5
Troubleshooting

How you establish the support infrastructure for your network is as important as what type of equipment you use. Once your network is connected to the Internet, or opened up to the general public, considerable threats may come from the Internet, or from your users themselves. These threats can range from the benign to the outright malevolent, but all will have an impact on your network if it is not properly configured.

This chapter will show you some common problems that are frequently encountered on modern networks, as well as simple methods for quickly determining the source of the trouble. But before you jump straight into the technical details, it is a good idea to get into the mindset of a good troubleshooter. You will solve your problems much faster if you cultivate an attitude of lateral thinking and problem solving. If networks could be fixed by simply trying random tactics without understanding the problem, then we would program a computer to do it for us.

Proper troubleshooting technique

No troubleshooting methodology can completely cover all bandwidth problems you will encounter when working with networks. Having a clear methodology for both preparing and responding to the problems that you encounter will keep you working the right direction.

But often, problems come down to one of a few common mistakes. Here are a few simple points to keep in mind that can get your troubleshooting effort working in the right direction.

Preparing for problems

- **Make regular backups.** Over time your network configuration will grow and expand to suit your particular network. Remembering the intricate details will become impossible making it very difficult to reproduce the same configuration if it is lost. Making regular backups ensures that you can rebuild your configuration from scratch if required. Having multiple backups means you can roll back to a previous known working state if a configuration change goes awry.
- **Disaster plan.** Technology is not always as reliable as we hope, and it is a certainty that at some point major problems will strike your network. By planning for these and having a procedure in place for dealing with them you will be in a far better situation when the lights go off!
- **Fallback network mode.** It is useful to prepare a basic network configuration state, which only allows a minimum set of services on a network. When a problem occurs which stops the network from functioning effectively you can implement this fallback mode, allowing others to use essential services whilst you are troubleshooting the problem.

Responding to a problem

- **Don't panic.** Don't panic, inform your users of the problem, and set about solving it in a methodical and guided manner.
- **Understand the problem.** If you are troubleshooting a system, that means that it was working at one time, and probably very recently. Before jumping in and making changes, survey the scene and assess exactly what is broken. If you have historical logs or statistics to work from, all the better. Be sure to collect information first, so you can make an informed decision before making changes.
- **Is it plugged in?** This step is often overlooked until many other avenues are explored. Plugs can be accidentally (or intentionally) unplugged very easily. Is the lead connected to a good power source? Is the other end connected to your device? Is the power light on? It may sound silly, but you will feel even sillier if you spend a lot of time checking out an antenna feed line only to realise that the AP was unplugged the entire time. Trust me, it happens more often than most of us would care to admit.
- **What was the last thing changed?** If you are the only person with access to the system, what is the last change you made? If others have access to it, what is the last change they made and when? When was the last time the system worked? Often, system changes have unintended consequences that may not be immediately noticed. Roll back that change and see what effect it has on the problem.

- **Make a backup.** This applies before you notice problems, as well as after. If you make a complicated software change to a system, having a backup means that you can quickly restore it to the previous settings and start again. When troubleshooting very complex problems, having a configuration that "sort-of" works can be much better than having a mess that doesn't work at all (and that you can't easily restore from memory).
- **The known good.** This idea applies to hardware, as well as software. A ***known good*** is any component that you can replace in a complex system to verify that its counterpart is in good, working condition. For example, you may carry a tested Ethernet cable in a tool kit. If you suspect problems with a cable in the field, you can easily swap out the suspect cable with the known good and see if things improve. This is much faster and less error-prone than re-crimping a cable, and immediately tells you if the change fixes the problem. Likewise, you may also pack a backup battery, antenna cable, or a CD-ROM with a known good configuration for the system. When fixing complicated problems, saving your work at a given point lets you return to it as a known good, even if the problem is not yet completely solved.
- **Change one variable at a time.** When under pressure to get a failed system back online, it is tempting to jump ahead and change many likely variables at once. If you do, and your changes seem to fix the problem, then you will not understand exactly what led to the problem in the first place. Worse, your changes may fix the original problem, but lead to more unintended consequences that break other parts of the system. By changing your variables one at a time, you can precisely understand what went wrong in the first place, and be able to see the direct effects of the changes you make.
- **Do no harm.** If you don't fully understand how a system works, don't be afraid to call in an expert. If you are not sure if a particular change will damage another part of the system, then either find someone with more experience or devise a way to test your change without doing damage. Putting a penny in place of a fuse may solve the immediate problem, but it may also burn down the building.

A basic approach to a broken network

It happens all the time. Suddenly, the network isn't working at all. What do you do? Here are some basic checks that will quickly point to the cause of the problem.

