3
Monitoring & Analysis

There's an old saying which applies to bandwidth management: "You can't manage it until you measure it." If your Internet connection is saturated with so much traffic that it makes your daily browsing seem like a trip to the dentist, you need to take a serious look at what is going down that pipe. Once you have a complete understanding of how your Internet connection is being used, it will become clear which course of action needs to be taken in order to fix the problem.

Without the insight that good monitoring tools and techniques provide, you cannot understand the effects that changes will make. Trying to fix network problems, without first establishing a clear picture of what is happening, is a lot like trying to fix a car engine by knocking on various parts with a hammer. You might get lucky and knock something into place that gets the car going again (for the moment), but you will inevitably run into more problems later. In the process of knocking on some parts, it's likely you will cause unintended damage to other parts of the engine.

Bandwidth management is not a dark art or a mystic philosophy; it is a methodical technique of problem identification, analysis, and resolution. By monitoring the performance of your network, and analysing the resulting data over time, you will be able to make effective changes that solve performance problems, yielding measurable improvements.

Before we can answer the question of where the network bottlenecks lie, we need to understand how the network works. Once we understand what makes information flow from here to there, we will have a better idea of what to look out for when that flow is not as fast as we would like it to be.

Networking 101

If you are already comfortable with the essentials of TCP/IP networking (including addressing, routing, switches, firewalls, and routers), you may want to skip ahead to **What is Network Monitoring?** on page **62**. We will now review the basics of Internet networking.

Introduction

Venice, Italy is a fantastic city to get lost in. The roads are mere foot paths that cross water in hundreds of places, and never go in a simple straight line. Postal carriers in Venice are some of the most highly trained in the world, specialising in delivery to only one or two of the six *sestieri* (districts) of Venice. This is necessary due to the intricate layout of that ancient city. Many people find that knowing the location of the water and the sun is far more useful than trying to find a street name on a map.

Figure 3.1: Another kind of network mask.

Just after the book development team met to formalize the outline for this book, a few of us spent a couple of days in Venice. One of us happened to find a particularly beautiful papier-mâché mask, and wanted to have it shipped from the studio in S. Polo, Venezia to an office in Seattle, USA. This may sound like an ordinary (or even trivial) task, but let's look at what actually happened.

The artist packed the mask into a shipping box and addressed it to the office in Seattle, USA. They then handed this off to a postal employee, who attached some official forms and sent it to a central package processing hub for international destinations. After several days, the package cleared Italian customs and found its way onto a transatlantic flight, arriving at a central import processing

location in the U.S. Once it was cleared through U.S. customs, the package was sent to the regional distribution point for the northwest U.S., then on to the Seattle postal processing centre. The package eventually made its way onto a delivery van which had a route that brought it to the proper address, on the proper street, in the proper neighborhood. A clerk at the office accepted the package and put it in the proper incoming mail box. Once it arrived, the box was retrieved and the mask itself was finally received.

The clerk at the office neither knows nor cares about how to get to the *sistiere* of S. Polo, Venezia. His job is simply to accept packages as they arrive, and deliver them to the proper person. Similarly, the postal carrier in Venice has no need to worry about how to get to the correct neighborhood in Seattle. His job is to pick up packages from his local neighborhood and forward them to the next closest hub in the delivery chain.

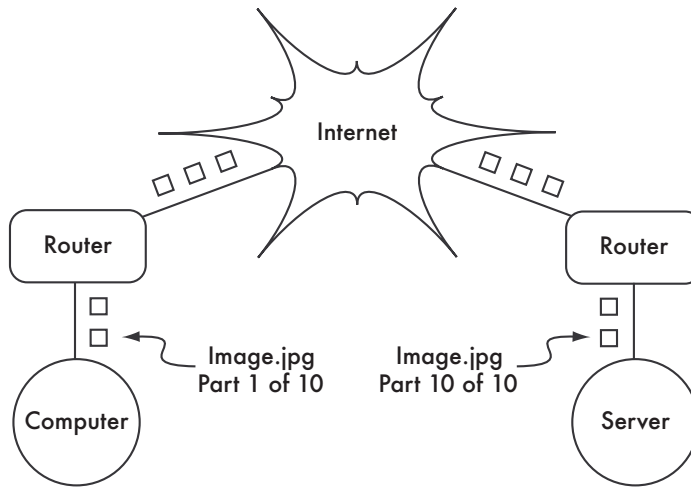


Figure 3.2: Internet networking. Packets are forwarded between routers until they reach their ultimate destination.

This is very similar to how Internet routing works. A particular message is split up into many individual ***packets***, and are labeled with their source and destination. The computer then sends these packets to a ***router***, which decides where to send them next. The router needs only to keep track of a handful of routes (for example, how to get to the local network, the best route to a few other local networks, and one route to a gateway to the rest of the Internet). This list of possible routes is called the ***routing table***. As packets arrive at the router, the destination address is examined and compared against its internal routing table. If the router has no explicit route to the destination in question, it sends the packet to the closest match it can find, which is often its own Internet gateway (via the default route). And the next router does the same, and so forth, until the packet eventually arrives at its destination.

Packages can only make their way through the international postal system because we have established a standardised addressing scheme for packages. For example, the destination address must be written legibly on the front of the package, and include all critical information (such as the recipient's name, street address, city, country, and postal code). Without this information, packages are either returned to the sender or are lost in the system.

Packets can only flow through the global Internet because we have agreed on a common addressing scheme and protocol for forwarding packets. These standard communication protocols make it possible to exchange information on a global scale.

Cooperative communications

Communication is only possible when the participants speak a common language. But once the communication becomes more complex than a simple conversation between two people, protocol becomes just as important as language. All of the people in an auditorium may speak English, but without a set of rules in place to establish who has the right to use the microphone, the communication of an individual's ideas to the entire room is nearly impossible. Now imagine an auditorium as big as the world, full of all of the computers that exist. Without a common set of communication protocols to regulate when and how each computer can speak, the Internet would be a chaotic mess where every machine tries to speak at once.

Of course, people have developed a number of communications frameworks to address this problem. The most well-known of these is the ***OSI model***.

