS, or looking at your network map
- Both ends are responsible for the link
- Usually cannot tell which end has the problem except by swapping it out
- Who controls the nearest end?
 - You? (investigate the traffic on the link)
 - Your ISP? (call your ISP)
 - Their carrier? (call your ISP, or pray)



Next

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- ✓ Troubleshooting an Internet connection
- **Monitoring an Internet connection**
 - Setting policy
 - Enforcing Policy: Social measures
 - Enforcing Policy: Technical measures
 - Summary and resources

Monitoring an Internet connection

- What do we want to monitor?
 - The same factors that we want to use for troubleshooting
 - The same factors that affect quality of service
 - Local and remote router availability and ping times (packet loss and latency)
 - Local and remote caching DNS server availability and query response times (failure rate and latency)
 - Link traffic overall, and by host and type
 - Remote websites (end-to-end test)
- Long-term monitoring helps to identify trends and sudden large changes



What Kind of Monitoring

- Spot check tools can identify some problems immediately
- Many problems require an idea of baseline performance (what changed? and how much?)
- Trending tools can gather baseline data
- Trending tools can help investigate problems after they disappear (e.g. intermittent, recurring)
- Trending tools require significant CPU, disk space, bandwidth and infrastructure investment



Tools of the Trade

| Variable | Spot Check | Trending |
|---------------------|-------------------------|-----------------------|
| End-to-end HTTP | wget, fetch, httpperf | Smokeping, Nagios |
| Ping latency | Ping, Traceroute, MTR | Smokeping, Nagios |
| Ping packet loss | Ping, Traceroute, MTR | Smokeping, Nagios |
| DNS latency | Host, Resperf | Smokeping, Nagios |
| DNS errors | Host, Resperf | Smokeping, Nagios |
| Total bandwidth use | Cisco “show interfaces” | Cacti, MRTG |
| Traffic flows | Cisco Top Talkers, Ntop | NfSen, Argus, pmGraph |
| Individual packets | Wireshark | tcpdump, Argus |



Quality of Service Monitoring

- Nagios to monitor websites, routers and DNS servers (local and upstream) and send alerts
- Cacti to monitor total bandwidth use on each interface, CPU and memory use on routers and switches
- Smokeping to monitor websites, latency and packet loss on upstream connections
- pmGraph to monitor traffic flows on Internet connections

Conventions

- File names and technical terms are in *italics*
- Commands to type are shown in monospaced bold italic purple type:
 - ***cat /etc/monospaced/bold/italic/purple***
- Long command lines are wrapped, but with a single bullet point at the start:
 - ***cat /usr/local/etc/foo/bar | less | more | grep | sed | awk
> /usr/local/tmp/foo/bar***
- Text that is output by a program, or should already be in a file, is shown in plain monospaced type:
 - `sshd_enable="YES"`



Installing Apache

- Install Apache binary package:
 - ***sudo pkg_add -r apache22***
 - You can ignore the message “pkg_add: apache-2.2.x is already installed”
- Edit */etc/rc.conf* and add the following line (if not already present):
 - ***apache22_enable=YES***
- Start Apache now:
 - ***/usr/local/etc/rc.d/apache22 start***
- Test that Apache is running



Installing Nagios (1)

- Install the Nagios binary package:
 - ***sudo pkg_add -r nagios***
- Edit */etc/rc.conf* and add the following line:
 - ***nagios_enable="YES"***
- Copy the sample files in */usr/local/etc/nagios* to their real names:
 - ***cd /usr/local/etc/nagios***
 - ***sudo cp nagios.cfg-sample nagios.cfg***
 - ***sudo cp cgi.cfg-sample cgi.cfg***
 - ***sudo cp resource.cfg-sample resource.cfg***



Installing Nagios (2)

- Edit *nagios.cfg* and comment out this line:
 - `cfg_file=/usr/local/etc/nagios/objects/localhost.cfg`
- Copy the sample files in */usr/local/etc/nagios/objects*:
 - **`sudo cp commands.cfg-sample commands.cfg`**
 - **`sudo cp contacts.cfg-sample contacts.cfg`**
 - **`sudo cp timeperiods.cfg-sample timeperiods.cfg`**
 - **`sudo cp templates.cfg-sample templates.cfg`**
- Edit */usr/local/etc/nagios/objects/contacts.cfg*:
 - Change `nagios@localhost` to your email address



Configuring Apache for Nagios (1)

- Create */usr/local/etc/apache22/Includes/nagios.conf* with the following contents:
 - `<Directory /usr/local/www/nagios>`
 - `Order deny,allow`
 - `Allow from all`
 - `AuthName "Nagios Access"`
 - `AuthType Basic`
 - `AuthUserFile /usr/local/etc/nagios/htpasswd.users`
 - `Require valid-user`
 - `</Directory>`
 - `<Directory /usr/local/www/nagios/cgi-bin>`
 - `Options ExecCGI`
 - `</Directory>`
 - `ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/`
 - `Alias /nagios/ /usr/local/www/nagios/`



Configuring Apache for Nagios (2)

- Create the password file and a user account for Nagios:
 - ***sudo htpasswd -c /usr/local/etc/nagios/htpasswd.users nagiosadmin***
 - At the “New Password:” prompt, enter the password you want for the nagiosadmin user
- Tell Apache to reload its configuration:
 - ***sudo /usr/local/etc/rc.d/apache22 reload***
- Test it by browsing to *http://localhost/nagios/*:
 - Log in as user nagiosadmin with the password you entered into *htpasswd*
 - You should see the Nagios logo and “Version 3.0.6”



Monitoring Routers with Nagios (1)

- Edit */usr/local/etc/nagios/objects/templates.cfg* and add these lines at the end:
 - ```
define host {
 host_name router-serena
 use generic-host
 address 196.200.215.254
 max_check_attempts 5
}
```
  - ```
define host {  
    host_name router-kist  
    use generic-host  
    address 196.200.217.254  
    max_check_attempts 5  
}
```
- (continued...)