First make sure the problem is not just with the one web server you want to contact. Can you open other websites such as *www.google.com*? If you can open popular web sites but not the one you requested, the problem is likely with the site itself, or with the network between you and the other end. If your web browser cannot load pages from the Internet, next try to browse to a server on the local network (if any). If local sites are shown quickly, then that may in-

dicate a problem with the Internet connection or the proxy server. If not, then you most likely have a problem with the local network.

If you suspect a problem with the proxy server, try accessing information with a different program, such as an email client. If you can send and receive email, but web browsing still doesn't work, then this may further indicate problems with the proxy server.

Your web browser and mail client can only provide so much information. If nothing seems to be working at all, it's time to switch to a more useful diagnostic tool such as **`ping`**.

Try pinging a well know server such as *google.com*.

```
$ ping www.google.com
PING www.l.google.com (66.102.9.99) 56(84) bytes of data.
64 bytes from 66.102.9.99: icmp_seq=1 ttl=243 time=30.8 ms
64 bytes from 66.102.9.99: icmp_seq=2 ttl=243 time=31.6 ms
64 bytes from 66.102.9.99: icmp_seq=3 ttl=243 time=30.9 ms

--- www.l.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 30.865/31.176/31.693/0.395 ms
```

This indicates that your network connection is working just fine, and the problem is very likely with your proxy (or your web browser's proxy settings). Sometimes, **`ping`** will hang while waiting for a DNS reply:

```
$ ping www.google.com
```

In this case, DNS resolution isn't working. Without DNS, just about everything else on the network will seem to be broken. Check your local machine settings to be sure that it is using the correct DNS server. If you can't ping a server using the domain name, the next step would be to ping a known IP address - 4.2.2.2 is an easy to remember address.

```
$ ping 4.2.2.2
PING 4.2.2.2 (4.2.2.2) 56(84) bytes of data.
64 bytes from 4.2.2.2: icmp_seq=1 ttl=247 time=85.6 ms
64 bytes from 4.2.2.2: icmp_seq=2 ttl=247 time=86.3 ms
64 bytes from 4.2.2.2: icmp_seq=3 ttl=247 time=84.9 ms
64 bytes from 4.2.2.2: icmp_seq=4 ttl=247 time=84.8 ms

--- 4.2.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 84.876/85.436/86.330/0.665 ms
```

If you can ping an IP address but not a domain name, then the network is fine but your computer is unable to convert the domain name (*www.google.com*)

into an IP address (66.102.9.99). Check to make sure that your DNS server is running and is reachable.

If you can't ping an Internet IP address, then it's a good idea to make sure that your local network connection is still working. Does your computer have a valid IP address? Use **ifconfig** on UNIX or **ipconfig** on Windows to make sure your IP settings are correct.

If you don't have an IP address then you are definitely not connected to the Internet. Check the cables from your computer (or the wireless settings if using wireless). Also check that your DHCP server is up and running, if you use DHCP. If you have an IP address but it is incorrect, then there are only two possibilities. Either your machine is using the wrong settings, or there is a rogue DHCP server on the local network (page **170**). Either change your local settings or track down the bad DHCP server.

If you do have a valid IP address, try pinging the gateway's IP address.

```
$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.489 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.496 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.406 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.449 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.406/0.460/0.496/0.035 ms
```

If you can't ping your gateway then the problem is definitely in your local network - maybe the switch or router needs to be restarted. Check all the cables (both network cables and power cables).

If you can ping your gateway, then you should next check the Internet connection. Maybe there is a problem upstream. One good test is to log into your gateway router and try to ping the gateway at your ISP. If you cannot ping your ISP's gateway, then the problem is with your Internet connection. If you can ping your ISP's gateway but no other Internet hosts, then the problem may exist beyond your ISP. It's probably a good idea to phone your ISP and check if they have a problem.

Is everything still not working? Then it's time to roll up your sleeves and get to work. You should reread the **Troubleshooting Technique** section (page **159**) and settle down for some slow and methodical work, checking each part of your network bit by bit.

Common symptoms

It can be difficult at times to diagnose a problem by simply looking at utilisation graphs or running spot check tools like **`ping`**. This is why establishing a baseline is so important. If you know what your network should "look like," then you can usually find the problem by looking for anything out of the ordinary. Here are examples of common, but often subtle problems that may cause anything from performance degradation to complete network outages.