The OSI model

The international standard for Open Systems Interconnection (OSI) is defined by the document ISO/IEC 7498-1, as outlined by the International Standards Organisation and the International Electrotechnical Commission. The full standard is available as publication "ISO/IEC 7498-1:1994," available from *http://standards.iso.org/ittf/PubliclyAvailableStandards/*.

The OSI model divides network traffic into a number of ***layers***. Each layer is independent of the layers around it, and each builds on the services provided by the layer below while providing new services to the layer above. The abstraction between layers makes it easy to design elaborate and highly reliable ***protocol stacks***, such as the ubiquitous ***TCP/IP*** stack. A protocol stack is an actual implementation of a layered communications framework. The OSI model doesn't define the protocols to be used in a particular network, but simply delegates each communications "job" to a single layer within a well-defined hierarchy.

While the ISO/IEC 7498-1 specification details how layers should interact with each other, it leaves the actual implementation details up to the manufacturer. Each layer can be implemented in hardware (more common for lower layers) or software. As long as the interface between layers adheres to the standard, implementers are free to use whatever means are available to build their protocol stack. This means that any given layer from manufacturer A can operate with the same layer from manufacturer B (assuming the relevant specifications are implemented and interpreted correctly).

Here is a brief outline of the seven-layer OSI networking model:

Layer	Name	Description
7	Application	The ***Application Layer*** is the layer that most network users are exposed to, and is the level at which human communication happens. HTTP, FTP, and SMTP are all application layer protocols. The human sits above this layer, interacting with the application.
6	Presentation	The ***Presentation Layer*** deals with data representation, before it reaches the application. This would include MIME encoding, data compression, formatting checks, byte ordering, etc.
5	Session	The ***Session Layer*** manages the logical communications session between applications. NetBIOS and RPC are two examples of a layer five protocol.
4	Transport	The ***Transport Layer*** provides a method of reaching a particular service on a given network node. Examples of protocols that operate at this layer are TCP and UDP. Some protocols at the transport layer (such as TCP) ensure that all of the data has arrived at the destination, and is reassembled and delivered to the next layer in the proper order. UDP is a "connectionless" protocol commonly used for video and audio streaming.

Layer	Name	Description
3	Network	IP (the Internet Protocol) is the most common ***Network Layer*** protocol. This is the layer where routing occurs. Packets can leave the link local network and be retransmitted on other networks. Routers perform this function on a network by having at least two network interfaces, one on each of the networks to be interconnected. Nodes on the Internet are reached by their globally unique IP address. Another critical Network Layer protocol is ICMP, which is a special protocol which provides various management messages needed for correct operation of IP. This layer is also sometimes referred to as the ***Internet Layer***.
2	Data Link	Whenever two or more nodes share the same physical medium (for example, several computers plugged into a hub, or a room full of wireless devices all using the same radio channel) they use the ***Data Link Layer*** to communicate. Common examples of data link protocols are Ethernet, Token Ring, ATM, and the wireless networking protocols (802.11a/b/g). Communication on this layer is said to be link-local, since all nodes connected at this layer communicate with each other directly. This layer is sometimes known as the ***Media Access Control*** (***MAC***) layer. On networks modeled after Ethernet, nodes are referred to by their ***MAC address.*** This is a unique 48 bit number assigned to every networking device when it is manufactured.
1	Physical	The ***Physical Layer*** is the lowest layer in the OSI model, and refers to the actual physical medium over which communications take place. This can be a copper CAT5 cable, a fibre optic bundle, radio waves, or just about any other medium capable of transmitting signals. Cut wires, broken fibre, and RF interference are all physical layer problems.

The layers in this model are numbered one through seven, with seven at the top. This is meant to reinforce the idea that each layer builds upon, and depends upon, the layers below. Imagine the OSI model as a building, with the foundation at layer one, the next layers as successive floors, and the roof at layer seven. If you remove any single layer, the building will not stand. Similarly, if the fourth floor is on fire, then nobody can pass through it in either direction.

The first three layers (Physical, Data Link, and Network) all happen "on the network." That is, activity at these layers is determined by the configuration of cables, switches, routers, and similar devices. A network switch can only distribute packets by using MAC addresses, so it need only implement layers one and two. A simple router can route packets using only their IP addresses, so it need implement only layers one through three. A web server or a laptop computer runs applications, so it must implement all seven layers. Some advanced routers may implement layer four and above, to allow them to make decisions based on the higher-level information content in a packet, such as the name of a website, or the attachments of an email.

The OSI model is internationally recognised, and is widely regarded as the complete and definitive network model. It provides a framework for manufacturers and network protocol implementers that can be used to build networking devices that interoperate in just about any part of the world.

From the perspective of a network engineer or troubleshooter, the OSI model can seem needlessly complex. In particular, people who build and troubleshoot TCP/IP networks rarely need to deal with problems at the Session or Presentation layers. For the majority of Internet network implementations, the OSI model can be simplified into a smaller collection of five layers.

The TCP/IP model

Unlike the OSI model, the TCP/IP model is not an international standard and its definitions vary. Nevertheless, it is often used as a pragmatic model for understanding and troubleshooting Internet networks. The vast majority of the Internet uses TCP/IP, and so we can make some assumptions about networks that make them easier to understand. The TCP/IP model of networking describes the following five layers:

Layer	Name
5	Application
4	Transport
3	Internet
2	Data Link
1	Physical

In terms of the OSI model, layers five through seven are rolled into the topmost layer (the Application layer). The first four layers in both models are identical.

Many network engineers think of everything above layer four as "just data" that varies from application to application. Since the first three layers are interoperable between virtually all manufacturers' equipment, and layer four works between all hosts using TCP/IP, and everything above layer four tends to apply to specific applications, this simplified model works well when building and troubleshooting TCP/IP networks. We will use the TCP/IP model when discussing networks in this book.

The TCP/IP model can be compared to a person delivering a letter to a downtown office building. The person first needs to interact with the road itself (the Physical layer), pay attention to other traffic on the road (the Data Link layer), turn at the proper place to connect to other roads and arrive at the correct address (the Internet layer), go to the proper floor and room number (the Transport layer), and finally give it to a receptionist who can take the letter from there (the Application layer). Once they have delivered the message to the receptionist, the delivery person is free to go on their way.