Monitoring Routers with Nagios

- Edit */usr/local/etc/nagios/objects/templates.cfg* and add these lines at the end:
 - ```
define hostgroup {
 hostgroup_name routers
 members router-serena, router-kist
}
```
  - ```
define service {  
    service_description ping  
    use generic-service  
    hostgroup routers  
    check_command check_ping!10,20%!20,40%  
}
```



Monitoring DNS Servers with Nagios

- ```
define hostgroup {
 hostgroup_name dns-servers
}
define host {
 name dns-server
 max_check_attempts 5
 hostgroups dns-servers
 register 0
}
define host {
 host_name soekris
 use dns-server
 address 196.200.223.1
}
define host {
 host_name upstream-dns-server
 use dns-server
 address 196.200.223.2
}
define command {
 command_name check_dns
 command_line $USER1$/check_dns -H www.yahoo.com -s $HOSTADDRESS$
}
define service {
 service_description dns
 use generic-service
 hostgroup dns-servers
 check_command check_dns
}
```



# Monitoring web sites with Nagios

---

- ```
define hostgroup {
    hostgroup_name websites
}
define host {
    name website
    max_check_attempts 5
    hostgroups websites
    register 0
}
define host {
    host_name www.yahoo.com
    use website
    address www.yahoo.com
}
define host {
    host_name www.google.com
    use website
    address www.google.com
}
define command {
    command_name check_site
    command_line $USER1$/check_http -H $HOSTADDRESS$
}
define service {
    service_description http
    use generic-service
    hostgroup websites
    check_command check_site
}
```



Testing Nagios

- Start Nagios now:
 - ***sudo /usr/local/etc/rc.d/nagios start***
- Click on the *Service Detail* link in the left-hand menu
- Check that the routers, DNS and Web servers come up in the PENDING state
- Click on the service name and reschedule the next check, make sure they become OK after a few seconds



Installing Smokeping

- Install from ports to enable some extra probes:
 - ***sudo pkg_add -r rrdtool p5-CGI-Session p5-PathTools p5-Digest-HMAC p5-Digest-MD5 p5-Net p5-Pod-Parser p5-SNMP_Session fping echoping p5-Net-DNS***
 - ***cd /usr/ports/net-mgmt/smokeping***
 - ***sudo make config***
 - Enable EchoPing and AnotherDNS probe options
 - ***sudo make deinstall clean install clean***
- Edit */etc/rc.conf* and add the following line:
 - ***smokeping_enable="YES"***



Configuring Smokeping

- Edit */usr/local/etc/smokeping/config* and change these settings:
 - sendmail = */usr/sbin/sendmail*
 - step = *60*
 - unison_tolerance = *10*
 - remove the “Slaves” section and “slaves =” lines
 - remove from “+ Test” to end of file



Configuring Apache for Smokeping

- Create */usr/local/etc/apache22/Includes/smokeping.conf* with these contents:
 - Alias /smokeping /usr/local/smokeping/htdocs
 <Location /smokeping>
 DirectoryIndex smokeping.cgi
 AddHandler cgi-script .cgi
 </Location>
 <Directory /usr/local/smokeping/htdocs>
 Allow from all
 </Directory>
- Tell Apache to reload its configuration:
 - ***/usr/local/etc/rc.d/apache22 reload***



Monitoring Routers with Smokeping

- Edit */usr/local/etc/smokeping/config* and append:
 - + kist
probe = FPing
host = 196.200.217.254
 - + serena
probe = FPing
host = 196.200.215.254
- Start Smokeping now:
 - `sudo /usr/local/etc/rc.d/smokeping start`



Monitoring DNS Servers

- In the ***** Probes ***** section, add:
 - + DNS
binary = /usr/bin/dig
lookup = www.yahoo.com
- In the ***** Targets ***** section, add:
 - + soekris
probe = DNS
host = 196.200.223.1
 - + upstream
probe = DNS
host = 196.200.223.2
- Restart Smokeping:
 - `sudo /usr/local/etc/rc.d/smokeping reload`