Automatic updates

Automatic updates for virus scanners, spyware scanners, and Microsoft Windows are very important to a healthy, virus-free network environment. However, most auto-update systems can cause very heavy network usage at inopportune times, especially when a new, critical patch is made available. Many auto-update systems make use of HTTP to download updates, so keep an eye out for specific auto-update URLs. It may be worth compiling a list of your own auto-update URLs by monitoring your logs, and checking widely-installed software. Some common URLs are *http://update.microsoft.com/* and *http://update.adobe.com/*.

You may want to configure your auto-update software to download updates outside of business hours, as some software allows you to set at which time to check and download updates. You may also want to research if your vendor offers products to cache the automatic updates on a server on your network. This way, you only need to transfer the update once over the Internet, and the patch or signature can be quickly sent to all hosts using the much faster local network.

Sometimes vendors do not offer cost effective solutions, or such solutions may not even exist. In this case, you may want to consider mirroring these sites using split horizon DNS (page **212**). Sometimes, you may be able to cache these updates on your web proxy.

Simply blocking the update site on the proxy server is not a good solution because some update services (such as Windows automatic updates) will simply retry more aggressively. If all workstations do that at once, it places a heavy load on the proxy server. The extract below is from the proxy log (Squid access log) where this was done by blocking Microsoft's cabinet (**`.cab`**) files. Much of the Squid log was full of lines like this:

```
2003.4.2 13:24:18 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
*DENIED* Banned extension .cab GET 0
```

While this may be tolerable for a few PC clients, the problem grows significantly as hosts are added to the network. Rather than forcing the proxy server to serve requests that will always fail, it makes more sense to redirect the Software Update clients to a local update server.

Spyware

Often when users browse sites, these sites will install little programs on the users' PCs. Some of these programs will collect information from the PC and send it to a site somewhere on the Internet. By itself this might not much of a threat to your Internet connection, but if you have hundreds of machines doing this it can be a problem as they will quickly fill your bandwidth and perhaps flood your proxy server. The symptom of this particular problem would be a slow client machine and lots of similar requests for machines in your proxy log files. It is very important to run anti-virus and anti-spyware software on all machines, and to ensure that it is updated regularly, to counter this threat.

P2P

Peer-To-Peer applications are used for the sharing of media, software, and other content between groups of users. They are often left running, and can use up bandwidth downloading and uploading data even when the user believes it to be closed.

Detecting P2P software can be difficult, other than monitoring the volume of traffic. P2P applications often use protocols designed to bypass firewalls and proxies, disguising the data as normal web traffic, for example.

It is possible to detect P2P traffic by monitoring the number of connections that a client machine is making to a remote server. P2P software often tries to make multiple connections to speed up transfers. Alternative techniques involve the detailed inspection of packets passing through a server and marking them as P2P traffic where detected. This is possible by using an application layer firewall such as L7-filter (*http://l7-filter.sourceforge.net/*). However, this kind of filtering is processor intensive and not 100% foolproof.

It is important that P2P traffic is not blocked, only rate limited heavily. Most software will attempt to jump ports and disguise itself more effectively if a block is detected. By rate limiting the traffic, the traffic can be kept under control.

Email

Monitoring the baseline percentage of email utilisation on your connection will give you a good understanding of typical email usage. If the volume of email traffic rises significantly then further inspection will be required to ascertain why.

Users restricted in other areas may attempt to use email to bypass the restrictions, perhaps sending large files. Limiting the size of email attachments can help alleviate the load.

Viruses will often use email to deliver their payload, and rising email volume can be a good indicator of a virus problem on the network.

Open email relay hosts

An SMTP server which allows a computer from any address to send email to any other address is called an ***open relay***. This kind of server can be compromised and made use of by spammers to send large quantities of email, thus consuming large amounts of bandwidth. They do this to hide the true source of the spam, and avoid getting caught. This kind of behaviour will show up in analyses of network communications, with increases in uses of email. However, the best policy is to ensure that your SMTP servers will only accept email to or from domains you control, avoiding the problem from ever occurring.

To test for an open relay host, the following test should be carried out on your mail server (or on the SMTP server that acts as a relay host on the perimeter of the campus network). Use **`telnet`** to open a connection to port 25 of the server in question (with some Windows versions of telnet, it may be necessary to type **`set local_echo`** before the text is visible):

```
telnet mail.uzz.ac.zz 25
```

Then, if an interactive command-line conversation can take place (for example, as follows), the server is an open relay host:

```
MAIL FROM: spammer@waste.com
250 OK - mail from <spammer@waste.com>
RCPT TO: innocent@university.ac.zz
250 OK - rcpt to spammer@waste.com
```

Instead, the reply after the first MAIL FROM should be something like:

```
550 Relaying is prohibited.
```

An online tester is available at sites such as *http://www.ordb.org/*. There is also information about the problem at this site. Since bulk emailers have automated methods to find such open relay hosts, an institution that does not protect its mail systems is almost guaranteed to be found and abused. Configuring the mail server not to be an open relay consists of specifying the networks and hosts that are allowed to relay mail through them in the MTA (e.g., Sendmail, Postfix, Exim, or Exchange). This will likely be the IP address range of the campus network.