The five layers can be easily remembered by using the mnemonic "**P**lease **D**on't **L**ook **I**n **T**he **A**ttic," which of course stands for "**P**hysical / **D**ata **L**ink / **I**nternet / **T**ransport / **A**pplication."

The Internet protocols

TCP/IP is the protocol stack most commonly used on the global Internet. The acronym stands for ***Transmission Control Protocol*** (***TCP***) and ***Internet Protocol*** (***IP***), but actually refers to a whole family of related communications protocols. TCP/IP is also called the ***Internet protocol suite***, and it operates at layers three and four of the TCP/IP model.

In this discussion, we will focus on version four of the IP protocol (IPv4) as this is now the most widely deployed protocol on the Internet. What follows is a brief overview of the critical aspects of TCP/IP that are needed in order to understand network utilisation. For a more thorough treatment of this complex subject, see the resources at the end of this chapter.

IP Addressing

In an IPv4 network, the address is a 32-bit number, normally written as four 8-bit numbers expressed in decimal form and separated by periods. Examples of IP addresses are 10.0.17.1, 192.168.1.1, or 172.16.5.23.

If you enumerated every possible IP address, they would range from 0.0.0.0 to 255.255.255.255. This yields a total of more than four billion possible IP addresses (255 * 255 * 255 * 255 = 4 228 250 625); although many of these are reserved for special purposes and cannot be assigned to hosts. Each of the

usable IP addresses is a unique identifier that distinguishes one network node from another.

Interconnected networks must agree on an IP addressing plan. IP addresses must be unique and cannot be used in different places on the Internet at the same time; otherwise, routers would not know how best to route packets to them.

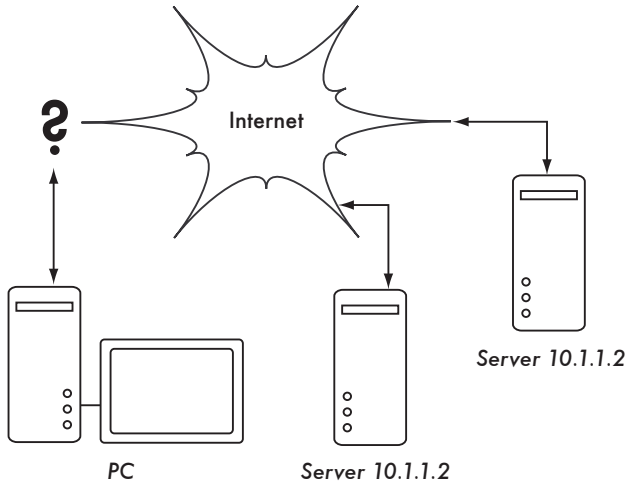


Figure 3.3: Without unique IP addresses, unambiguous global routing is impossible. If the PC requests a web page from 10.1.1.2, which server will it reach?

IP addresses are allocated by a central numbering authority, which provides a consistent and coherent numbering method. This ensures that duplicate addresses are not used by different networks. The authority assigns large blocks of consecutive addresses to smaller authorities, who in turn assign smaller consecutive blocks within these ranges to other authorities, or to their customers. These groups of addresses are called sub-networks, or ***subnets*** for short. Large subnets can be further subdivided into smaller subnets. A group of related addresses is referred to as an ***address space***.

Subnets

By applying a ***subnet mask*** (also called a ***network mask***, or simply ***netmask***) to an IP address, you can logically define both a host and the network to which it belongs. Traditionally, subnet masks have been expressed using dotted decimal form, much like an IP address. For example, 255.255.255.0 is one common netmask. You will find this notation used when configuring network interfaces, creating routes, etc. However, subnet masks are more succinctly expressed using ***CIDR notation***, which simply enumerates the number of bits in the mask after a forward slash (/). Thus, 255.255.255.0 can be simplified as

/24. CIDR is short for ***Classless Inter-Domain Routing***, and is defined in RFC1518*.

A subnet mask determines the size of a given network. Using a /24 netmask, 8 bits are reserved for hosts (32 bits total - 24 bits of netmask = 8 bits for hosts). This yields up to 256 possible host addresses (2^8 = 256). By convention, the first value is taken as the ***network address*** (.0 or 00000000), and the last value is taken as the ***broadcast address*** (.255 or 11111111). This leaves 254 addresses available for hosts on this network.

Subnet masks work by applying AND logic to the 32 bit IP number. In binary notation, the "1" bits in the mask indicate the network address portion, and "0" bits indicate the host address portion. A logical AND is performed by comparing two bits. The result is "1" if both of the bits being compared are also "1". Otherwise the result is "0". Here are all of the possible outcomes of a binary AND comparison between two bits.

Bit 1	Bit 2	Result
0	0	0
0	1	0
1	0	0
1	1	1

To understand how a netmask is applied to an IP address, first convert everything to binary. The netmask 255.255.255.0 in binary contains twenty-four "1" bits:

```
255      255      255      0
11111111.11111111.11111111.00000000
```

When this netmask is combined with the IP address 10.10.10.10, we can apply a logical AND to each of the bits to determine the network address.

```
  10.10.10.10: 00001010.00001010.00001010.00001010
255.255.255.0: 11111111.11111111.11111111.00000000
               -----------------------------------
   10.10.10.0: 00001010.00001010.00001010.00000000
```

* RFC is short for Request For Comments. RFCs are a numbered series of documents published by the Internet Society that document ideas and concepts related to Internet technologies. Not all RFCs are actual standards. RFCs can be viewed online at http://rfc.net/

This results in the network 10.10.10.0/24. This network consists of the hosts 10.10.10.1 through 10.10.10.254, with 10.10.10.0 as the network address and 10.10.10.255 as the broadcast address.

Subnet masks are not limited to entire octets. One can also specify subnet masks like 255.254.0.0 (or /15 CIDR). This is a large block, containing 131,072 addresses, from 10.0.0.0 to 10.1.255.255. It could be further subdivided, for example into 512 subnets of 256 addresses each. The first one would be 10.0.0.0-10.0.0.255, then 10.0.1.0-10.0.1.255, and so on up to 10.1.255.0-10.1.255.255. Alternatively, it could be subdivided into 2 blocks of 65,536 addresses, or 8192 blocks of 16 addresses, or in many different ways. It could even be subdivided into a mixture of different block sizes, as long as none of them overlap, and each is a valid subnet whose size is a power of two.