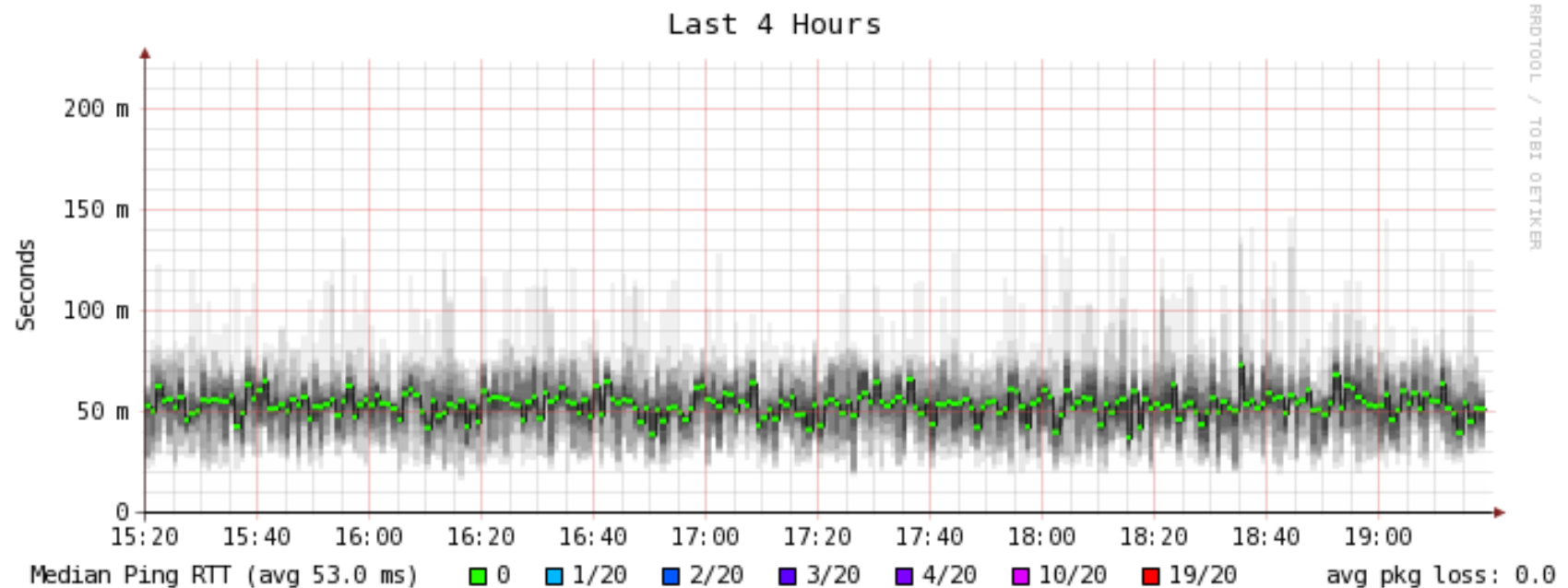


Monitoring Web Servers

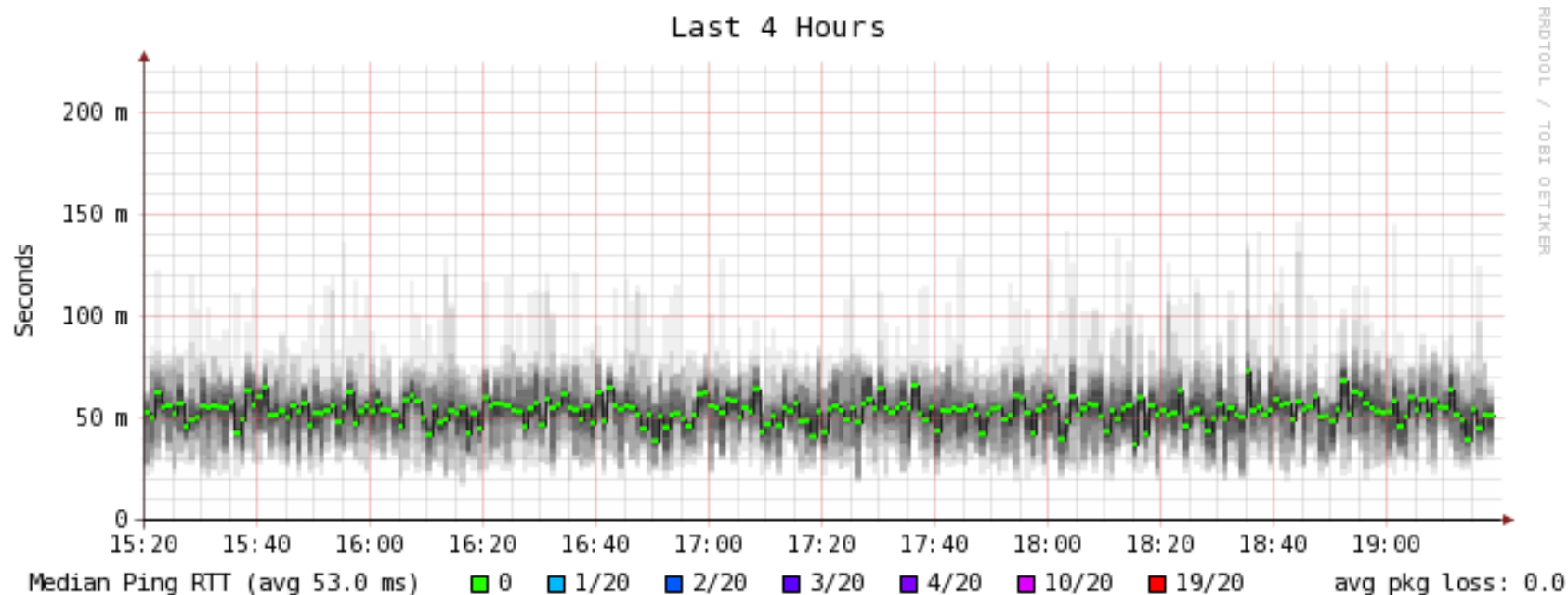
- In the `*** Probes ***` section, add:
 - + EchoPingHttp
binary = /usr/local/bin/echoping
timeout = 30
pings = 5
- In the `*** Targets ***` section, add:
 - + google
probe = EchoPingHttp
host = www.google.com
 - + yahoo
probe = EchoPingHttp
host = www.yahoo.com
- Restart Smokeping again



Reading Smokeping Graphs (1)



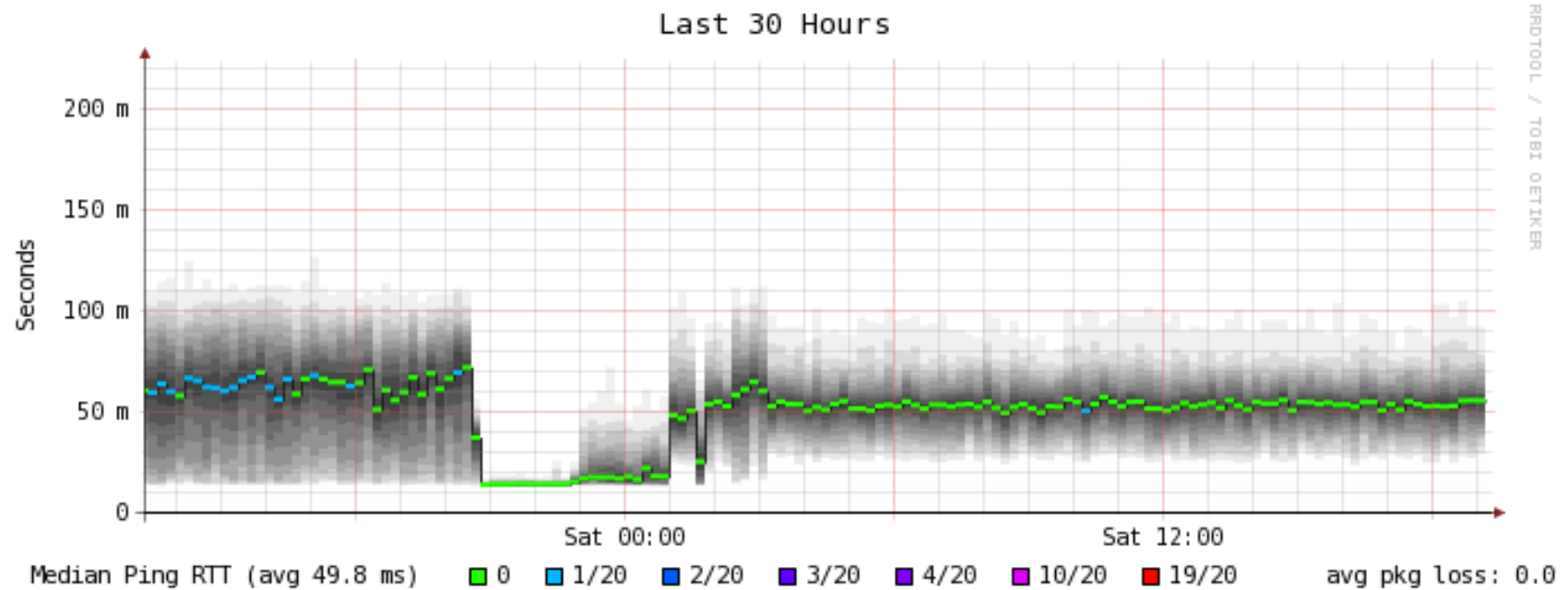
Reading Smokeping Graphs (1)