Email forwarding loops

Occasionally, a single user making a mistake can cause a problem. For example, a user whose university account is configured to forward all mail to her Yahoo account. The user goes on holiday. All emails sent to her in her absence are still forwarded to her Yahoo account, which can grow to only 2 MB. When the Yahoo account becomes full, it starts bouncing the emails back to the university account, which immediately forwards it back to the Yahoo account. An email loop is formed that might send hundreds of thousands of emails back and forth, generating massive traffic and crashing mail servers.

There are features of mail server programs that can recognise loops. These should be turned on by default. Administrators must also take care that they do not turn this feature off by mistake, or install an SMTP forwarder that modifies mail headers in such a way that the mail server does not recognise the mail loop.

Open proxies

A proxy server should be configured to accept only connections from the university network, not from the rest of the Internet. This is because people elsewhere will connect and use open proxies for a variety of reasons, such as to avoid paying for international bandwidth. The way to configure this depends on the proxy server you are using. For example, you can specify the IP address range of the campus network in your **`squid.conf`** file as the only network that can use Squid (page **272**). Alternatively, if your proxy server lies behind a border firewall, you can configure the firewall to only allow internal hosts to connect to the proxy port.

Programs that install themselves

There are programs that automatically install themselves from the Internet and then keep on using bandwidth - for example, the so-called Bonzi-Buddy, the Microsoft Network, and some kinds of worms. Some programs are spyware, which keep sending information about a user's browsing habits to a company somewhere on the Internet. These programs are preventable to some extent by user education and locking down PCs to prevent administrative access for normal users. In other cases, there are software solutions to find and remove these problem programs, such as Spychecker (*http://www.spychecker.com/*), Ad-Aware (*http://www.lavasoft.de/*), or xp-antispy (*http://www.xp-antispy.de/*).

Programs that assume a high bandwidth link

In addition to Windows updates, many other programs and services assume that bandwidth is not a problem, and therefore consume bandwidth for reasons

the user might not predict. For example, anti-virus packages (such as Norton Antivirus), periodically update themselves automatically and directly from the Internet. It is better if these updates are distributed from a local server.

Other programs, such as the RealNetworks video player, automatically download updates and advertisements, as well as upload usage patterns back to a site on the Internet. Innocuous looking applets (like Konfabulator and Dashboard widgets) continually poll Internet hosts for updated information. These can be low bandwidth requests (like weather or news updates), or very high bandwidth requests (such as webcams). These applications may need to be throttled or blocked altogether.

The latest versions of Windows and Mac OS X also have a time synchronisation service. This keeps the computer clock accurate by connecting to time servers on the Internet. It is better to install a local time server and distribute accurate time from there, rather than to tie up the Internet link with these requests.

Windows traffic on the Internet link

Windows computers communicate with each other via ***NetBIOS*** and ***Server Message Block*** (***SMB***). These protocols work on top of TCP/IP or other transport protocols. It is a protocol that works by holding elections to determine which computer will be the ***master browser***. The master browser is a computer that keeps a list of all the computers, shares, and printers that you can see in ***Network Neighborhood*** or ***My Network Places***. Information about available shares are also broadcast at regular intervals.

The SMB protocol is designed for LANs and causes problems when the Windows computer is connected to the Internet. Unless SMB traffic is filtered, it will also tend to spread to the Internet link, wasting the organisation's bandwidth. The following steps might be taken to prevent this:

- **Block outgoing SMB/NetBIOS traffic on the perimeter router or firewall.** This traffic will eat up Internet bandwidth, and worse, poses a potential security risk. Many Internet worms and penetration tools actively scan for open SMB shares, and will exploit these connections to gain greater access to your network. To block this traffic, you should filter TCP and UDP ports 135-139, and TCP port 445.
- **Install ZoneAlarm on all workstations (not the server).** A free version can be found at *http://www.zonelabs.com/*. This program allows the user to determine which applications can make connections to the Internet and which ones cannot. For example, Internet Explorer needs to connect to the Internet, but Windows Explorer does not. ZoneAlarm can block Windows Explorer from doing so.

- **Reduce network shares.** Ideally, only the file server should have any shares. You can use a tool such as SoftPerfect Network Scanner (from *http://www.softperfect.com/*) to easily identify all the shares in your network.