While many netmasks are possible, common netmasks include:

CIDR	Decimal	# of Hosts
/30	255.255.255.252	4
/29	255.255.255.248	8
/28	255.255.255.240	16
/27	255.255.255.224	32
/26	255.255.255.192	64
/25	255.255.255.128	128
/24	255.255.255.0	256
/16	255.255.0.0	65 536
/8	255.0.0.0	16 777 216

With each reduction in the CIDR value the IP space is doubled. Remember that two IP addresses within each network are always reserved for the network address and broadcast address.

There are three common netmasks that have special names. A /8 network (with a netmask of 255.0.0.0) defines a ***Class A*** network. A /16 (255.255.0.0) is a ***Class B***, and a /24 (255.255.255.0) is called a ***Class C***. These names were around long before CIDR notation, but are still often used for historical reasons.

Global IP Addresses

Have you ever wondered who controls the allocation of IP space? ***Globally routable IP addresses*** are assigned and distributed by ***Regional Internet Registrars*** (***RIR***s) to ISPs. The ISP then allocates smaller IP blocks to their clients as required. Virtually all Internet users obtain their IP addresses from an ISP.

The 4 billion available IP addresses are administered by the ***Internet Assigned Numbers Authority*** (***IANA***, *http://www.iana.org/*). IANA has divided this space into large subnets, usually /8 subnets with 16 million addresses each. These subnets are delegated to one of the five regional Internet registries (RIRs), which are given authority over large geographic areas.

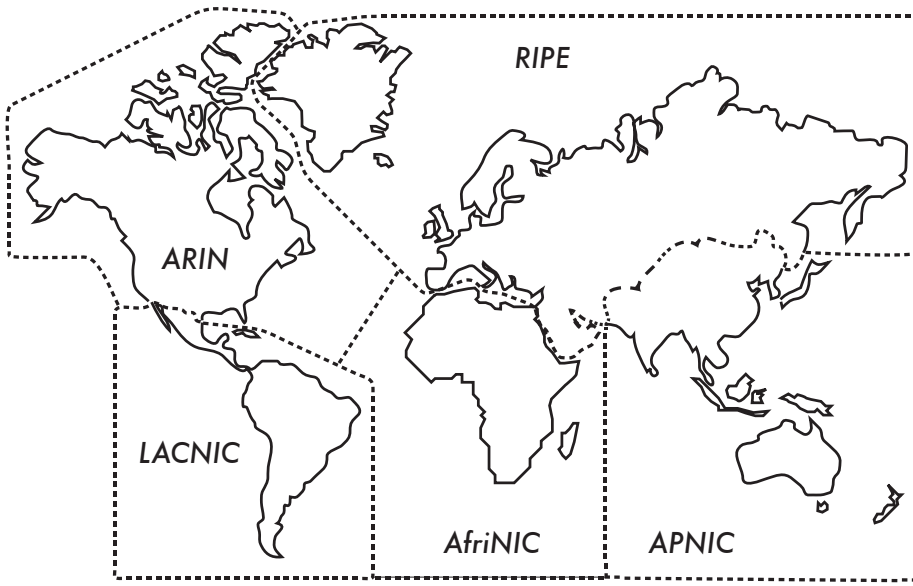


Figure 3.4: Authority for Internet IP address assignments is delegated to the five Regional Internet Registrars.

The five RIRs are:

- African Network Information Centre (AfriNIC, *http://www.afrinic.net/*)
- Asia Pacific Network Information Centre (APNIC, *http://www.apnic.net/*)
- American Registry for Internet Numbers (ARIN, *http://www.arin.net/*)
- Regional Latin-American and Caribbean IP Address Registry (LACNIC, *http://lacnic.net/*)
- Réseaux IP Européens (RIPE NCC, *http://www.ripe.net/*)

Your ISP will assign globally routable IP address space to you from the pool allocated to it by your RIR. The registry system assures that IP addresses are not reused in any part of the network anywhere in the world.

Once IP address assignments have been agreed upon, it is possible to pass packets between networks and participate in the global Internet. The process of moving packets between networks is called ***routing***.

Static IP Addresses

A static IP address is an address assignment that never changes. Static IP addresses are important because servers using these addresses may have DNS mappings pointed towards them, and typically serve information to other machines (such as email services, web servers, etc.).

Blocks of static IP addresses may be assigned by your ISP, either by request or automatically depending on your means of connection to the Internet.

Dynamic IP Addresses

Dynamic IP addresses are assigned by an ISP for non-permanent nodes connecting to the Internet, such as a home computer which is on a dial-up connection.

Dynamic IP addresses can be assigned automatically using the ***Dynamic Host Configuration Protocol*** (***DHCP***), or the ***Point-to-Point Protocol*** (***PPP***), depending on the type of Internet connection. A node using DHCP first requests an IP address assignment from the network, and automatically configures its network interface. IP addresses can be assigned randomly from a pool by your ISP, or might be assigned according to a policy. IP addresses assigned by DHCP are valid for a specified time (called the ***lease time***). The node must renew the DHCP lease before the lease time expires. Upon renewal, the node may receive the same IP address or a different one from the pool of available addresses.

Dynamic addresses are popular with Internet service providers, because it enables them to have fewer IP addresses than their total number of customers. They only need an address for each customer who is **active at any one time**. Globally routable IP addresses cost money, and some authorities that specialise in the assignment of addresses (such as RIPE, the European RIR) are very strict on IP address usage for ISP's. Assigning addresses dynamically enables ISPs to save money, and normally they will charge extra for a static IP address.

Private IP addresses

Most private networks do not require the allocation of a globally routable, public IP addresses for every computer in the organisation. In particular, computers which are not public servers do not need to be addressable from the public Internet. Organisations typically use IP addresses from the ***private address space*** for machines on the internal network.

There are currently three blocks of private address space reserved by IANA: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16. These are defined in RFC1918. These addresses are not intended to be routed on the Internet, and are typically unique only within an organisation or group of organisations which choose to follow the same numbering scheme.