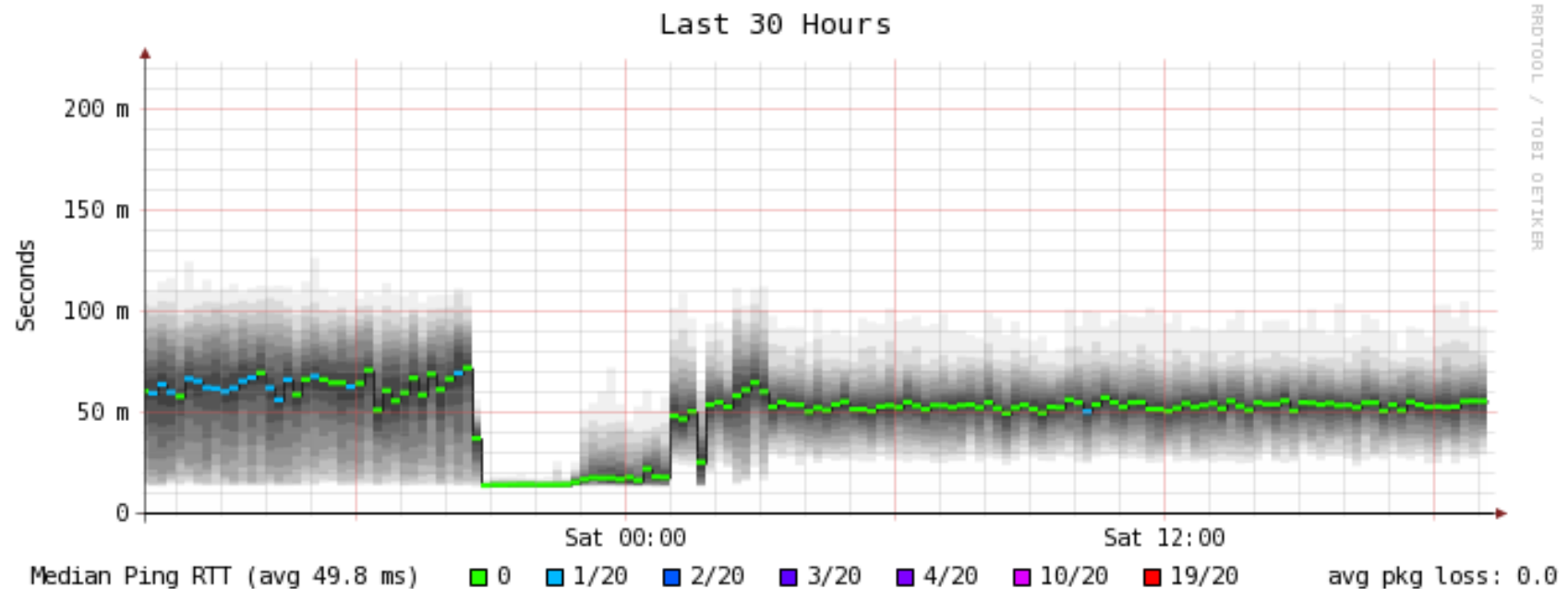
- Overall latency a little high for first hop
- Rather high jitter
- No packet loss



Reading Smokeping Graphs (2)



Reading Smokeping Graphs (2)



- Significant drop in latency and packet loss for a short period
- Conclusion: link is heavily loaded most of the time

Diagnosing Busy Connections

- Heavily loaded link could be due to:
 - inbound traffic
 - downloads, bittorrent, attacks, incoming spam
 - outbound traffic
 - uploads, bittorrent, virus or worm-infected PCs, outgoing spam
 - both at the same time
- Total volume of traffic is not helpful
- Need to identify the source of the traffic
 - Identifying the destination may not help



Finding the Culprit

- Switch LEDs may help you track down busy ports
 - Do not discriminate between local and remote traffic
- Managed switches can have traffic on each port monitored remotely by SNMP
- Flows are the next level down
 - Cisco or Juniper router with NetFlow/sFlow
 - Unix router or bridge running pmacct or ntop
- Packets are the lowest level
 - Unix router or transparent bridge running Wireshark
 - Expensive hardware network analysers



Going with the Flow

- Flows are useful tools for traffic monitoring
 - Identify who is talking to who, and often the protocol or type of traffic
 - Much less verbose and easier to understand than packets
- A flow is (usually) a unique:
 - pair of IP addresses
 - pair of ports
 - protocol
- Flows are sampled (number of bytes reported) at fixed time intervals to add a sixth dimension
- Generated by Cisco/Juniper router, or pmacct