Streaming media / Voice over IP

Streaming audio and video comes in many forms, Internet radio and video are popular on the web, whilst people can communicate through video and audio using Voice over IP and instant messaging tools. All these types of media streaming use large amounts of bandwidth, and can reduce availability for other services. Many of these services use well known ports, which can be detected and limited or blocked by firewalls.

Service	TCP	UDP
Realtime Streaming Protocol (RTSP) - for Quicktime 4, Real Video etc.	554	5005
Realtime Transport Protocol (RTP) - used by iChat for audio & video		16384-16403
Real Audio & Video	7070	6970-7170
Windows Media	1755	1755
Shoutcast Audio	8000	
Yahoo Messenger (voice)	5000-5001	5000-5010
AIM (AOL Instant Messenger) (video)	1024-5000	1024-5000
Yahoo Messenger (video)	5100	
Windows Messenger (voice)		2001-2120, 6801, 6901
MSN file transfers	6891-6900	
MSN Messenger (voice)	6901	6901

Your main line of defense is user education, as most users rarely consider bandwidth as a limited resource. If streaming continues to be a problem, you may want to consider traffic shaping or blocking of these ports.

Sometimes streaming is required and may be part of your organisation's vision in multimedia services. In this case, you may want to research using multicast

as a more cost effective way of video distribution. When using multicast properly, streams are only sent to those who request it, and any additional requests for the same feed will not result in any increase in bandwidth.

Most streaming can be tunneled through a proxy server, so here again an authenticated proxy server (or firewall) is your best defense.

Skype and other VoIP services can be difficult to block, as they use various techniques to bypass firewalls. You can block the SIP port, which is UDP port 5060 for many VoIP clients, but almost all VoIP traffic is sent on randomized high UDP ports. An application layer firewall (such as l7-filter, *http:///l7-filter.sourceforge.net/*) can help detect and filter it.

Denial of Service

Denial of Service (***DoS***) attacks occur when an attacker sends a large number of connection requests to a server, flooding it and effectively disabling it. This will show up in your firewall logs, but by this point it will be too late. The only effective defense is to cut off the traffic further upstream; doing this will require the cooperation of your ISP.

Rogue DHCP servers

A misconfigured DHCP server, either by accident or intentionally malicious, can wreak havoc on a local area network. When a host sends a DHCP request on the local network, it accepts whichever response it receives the fastest. If the rogue DHCP server hands an incorrect address faster than your own DHCP server, it can potentially blackhole some of your clients.

Most of the time, a rogue DHCP server is either a misconfigured server or wireless router. Rogue DHCP servers are difficult to track down, but here are some symptoms to look for:

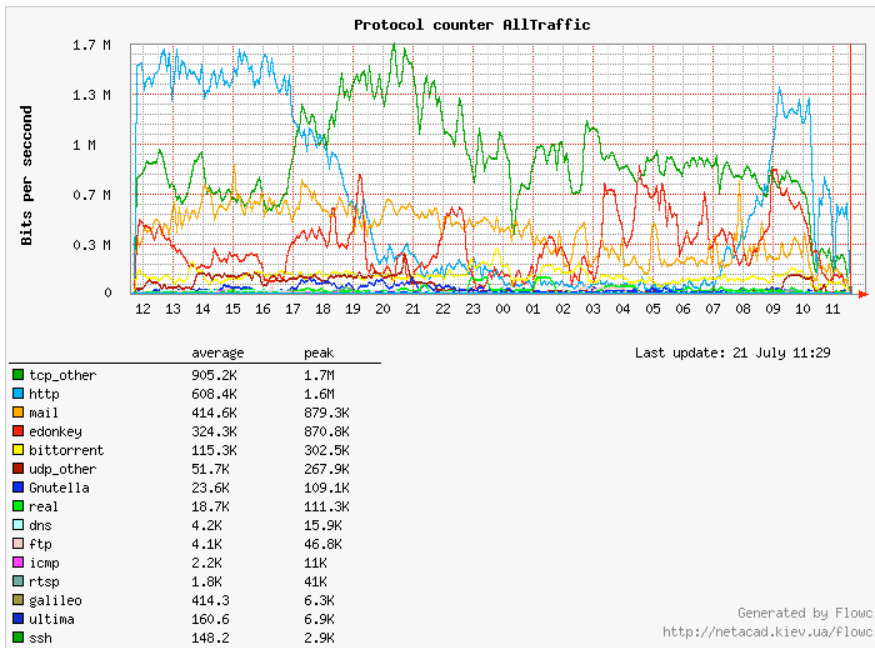
- Clients with improper IP addresses, netmasks, or gateways, even though your DHCP server is configured correctly.
- Some clients can communicate on the network, others cannot. Different IP addresses are being assigned to hosts on the same network.
- While sniffing network traffic, you see a DHCP response from a server IP address that you do not recognise.