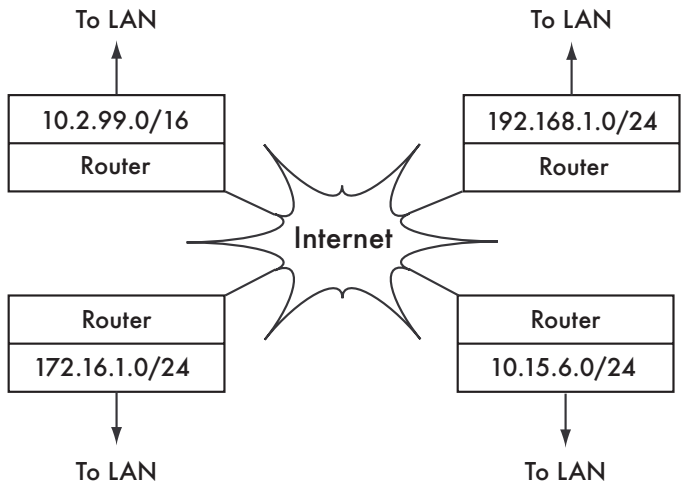


Figure 3.5: RFC1918 private addresses may be used within an organisation, and are not routed on the global Internet.

If you ever intend to link together private networks that use RFC1918 address space, be sure to use unique addresses throughout all of the networks. For example, you might break the 10.0.0.0/8 address space into multiple Class B networks (10.1.0.0/16, 10.2.0.0/16, etc.). One block could be assigned to each network according to its physical location (the campus main branch, field office one, field office two, dormitories, and so forth). The network administrators at each location can then break the network down further into multiple Class C networks (10.1.1.0/24, 10.1.2.0/24, etc.). or into blocks of any other logical size. In the future, should the networks ever be linked (either by a physical connection, wireless link, or VPN), then all of the machines will be reachable from any point in the network without having to renumber network devices.

Some Internet providers may allocate private addresses like these instead of public addresses to their customers, although this has serious disadvantages.

Since these addresses cannot be routed over the Internet, computers which use them are not really "part" of the Internet, and are not directly reachable from it. In order to allow them to communicate with the Internet, their private addresses must be translated to public addresses. This translation process is known as ***Network Address Translation*** (***NAT***), and is normally performed at the gateway between the private network and the Internet. We will look at NAT in more detail on page **42**.

Routing

Imagine a network with three hosts: A, B, and C. They use the corresponding IP addresses 192.168.1.1, 192.168.1.2 and 192.168.1.3. These hosts are part of a /24 network (their network mask is 255.255.255.0).

For two hosts to communicate on a local network, they must determine each others' MAC addresses. It is possible to manually configure each host with a mapping table from IP address to MAC address, but normally the ***Address Resolution Protocol*** (***ARP***) is used to determine this automatically.

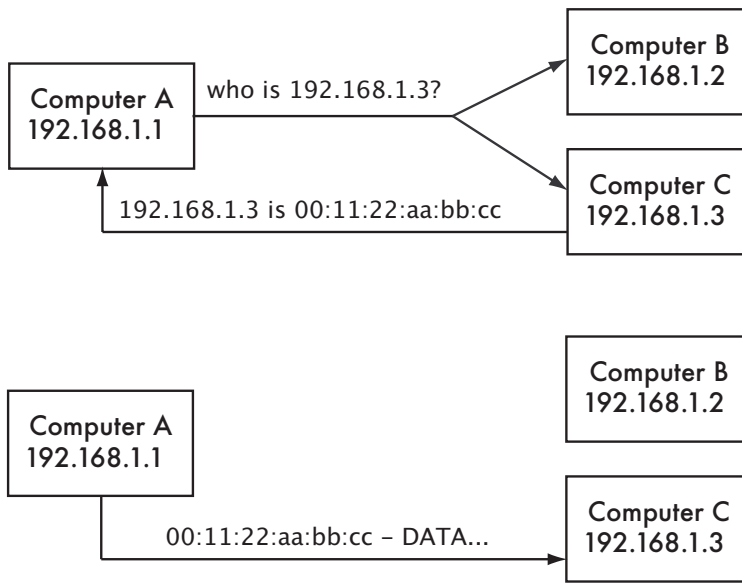


Figure 3.6: Computer A needs to send data to 192.168.1.3. But it must first ask the whole network for the MAC address that responds to 192.168.1.3.

When using ARP, host A broadcasts to all hosts the question, "Who has the MAC address for the IP 192.168.1.3?" When host C sees an ARP request for its own IP address, it replies with its MAC address.

Consider now another network with 3 hosts, D, E, and F, with the corresponding IP addresses 192.168.2.1, 192.168.2.2, and 192.168.2.3. This is another

/24 network, but it is not in the same range as the network above. All three hosts can reach each other directly (first using ARP to resolve the IP address into a MAC address, and then sending packets to that MAC address).

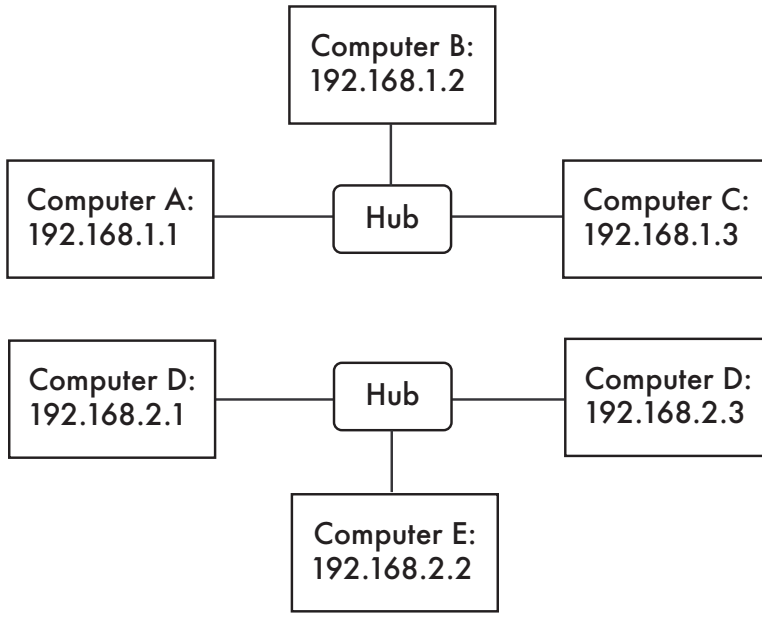


Figure 3.7: Two separate IP networks.