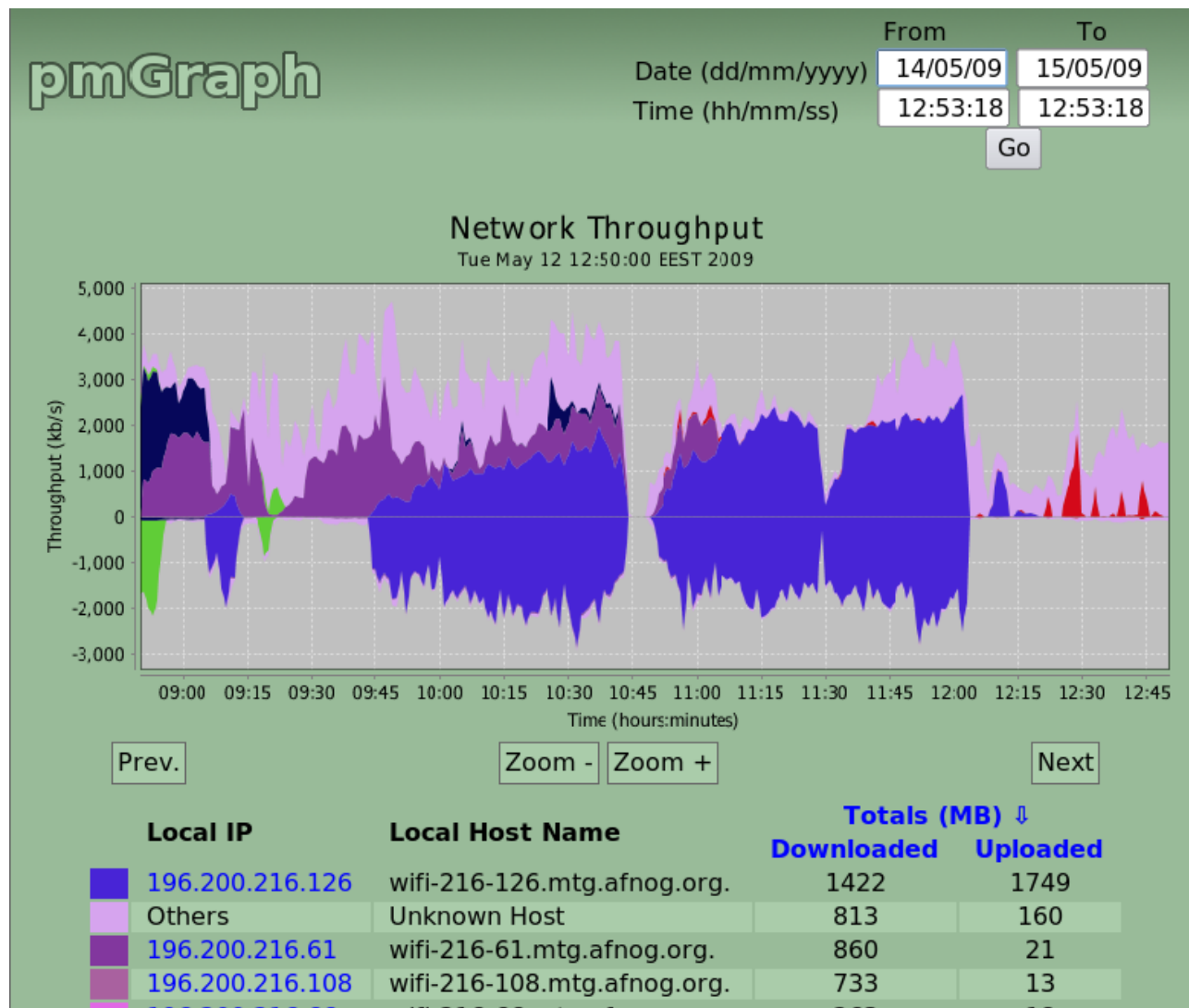


What do Flows look like?

| ip_src | ip_dst | sport | dport | proto | pkts | bytes | inserted |
|-----------------|-----------------|-------|-------|-------|------|--------|----------------|
| 41.190.128.29 | 196.200.216.44 | 143 | 63221 | tcp | 33 | 21776 | 25/05/10 13:32 |
| 196.200.216.94 | 213.254.211.14 | 50155 | 80 | tcp | 185 | 10160 | 25/05/10 13:32 |
| 213.254.211.14 | 196.200.216.94 | 80 | 50155 | tcp | 277 | 415500 | 25/05/10 13:32 |
| 213.199.149.57 | 196.200.216.156 | 80 | 50553 | tcp | 65 | 68255 | 25/05/10 13:33 |
| 196.200.216.100 | 213.254.211.8 | 58626 | 80 | tcp | 19 | 14192 | 25/05/10 13:32 |
| 196.200.216.133 | 69.64.75.206 | 49479 | 80 | tcp | 262 | 13374 | 25/05/10 13:32 |
| 69.64.75.206 | 196.200.216.133 | 80 | 49479 | tcp | 429 | 602968 | 25/05/10 13:32 |
| 209.85.229.155 | 196.200.216.133 | 80 | 49495 | tcp | 17 | 10428 | 25/05/10 13:32 |
| 209.85.229.155 | 196.200.216.133 | 80 | 49494 | tcp | 16 | 12119 | 25/05/10 13:32 |
| 69.64.72.239 | 196.200.216.133 | 80 | 49510 | tcp | 23 | 29652 | 25/05/10 13:32 |



What can we do with Flows?



Installing MySQL on FreeBSD

- Binary packages work fine:
 - ***sudo pkg_add -r mysql50-server***
- Edit */etc/rc.conf* and add the following line:
 - ***mysql_enable=YES***
- Start the MySQL server now:
 - ***sudo /usr/local/etc/rc.d/mysql-server start***



Installing pmacct on FreeBSD

- Install *pmacct* from ports to enable MySQL:
 - ***cd /usr/ports/net-mgmt/pmacct***
 - ***sudo make config***
 - Enable MySQL support
 - ***sudo make deinstall clean install***
- There are no *rc* scripts for *pmacct* in the port, so create */etc/rc.local* as follows:
 - ***#!/bin/sh***
 - ***/usr/local/sbin/nfacctd -D -f ***
 - ***/usr/local/etc/pmacct/nfacctd.conf***



Creating the *pmacct* Database

- *mysqladmin -u root create pmacct*
- *cat /usr/ports/net-mgmt/pmacct/work/pmacct-0.11.6/sql/pmacct-create-db_v6.mysql*
- ```
CREATE TABLE `acct_v6` (
 `ip_src` char(15) NOT NULL,
 `ip_dst` char(15) NOT NULL,
 `src_port` int(2) unsigned NOT NULL,
 `dst_port` int(2) unsigned NOT NULL,
 `ip_proto` char(6) NOT NULL,
 `packets` int(10) unsigned NOT NULL,
 `bytes` bigint(20) unsigned NOT NULL,
 `flows` int(10) unsigned NOT NULL,
 `stamp_inserted` datetime NOT NULL,
 `stamp_updated` datetime default NULL,
 KEY `isdb` (`stamp_inserted`, `ip_src`,
 `ip_dst`, `src_port`, `dst_port`, `bytes`);
```
- *mysql -u root pmacct < /usr/ports/net-mgmt/pmacct/work/pmacct-0.11.6/sql/pmacct-create-db\_v6.mysql*



# Grant Permissions on Database

---

- Create a password for the database
  - *mysql -u root pmacct*
  - mysql> *GRANT ALL ON pmacct.\* TO pmacct@localhost IDENTIFIED BY 'XXXXXXXX';*  
Query OK, 0 rows affected (0.00 sec)
  - mysql> *GRANT ALL ON pmacct.\* TO pmacct@127.0.0.1 IDENTIFIED BY 'XXXXXXXX';*  
Query OK, 0 rows affected (0.00 sec)