Once you have determined the rogue DHCP server's MAC address from a packet trace, you can then make use of various layer 2 tracing techniques to determine the location of the rogue DHCP server, and isolate it.

There are several ways to prevent rogue DHCP servers from appearing on your network. First, educate your users on the dangers of misconfiguring or enabling DHCP services on your local LAN. Windows and UNIX systems engineers and users setting up access points on your local LAN should be careful not to place such a service on the local LAN. Second, some switching hardware platforms have layer 2 filtering capabilities to block DHCP responses from network interfaces that should never be connected to a DHCP server. On Cisco switching platforms, you may want to use the "dhcp snooping" feature set to specify trusted interfaces were DHCP responses can be transmitted. Apply these to server access ports and all uplink ports on your switch fabric.

Port analysis

There are several programs around that graphically display for you the network ports that are active. You can use this information to identify which ports need to be blocked at your firewall, proxy or router. However, some applications (like peer-to-peer programs) can masquerade on other ports, rendering this technique ineffective. In this case you would need a deep packet reader such as BWM Tools to analyse the application protocol information regardless of the port number. Figure 5.1 shows a graph from a protocol analyser known as FlowC.



	average	peak
tcp_other	905.2K	1.7M
http	608.4K	1.6M
mail	414.6K	879.3K
edonkey	324.3K	870.8K
bittorrent	115.3K	302.5K
udp_other	51.7K	267.9K
Gnutella	23.6K	109.1K
real	18.7K	111.3K
dns	4.2K	15.9K
ftp	4.1K	46.8K
icmp	2.2K	11K
rtsp	1.8K	41K
galileo	414.3	6.3K
ultima	160.6	6.9K
ssh	148.2	2.9K

Figure 5.1: Traffic utilisation broken down by protocol.

Browser prefetch

Some web browsers support a "prefetch" facility, which makes the browser download links on a web page before they are clicked on by the user. This functionality means that links can be displayed immediately, since the download has already taken place in the background. Browser prefetching will fetch many pages that the user will never view, thus consuming larger amounts of bandwidth than the user would otherwise require. Other than a marked increase in bandwidth use, this kind of behaviour is very difficult to detect. The only real response to this problem is to educate users, and explain that these tools can absorb large quantities of network bandwidth.

Benchmark your ISP

It is important to be sure that your Internet Service Provider has provided for you the level of service that you are paying for. One method of checking this is to test your connection speed to locations around the world. A list of servers around the world can be found at *http://www.dslreports.com/stest* . Another popular speed tester is *http://speedtest.net/*.

It is important to note that these tests have limitations. The tests are impacted by network conditions, both locally and across the entire route, at a particular moment in time. To obtain a full understanding, multiple tests should be run at different times of day, and with an understanding of local network conditions at the time.

Large downloads

A user may start several simultaneous downloads, or download large files such as 650MB ISO images. In this way, a single user can use up most of the bandwidth. The solutions to this kind of problem lie in training, offline downloading, and monitoring (including real-time monitoring, as outlined in chapter six). Offline downloading can be implemented in at least two ways:

- At the University of Moratuwa, a system was implemented using URL redirection. Users accessing *ftp://* URLs are served a directory listing in which each file has two links: one for normal downloading, and the other for offline downloading. If the offline link is selected, the specified file is queued for later download and the user notified by email when the download is complete. The system keeps a cache of recently downloaded files, and retrieves such files immediately when requested again. The download queue is sorted by file size. Therefore, small files are downloaded first. As some bandwidth is allocated to this system even during peak hours, users requesting small files may receive them within minutes, sometimes even faster than an online download.

- Another approach would be to create a web interface where users enter the URL of the file they want to download. This is then downloaded overnight using a ***cron job*** or scheduled task. This system would only work for users who are not impatient, and are familiar with what file sizes would be problematic for download during the working day.

Large uploads

When users need to transfer large files to collaborators elsewhere on the Internet, they should be shown how to schedule the upload. In Windows, an upload to a remote FTP server can be done using an FTP script file, which is a text file containing FTP commands, similar to the following (saved as **`c:\ftpscript.txt`**):

```
open ftp.ed.ac.uk
gventer
mysecretword
delete data.zip
binary
put data.zip
quit
```

To execute, type this from the command prompt:

```
ftp -s:c:\ftpscript.txt
```

On Windows NT, 2000 and XP computers, the command can be saved into a file such as **`transfer.cmd`**, and scheduled to run at night using the Scheduled Tasks (Start -> Settings -> Control Panel -> Scheduled Tasks). In Unix, the same can be achieved by using **`at`** or **`cron`**.