Now we will add host G. This host has two network cards, with one plugged into each network. The first network card uses the IP address 192.168.1.4, and the other uses 192.168.2.4. Host G is now link-local to both networks, and can route packets between them.

But what if hosts A, B, and C want to reach hosts D, E, and F? They will need to add a route to the other network via host G. For example, hosts A-C would add the following route:

```
# ip route add 192.168.2.0/24 via 192.168.1.4
```

...and hosts D-F would add the following:

```
# ip route add 192.168.1.0/24 via 192.168.2.4
```

(These examples use the Linux syntax for manipulating routes, which varies according to your operating system). The result is shown in Figure 3.8. Notice that the route is added via the IP address on host G that is link-local to the respective network. Host A could not add a route via 192.168.2.4, even though it is the same physical machine as 192.168.1.4 (host G), since that IP is not link-local.

A route tells the OS that the desired network doesn't lie on the immediate link-local network, and it must ***forward*** the traffic through the specified router. If host A wants to send a packet to host F, it would first send it to host G. Host G would then look up host F in its routing table, and see that it has a direct connection to host F's network. Finally, host G would resolve the hardware (MAC) address of host F and forward the packet to it.

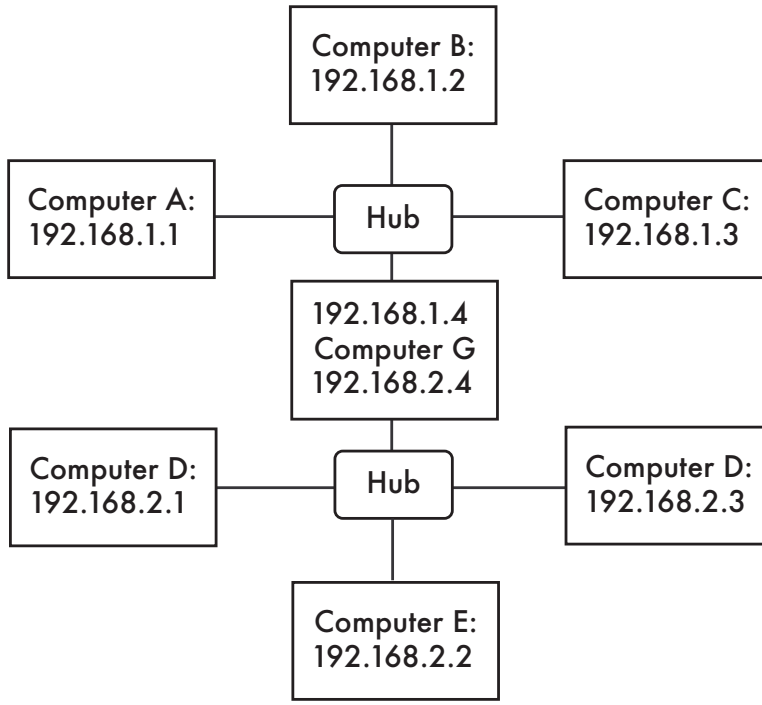
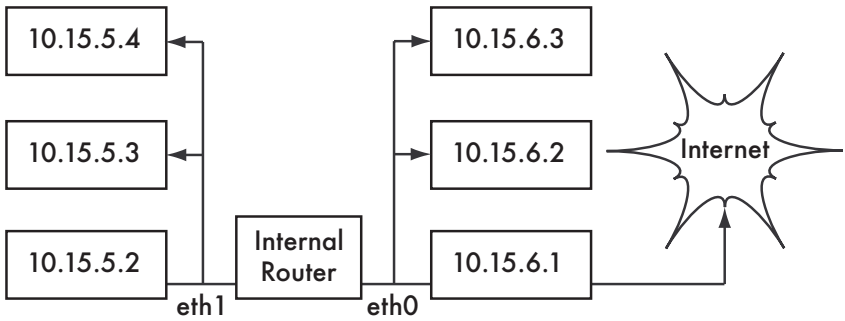


Figure 3.8: Host G acts as a router between the two networks.

This is a very simple routing example, where the destination is only a single ***hop*** away from the source. As networks get more complex, many hops may need to be made to reach the ultimate destination. Since it isn't practical for every machine on the Internet to know the route to every other, we make use of a routing entry known as the ***default route*** (also known as the ***default gateway***). When a router receives a packet destined for a network for which it has no explicit route, the packet is forwarded to its default gateway.

The default gateway is typically the best route out of your network, usually in the direction of your ISP. An example of a router that uses a default gateway is shown in Figure 3.9.




```
Routing table for internal router:

Destination     Gateway     Genmask         Flags  Metric  Iface
10.15.5.0       *           255.255.255.0   U      0       eth1
10.15.6.0       *           255.255.255.0   U      0       eth0
default         10.15.6.1   0.0.0.0         UG     0       eth0
```

Figure 3.9: When no explicit route exists to a particular destination, a host uses the default gateway entry in its routing table.

Routes can be updated manually, or can dynamically react to network outages and other events. Some examples of popular dynamic routing protocols are RIP, OSPF, BGP, and OLSR. Configuring dynamic routing is beyond the scope of this book, but for further reading on the subject, see the resources at the end of this chapter.

Network Address Translation (NAT)

In order to reach hosts on the Internet, RFC1918 addresses must be converted to global, publicly routable IP addresses. This is achieved using a technique known as ***Network Address Translation***, or ***NAT***. A NAT device is a router that manipulates the addresses of packets instead of simply forwarding them. On a NAT router, the Internet connection uses one (or more) globally routed IP addresses, while the private network uses an IP address from the RFC1918 private address range. The NAT router allows the global address(es) to be shared with all of the inside users, who all use private addresses from the same range. It converts the packets from one form of addressing to the other as the packets pass through it. As far as the network users can tell, they are directly connected to the Internet and require no special software or drivers. They simply use the NAT router as their default gateway, and address packets as they normally would. The NAT router translates outbound packets to use the global IP address as they leave the network, and translates them back again as they are received from the Internet.