# Setting up a Netflow Collector

---

- Create */usr/local/etc/pmacct/nfacctd.conf* like this:
  - daemonize: false  
debug: true  
pidfile: /var/run/nfacctd.pid  
logfile: /var/log/nfacctd.log  
! syslog: daemon  
nfacctd\_port: 4096  
plugins: mysql  
aggregate: src\_host, src\_port, dst\_host, dst\_port, proto  
sql\_db: pmacct  
sql\_table: acct\_v6  
sql\_history: 1m  
sql\_history\_roundoff: m  
sql\_table\_version: 6  
sql\_host: 127.0.0.1  
sql\_user: pmacct  
sql\_passwd: **XXXXXXXXXX**  
sql\_refresh\_time: 60  
sql\_dont\_try\_update: true  
sql\_optimize\_clauses: true  
! sql\_preprocess: minb = 1000



# Enabling Netflow on Cisco

- You should enable Netflow on all active interfaces

- rtr-tedata> **show interface summary**

| Interface         | IHQ | IQD | OHQ | OQD | RXBS    | RXPS | TXBS    | TXPS | TRTL |
|-------------------|-----|-----|-----|-----|---------|------|---------|------|------|
| FastEthernet0/0   | 0   | 0   | 0   | 0   | 0       | 0    | 0       | 0    | 0    |
| * FastEthernet0/1 | 1   | 0   | 0   | 0   | 1684000 | 369  | 1944000 | 315  | 0    |
| * Serial0/0/0     | 0   | 0   | 0   | 0   | 957000  | 148  | 703000  | 165  | 0    |
| * Serial0/0/1     | 0   | 0   | 0   | 0   | 1324000 | 182  | 1223000 | 201  | 0    |
| * Serial0/2/0     | 0   | 0   | 0   | 0   | 469000  | 101  | 887000  | 140  | 0    |

- rtr-tedata# **conf t**

rtr-tedata(config)# **interface FastEthernet0/1**

rtr-tedata(config-if)# **ip route-cache flow**

rtr-tedata(config-if)# **exit**

rtr-tedata(config)# **interface Serial0/0/0**

rtr-tedata(config-if)# **ip route-cache flow**

rtr-tedata(config-if)# **exit**

- rtr-tedata# **show ip flow top-talkers**

| SrcIf   | SrcIPaddress    | DstIf  | DstIPaddress    | Pr | SrcP | DstP | Bytes |
|---------|-----------------|--------|-----------------|----|------|------|-------|
| Se0/0/0 | 213.136.96.104  | Fa0/1* | 196.200.216.77  | 11 | 04AA | 04A4 | 1539K |
| Se0/0/0 | 24.17.17.180    | Fa0/1* | 196.200.216.125 | 06 | A6CE | 1C2A | 1522K |
| Se0/0/0 | 188.24.50.177   | Fa0/1* | 196.200.216.125 | 06 | E87A | 1C2A | 1433K |
| Se0/2/0 | 207.148.178.122 | Fa0/1* | 196.200.216.125 | 06 | BE90 | 1C2A | 834K  |
| Se0/0/1 | 195.226.227.100 | Fa0/1* | 196.200.216.125 | 06 | EEA3 | 1C2A | 647K  |



# Exporting Netflow Data from Cisco

---

- If your collector's IP address is 1.2.3.4:
  - ssh cisco
  - enable
  - conf t
  - ip flow-cache timeout active 1
  - ip flow-cache timeout inactive 60
  - ip flow-export version 5
  - ip flow-export destination 1.2.3.4 4096
  - exit
  - write



# Alternative: Monitoring Box

---

- Need a Unix box that can sniff the traffic:
  - Attached to a monitoring port of a managed switch
  - Attached to a dumb hub
  - Routing traffic between subnets
  - Bridging two LANs
- Options:
  - Use an existing Unix router or proxy
  - Create a new transparent bridge
  - Add a router outside of LAN (e.g. WAN side)
  - Reconfigure entire LAN



# Transparent Bridging with FreeBSD

---

- Need a PC with at least two LAN interfaces
- Add the following lines to /etc/rc.conf:
  - *cloned\_interfaces="bridge0"*
  - *ifconfig\_bridge0="addm em0 addm re0 up DHCP"*
  - *ifconfig\_em0="up"*
  - *ifconfig\_re0="up"*
- Restart networking:
  - *sudo /etc/rc.d/netif restart*
- Insert bridge in front of client PC(s)
- Test that clients can still access the Internet!



# Setting up the Flow Logger

---

- ***cat /usr/local/etc/pmacct/pmacctd.conf***  
! daemonize: true  
debug: true  
pidfile: /var/run/pmacctd.pid  
! syslog: daemon  
plugins: mysql  
aggregate: src\_host, src\_port, dst\_host, dst\_port, proto  
interface: bridge0  
sql\_db: pmacct  
sql\_table: acct\_v6  
sql\_history: 1m  
sql\_history\_roundoff: m  
sql\_table\_version: 6  
sql\_host: 127.0.0.1  
sql\_user: pmacct  
sql\_passwd: **XXXXXXXXXX**  
sql\_refresh\_time: 60  
sql\_dont\_try\_update: true  
sql\_optimize\_clauses: true  
! sql\_preprocess: minb = 10000



# Starting the Flow Logger

---

- ***sudo /usr/local/sbin/pmacctd -f /usr/local/etc/pmacct/pmacctd.conf***
- ***mysql pmacct -u root***
- ***mysql> select ip\_src, ip\_dst, src\_port, dst\_port, bytes, stamp\_inserted from acct\_v6 limit 5;***

- | ip_src         | ip_dst         | src_port | dst_port | bytes | stamp_inserted      |
|----------------|----------------|----------|----------|-------|---------------------|
| 196.200.223.2  | 196.200.208.4  | 60346    | 22       | 37792 | 2009-05-09 17:22:00 |
| 196.200.223.2  | 196.200.208.4  | 52755    | 22       | 37872 | 2009-05-09 17:27:00 |
| 196.200.216.38 | 196.200.208.20 | 50689    | 22       | 12976 | 2009-05-09 17:48:00 |
| 196.200.216.51 | 196.200.208.20 | 52059    | 23       | 12108 | 2009-05-09 17:48:00 |
| 196.200.216.32 | 69.147.102.99  | 64980    | 80       | 10292 | 2009-05-09 17:50:00 |