Users sending each other files

Users often need to send each other large files. It is a waste of bandwidth to send these via the Internet if the recipient is local. A file share should be created on a local file server, where a user can put the large file for others to access.

Alternatively, a web front-end can be written for a local web server to accept a large file and place it in a download area. After uploading it to the web server, the user receives a URL for the file. He can then give that URL to his local or international collaborators, and when they access that URL they can download it. This is what the University of Bristol has done with their ***FLUFF*** system. The University offers a facility for the upload of large files available from *http://www.bristol.ac.uk/fluff/*. These files can then be accessed by anyone who has been given their location. The advantage of this approach is that users can give external users access to their files, whereas the file share method can

work only for users within the campus network. A system like this can easily be implemented as a CGI script using Python and Apache.

Viruses and worms

Viruses are self-replicating computer programs that spread by copying themselves into (or ***infecting***) other files on your PC. Viruses are just one type of ***malware*** (or malicious software). Other types include ***worms*** and ***trojan horses***. Often, the term "virus" is used in the broader sense to include all types of malware. Network viruses spread by using popular network programs and protocols, such as SMTP, to spread themselves from one computer to another. Worms are similar to viruses in that they are spread from machine to machine, but worms typically do not have a malicious payload. While the goal of a virus is to steal data or damage computers, worms are simply intent on replicating as fast and as widely as possible. Trojan horses include software that masquerades as something useful (such as a utility or game), but secretly installs malware on your machine. Viruses and worms sometimes use trojan horses as a ***vector*** for infection. If a user can be tricked into double-clicking an email attachment or other program, then the programs can infect the user's machine.

Viruses can saturate a network with random traffic that can slow down a local area network and bring an Internet connection to a standstill. A virus will spread across your network much like the common flu will spread around a city. A network virus will typically start on a single PC, and then scan the entire local network looking for hosts to infect. It is this traffic that typically will kill the network. A single infected PC may not cause a big problem for the network, but as the number of infected PCs grows, the network traffic will grow exponentially. Another strategy viruses use is to send themselves through email. This type of virus will typically attempt to email itself to everyone in your address book with the intention of infecting them with the virus.

Recently, more diabolical viruses have been found that create vast ***bot networks***. These are used to send spam, perform DDoS attacks, or simply log traffic and send it back to a central location. These bots often use IRC as a control channel, where it receives further instructions. If your organisation does not rely on IRC for communication, it is prudent to filter IRC traffic using a firewall, proxy server, or with l7-filter.

So if a virus can be so detrimental to your network, how do you spot them? By keeping an eye on your bandwidth graph with programs like MRTG or Cacti, you can detect a sudden unexpected increase in traffic. You can figure out what type of traffic they are generating by using a protocol analyser.

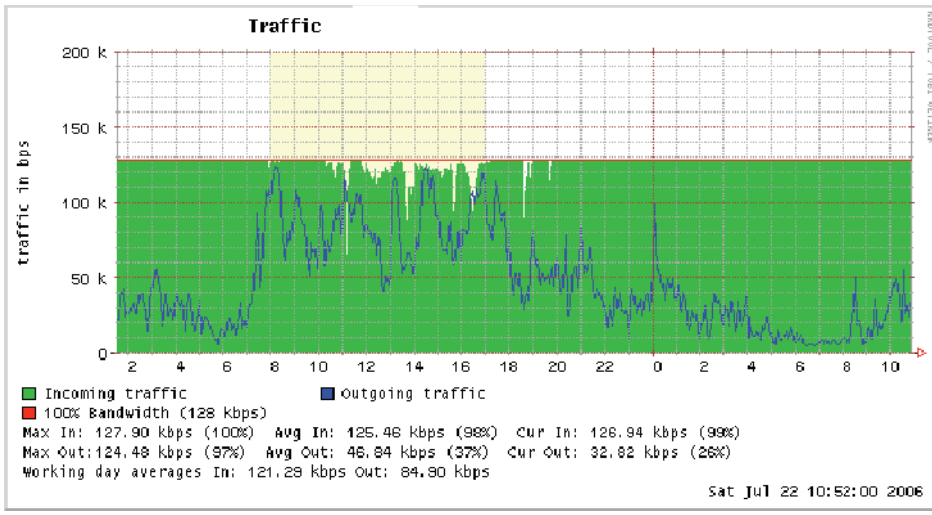


Figure 5.2: Something has been using all 128k of the inbound bandwidth for a period of several hours. But what is causing the problem?