The major consequence of using NAT is that machines from the Internet cannot easily reach servers within the organisation without setting up explicit forward-

ing rules on the router. Connections initiated from within the private address space generally have no trouble, although some applications (such as Voice over IP and some VPN software) can have difficulty dealing with NAT.

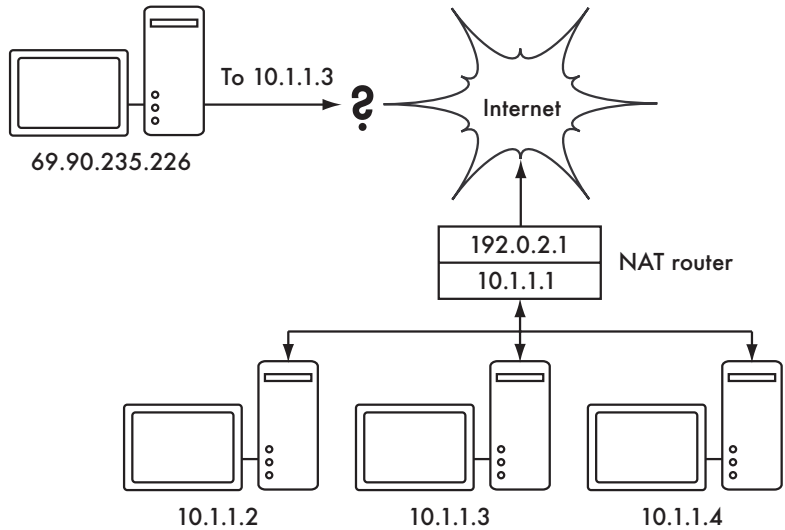


Figure 3.10: Network Address Translation allows you to share a single IP address with many internal hosts, but can make it difficult for some services to work properly.

Depending on your point of view, this can be considered a bug (since it makes it harder to set up two-way communication) or a feature (since it effectively provides a "free" firewall for your entire organisation). RFC1918 addresses should be filtered on the edge of your network to avoid accidental or intentional RFC1918 traffic entering or leaving your network. While NAT performs some firewall-like functions, it is not a replacement for a real firewall.

Internet Protocol Suite

Machines on the Internet use the Internet Protocol (IP) to reach each other, even when separated by many intermediary machines. There are a number of protocols that are run in conjunction with IP that provide features as critical to normal operations as IP itself. Every packet specifies a protocol number which identifies the packet as one of these protocols. The most commonly used protocols are the ***Transmission Control Protocol*** (***TCP***, number 6), ***User Datagram Protocol*** (***UDP***, number 17), and the ***Internet Control Message Protocol*** (***ICMP***, number 1). Taken as a group, these protocols (and others) are known as the ***Internet Protocol Suite***, or simply ***TCP/IP*** for short.

The TCP and UDP protocols introduce the concept of port numbers. These allow multiple services to be run on the same IP address, and still be distinguished from each other. Every packet has a source and destination port num-

ber. Some port numbers are well defined standards, used to reach well known services such as email and web servers. For example, web servers normally ***listen*** on TCP port 80, and SMTP email servers listen on TCP port 25. When we say that a service "listens" on a port (such as port 80), we mean that it will accept packets that use its IP as the destination IP address, and 80 as the destination port. Servers usually do not care about the source IP or source port, although sometimes they will use them to establish the identity of the other side. When sending a response to such packets, the server will use its own IP as the source IP, and 80 as the source port.

When a client connects to a service, it may use any source port number on its side which is not already in use, but it must connect to the proper port on the server (e.g. 80 for web, 25 for email). TCP is a ***session oriented*** protocol with guaranteed delivery and transmission control features (such as detection and mitigation of network congestion, retries, packet reordering and reassembly, etc.). UDP is designed for ***connectionless*** streams of information, and does not guarantee delivery at all, or in any particular order.

The ICMP protocol is designed for debugging and maintenance on the Internet. Rather than port numbers, it has ***message types***, which are also numbers. Different message types are used to request a simple response from another computer (echo request), notify the sender of another packet of a possible routing loop (time exceeded), or inform the sender that a packet that could not be delivered due to firewall rules or other problems (destination unreachable).

By now you should have a solid understanding of how computers on the network are addressed, and how information flows on the network between them. Now let's take a brief look at the physical hardware that implements these network protocols.

Networking hardware

Before you can monitor the performance of your network, you first need to understand the capabilities of the network hardware. Can your router keep up with the bandwidth provided by your ISP? Is there a bottleneck at your internal network interconnections? You should know how your hardware will respond to demands for network resources so you understand the hard limits of what the network can provide. No amount of protocol optimisation or caching can help your Internet performance if your networking hardware simply cannot keep up with the demand.

Ethernet

Ethernet is the name of the most popular standard for connecting together computers on a ***Local Area Network*** (***LAN***). It is sometimes used to connect individual computers to the Internet, via a router, ADSL modem, or wireless de-

vice. However, if you connect a single computer to the Internet, you may not use Ethernet at all. The name comes from the physical concept of the ether, the medium which was once supposed to carry light waves through free space. The official standard is called IEEE 802.3.

The most common Ethernet standard is called 100baseT. This defines a data rate of 100 megabits per second, running over twisted pair wires, with modular RJ-45 connectors on the end. The network topology is a star, with switches or hubs at the centre of each star, and end nodes (devices and additional switches) at the edges.

MAC addresses

Every device connected to an Ethernet network has a unique MAC address, assigned by the manufacturer of the network card. Its function is like that of an IP address, since it serves as a unique identifier that enables devices to talk to each other. However, the scope of a MAC address is limited to a broadcast domain, which is defined as all the computers connected together by wires, hubs, switches, and bridges, but not crossing routers or Internet gateways. MAC addresses are never used directly on the Internet, and are not transmitted across routers.

Hubs

Ethernet ***hubs*** connect multiple twisted-pair Ethernet devices together. They work at the physical layer (the lowest or first layer). They repeat the signals received by each port out to all of the other ports. Hubs can therefore be considered to be simple repeaters. Due to this design, only one port can successfully transmit at a time. If two devices transmit at the same time, they corrupt each other's transmissions, and both must back off and retransmit their packets later. This is known as a ***collision***, and each host remains responsible for detecting collisions during transmission, and retransmitting its own packets when needed.