5 rows in set (0.00 sec)

- Change *pmacctd.conf* to enable running as a daemon
- Add the command above to */etc/rc.local*



# Installing Tomcat

---

- Install from ports:
  - *cd /usr/ports/www/tomcat6*
  - *sudo make install clean*
- You may need to follow the instructions to download 180 MB of Java from Sun's website
- Edit */etc/rc.conf* and add the following lines:
  - *tomcat60\_enable=YES*
  - *tomcat60\_java\_opts="-Djava.awt.headless=true"*
- Start Tomcat now:
  - */usr/local/etc/rc.d/tomcat6 start*



# Installing pmGraph

---

- Download *pmgraph-1.3.war* from *http://pmgraph.sourceforge.net*
  - Should be a copy in /home/afnog on your machine
- ***cd /usr/local/apache-tomcat6.0/webapps***
- ***sudo mkdir pmgraph***
- ***cd pmgraph***
- ***sudo jar xf /home/afnog/pmgraph-1.3.war***



# Configuring pmGraph

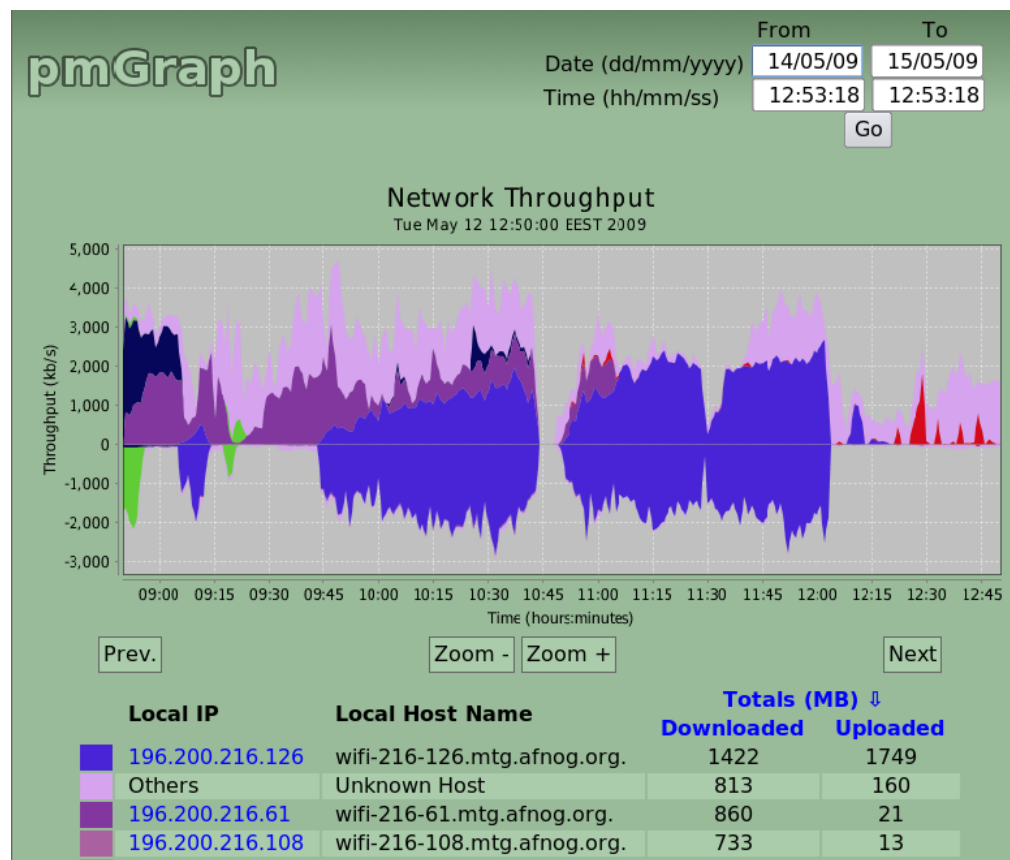
---

- *cd /usr/local/apache-tomcat6.0/webapps/pmgraph/WEB-INF/classes*
- *sudo vi database.properties*
  - DatabaseURL = jdbc:mysql://*localhost/pmacct*
  - DatabasePass = *XXXXXXXXXX*
  - LocalSubnet = *196.200.219.*
- Restart Tomcat:
  - *sudo killall java*
  - *sudo /usr/local/etc/rc.d/tomcat6 start*
- Should work but doesn't:
  - *sudo /usr/local/etc/rc.d/tomcat6 restart*



# Testing pmGraph

- *fetch ftp://noc.ws.afnog.org/pub/g4l-v0.33.iso*
- Browse to *http://localhost:8180/pmgraph:*



# Next

---

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- ✓ Troubleshooting an Internet connection
- ✓ Monitoring an Internet connection
- **Setting policy**
  - Enforcing Policy
    - Social measures
    - Technical measures
  - Summary and resources

# What Next?

---

- Internet connection is sometimes full
- What can be done about it?
  - Block traffic that nobody wants (viruses, spam)
  - Efficiency savings (perhaps 10-50%)
  - Changing user behaviour
- Changing behaviour requires education and policy

# Blocking Unwanted Traffic

---

- Outbound worm traffic is the most likely candidate
  - Identify infected machines (using monitoring tools)
  - Clean them and install antivirus software
  - Keep antivirus up to date
  - Block ports used by worms
  - Set alarms to detect infected machines in future
- Incoming spam may waste some capacity
  - Monitoring will tell you how much traffic is email
  - Good local spam filtering can help, but is difficult!
  - Remote email filtering services can help (e.g. Barracuda, LBSD)



# Efficiency Savings

---

- Run a local DNS cache
- Run a local web cache
- Identify commonly downloaded files as candidates for local mirroring
- Check for inter-site traffic due to Active Directory and VPNs
- Don't expect too much improvement here



# What is a Policy

---

- Rules on what a network (or Internet connection) can or can't be used for
  - also known as an Acceptable Use Policy
- Every good network has some kind of Acceptable Use Policy
- Users of a shared connection are entitled to agree on rules for sharing it
  - Rules imposed from above are usually unpopular
- How can we set policy fairly?