Figure 5.2 shows an MRTG graph taken from an ISP with a 128 Kbps link that has been saturated by a certain virus. The virus was sending out massive amounts of traffic (the light grey area incoming), seen as inbound traffic to the ISP. This is a sign of trouble, and a protocol analyser confirms it.

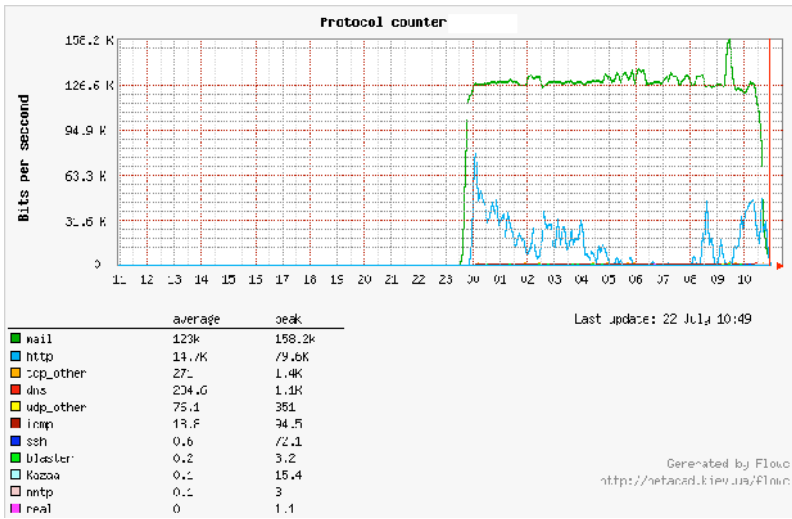


Figure 5.3: The highest line represents email utilisation. That definitely does not fit with the typical baseline usage for this site.

It is a mail virus known as Bagle. It is now sending out mass emails and may result in the IP getting blacklisted. The pie chart in figure 5.4 also reveals the damage done to browsing speeds.

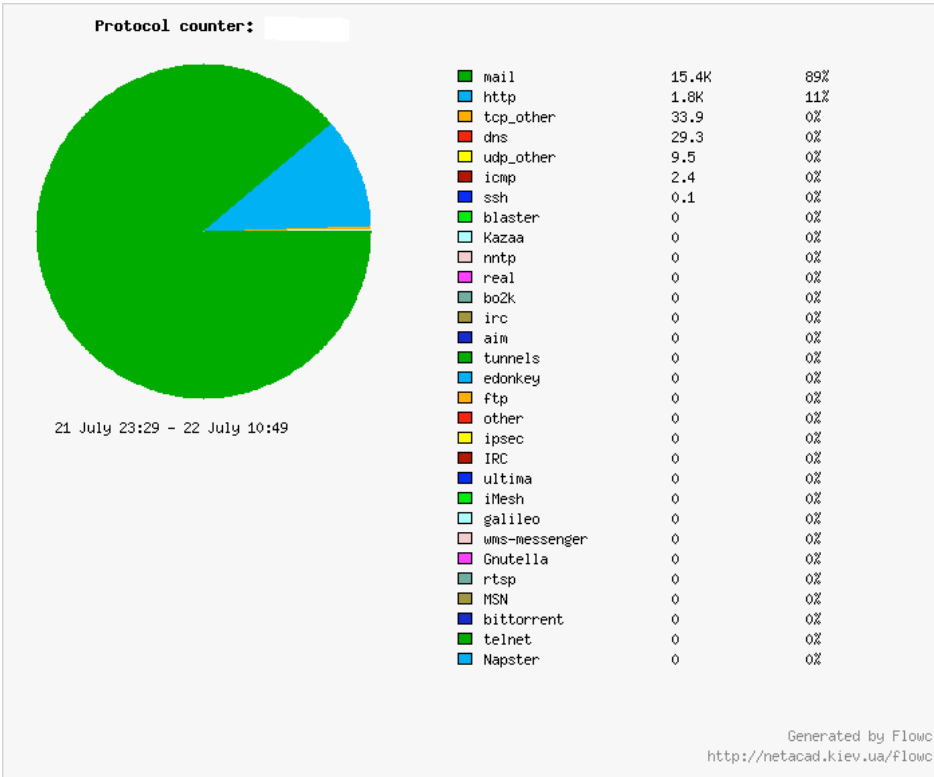


Figure 5.4: Email represents 89% of the total observed traffic on the link. Since typical usage for this site is < 5%, this is a clear indication of a virus.