# Why Set a Policy

---

- Network abuse is a social problem
- Social problems require social solutions
  - Changing network traffic means changing user behaviour
  - Technical solutions doesn't change attitudes
  - Rules, conventions, debate, consultation and consensus can be more effective
  - Requires buy-in from the top levels of organisations
- Policy guides implementation
  - Easier to decide what to block or restrict
  - Implementation without policy can be accused of being unaccountable, unfair, arbitrary or just wrong



# What's in a Policy

---

- The best Acceptable Use Policies would be:
  - Based on evidence
  - Set by consensus
  - Known by all
  - Monitored
  - Enforced
  - Reviewed regularly

# Collecting Evidence

---

- Show effects of high network traffic on essential applications (e.g. by correlation or measurement)
- Show how much network traffic is used for different purposes (without prejudging)
- Show how much network traffic is used by the top users and departments (without naming them)
- Show the causes of high network traffic (applications, working practices, visibility)
- Show how much could be saved by efficiency measures (e.g. caches)



# Proposing a Policy

---

- Consider whether certain applications have a good case for work use
  - Who says P2P, banner adverts or Skype are not business functions?
- Consider charging for usage (by volume or rate)
- Consider quotas on bandwidth use
- Consider throttling user traffic based on usage
- Consider applying the same rules as for phone calls, printing, photocopying



# Reaching Consensus

---

- Involve all stakeholders (worth the effort)
- Present the evidence, and create space for discussion
- Explore all possible social and technical solutions
- Ensure that all views are taken into account
  - Try to accommodate dissent, e.g. allow personal use out of hours or within defined limits
- Try to avoid “design by committee” bloat
  - Make a case for simplicity
  - Don’t be afraid to leave open to interpretation, e.g. “academic use” or “business use”



# Consensus Failure

---

- If consensus cannot be reached:
  - Find out why it's being blocked
  - Check that all views were taken into account
  - Make another proposal
  - Consider delaying implementation
  - Try a different decision mechanism
  - Consider imposing a temporary policy (with a time limit)



# Publishing Policy

---

- Important that all users knows the policy
  - Users won't follow unwritten rules
- Post in the usual places (computer rooms, letters to new members and users)
- If possible, collect signatures before allowing access (issuing user identifiers)
- Publish the complete policy
  - even if some of it only applies to some users
  - more reason to keep it short and simple!



# Reviewing Policy

---

- Decide and publish the review date in the policy
- Users are more likely to accept a temporary restriction than a permanent one
- Users are more likely to agree if they feel that:
  - They are being listened to
  - Their views have an influence on the policy
- Solicit comments in the policy document itself
- Log comments for review time
- Help people to comment anonymously



# Next

---

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- ✓ Troubleshooting an Internet connection
- ✓ Monitoring an Internet connection
- ✓ Setting policy
- **Enforcing Policy**
  - Social measures
  - Technical measures
- Summary and resources



# Monitoring Compliance

---

- Easy to set policy and never monitor compliance
- Sometimes only checked when a breach is suspected
- Data may no longer be available
- Users will lose respect for policy over time
- Better to at least collect compliance data continuously
- Good idea to delete data after some time
- Good idea to inform users (privacy policy)



# Accountability

---

- Monitoring often gives a list of IP addresses
- How to connect them to users?
  - NAT problem
  - IP address spoofing
  - MAC address spoofing
  - Switch port security
  - Shared computers (e.g. labs)
  - Wireless clients
- 802.1x authentication solves many problems
- Proxy authentication can be a partial solution



# Next

---

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- ✓ Troubleshooting an Internet connection
- ✓ Monitoring an Internet connection
- ✓ Setting policy
- Enforcing Policy
  - **Social measures**
  - Technical measures
- Summary and resources



# Social Measures

---

- Network abuse is a social problem, not technical
- In most cases, social solutions work better:
  - Users may not be aware of their bandwidth use
  - Consider educating users on bandwidth use and tools
  - Likely to be few network abusers (about 5%)
  - Likely to be the most technically skilled
  - Discuss the problem with them first, in private
  - Consider publishing a list of the heaviest users
  - Consider disciplinary action, revoking privileges
- If necessary, technical options are available



# Next

---

- ✓ What is bandwidth management
- ✓ When to manage bandwidth
- ✓ Troubleshooting an Internet connection
- ✓ Monitoring an Internet connection
- ✓ Setting policy
- Enforcing Policy
  - ✓ Social measures
  - **Technical measures**
- Summary and resources

# Technical Measures

---

- Traffic prioritisation (tc, dummynet, altq)
- Limiting bandwidth used by some kinds of traffic
- Interactive fair sharing between IPs (SFQ, WFQ)
- Hard bandwidth quotas (cut off users over limit)
- Soft bandwidth quotas (throttle users over limit)
- Flexible throttling (progressively reduce bandwidth)

# Traffic Prioritisation (1)

---

- client: ping 4.2.2.2
- sudo kldload ipfw dummmynet
- sudo ipfw add pipe 1 ip from any to 196.200.218.0/24
- sudo ipfw add pipe 2 ip from 196.200.218.0/24 to any
- sudo vi /etc/sysctl.conf
  - net.link.bridge.ipfw=1
- sudo /etc/rc.d/sysctl restart
- client: fetch <http://196.200.218.200/bigfile>

# Traffic Prioritisation (2)

---

- `sudo ipfw queue 1 config pipe 1 weight 100`
- `sudo ipfw queue 2 config pipe 1 weight 50`
- `sudo ipfw queue 3 config pipe 2 weight 100`
- `sudo ipfw queue 4 config pipe 2 weight 50`
- `sudo ipfw flush`
- `sudo ipfw add queue 1 icmp from any to 196.200.218.0/24`
- `sudo ipfw add queue 2 ip from any to 196.200.218.0/24`
- `sudo ipfw add queue 3 icmp from 196.200.218.0/24 to any`
- `sudo ipfw add queue 4 ip from 196.200.218.0/24 to any`

# Hard quotas

---

- pmacct database comes in very useful!
- ```
echo 'SELECT ip_dst, sum(bytes) AS bytes
FROM acct_v6
WHERE ip_dst LIKE "196.200.218.%"
AND ip_src NOT LIKE "196.200.218.%"
GROUP BY ip_dst
HAVING bytes > 1000000' |
mysql pmacct -u root |
while read ip bytes; do
    ipfw add deny ip from $ip to any
    ipfw add deny ip from any to $ip
done
```