When problems such as excessive collisions are detected on a port, some hubs can disconnect (***partition***) that port for a while to limit its impact on the rest of the network. While a port is partitioned, devices attached to it cannot communicate with the rest of the network. Hub-based networks are generally more robust than coaxial Ethernet (also known as 10base2 or ThinNet), where misbehaving devices can disable the entire segment. But hubs are limited in their usefulness, since they can easily become points of congestion on busy networks.

Switches

A ***switch*** is a device which operates much like a hub, but provides a dedicated (or ***switched***) connection between ports. Rather than repeating all traffic on every port, the switch determines which ports are communicating directly and connects them together. Switches generally provide much better performance than hubs, especially on busy networks with many computers. They are not much more expensive than hubs, and are replacing them in many situations.

Switches work at the data link layer (the second layer), since they interpret and act upon the MAC address in the packets they receive. When a packet arrives at a port on a switch, it makes a note of the source MAC address, which it associates with that port. It stores this information in an internal ***MAC table***. The switch then looks up the destination MAC address in its MAC table, and transmits the packet on the matching port. If the destination MAC address is not found in the MAC table, the packet is then sent to all of the connected interfaces. If the destination port matches the incoming port, the packet is filtered and is not forwarded.

Hubs vs. Switches

Hubs are considered to be fairly unsophisticated devices, since they inefficiently rebroadcast all traffic on every port. This simplicity introduces both a performance penalty and a security issue. Overall performance is slower, since the available bandwidth must be shared between all ports. Since all traffic is seen by all ports, any host on the network can easily monitor all of the network traffic.

Switches create virtual connections between receiving and transmitting ports. This yields better performance because many virtual connections can be made simultaneously. More expensive switches can switch traffic by inspecting packets at higher levels (at the transport or application layer), allow the creation of VLANs, and implement other advanced features.

A hub should be used when repetition of traffic on all ports is desirable; for example, when you want to explicitly allow a monitoring machine to see all of the traffic on the network. Most switches provide ***monitor port*** functionality that enables repeating on an assigned port specifically for this purpose.

Hubs were once cheaper than switches. However, the price of switches have reduced dramatically over the years. Therefore, old network hubs should be replaced whenever possible with new switches.

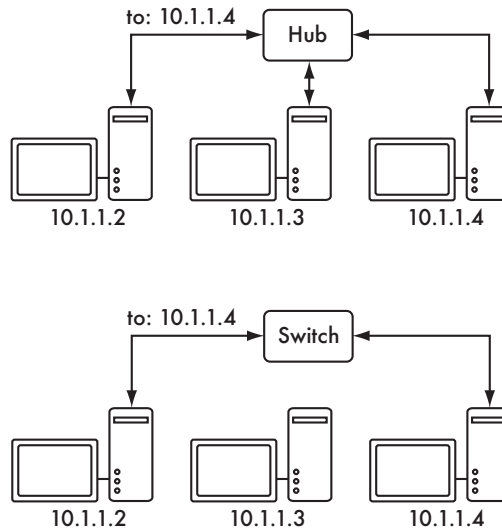


Figure 3.11: A hub simply repeats all traffic on every port, while a switch makes a temporary, dedicated connection between the ports that need to communicate.

Both hubs and switches may offer ***managed*** services. Some of these services include the ability to set the link speed (10baseT, 100baseT, 1000baseT, full or half duplex) per port, enable triggers to watch for network events (such as changes in MAC address or malformed packets), and usually include ***port counters*** for easy bandwidth accounting. A managed switch that provides upload and download byte counts for every physical port can greatly simplify network monitoring. These services are typically available via SNMP, or they may be accessed via telnet, ssh, a web interface, or a custom configuration tool.

Routers, firewalls, and NAT

While hubs and switches provide connectivity on a local network segment, a router's job is to forward packets between different network segments. A router typically has two or more physical network interfaces. It may include support for different types of network media, such as Ethernet, ATM, DSL, or dial-up. Routers can be dedicated hardware devices (such as Cisco or Juniper routers) or they can be made from a standard PC with multiple network cards and appropriate software.

Routers sit at the ***edge*** of two or more networks. By definition, they have one connection to each network, and as border machines they may take on other responsibilities as well as routing. Many routers have ***firewall*** capabilities that provide a mechanism to filter or redirect packets that do not fit security or access policy requirements. They may also provide Network Address Translation (NAT) services.

Routers vary widely in cost and capabilities. The lowest cost and least flexible are simple, dedicated hardware devices, often with NAT functionality, used to cheaply share an Internet connection between multiple computers. The next step up is a software router, which consists of an operating system running on a standard PC with multiple network interfaces. Standard operating systems such as Microsoft Windows, Linux, and BSD are all capable of routing, and are much more flexible than the low-cost hardware devices. However, they suffer from the same problems as conventional PCs, with high power consumption, a large number of complex and potentially unreliable parts, and more involved configuration.

The most expensive devices are high-end dedicated hardware routers, made by companies like Cisco and Juniper. They tend to have much better performance, more features, and higher reliability than software routers on PCs. It is also possible to purchase technical support and maintenance contracts for them.

Most modern routers offer mechanisms to monitor and record performance remotely, usually via the Simple Network Management Protocol (SNMP), although the least expensive devices often omit this feature.

Other equipment

Each physical network has an associated piece of terminal equipment. For example, VSAT connections consist of a satellite dish connected to a terminal that either plugs into a card inside a PC, or ends at a standard Ethernet connection. DSL lines use a ***DSL modem*** that bridges the telephone line to a local device, either an Ethernet network or a single computer via USB. ***Cable modems*** bridge the television cable to Ethernet, or to an internal PC card bus. Some kinds of telecom circuit (such as a T1 or T3) use a CSU/DSU to bridge the circuit to a serial port or Ethernet. Standard dialup lines use modems to connect a computer to the telephone, usually via a plug-in card or serial connection. And there are many different kinds of wireless networking equipment that connect to a variety of radios and antennas, but nearly always end at an Ethernet jack.

The functionality of these devices can vary significantly between manufacturers. Some provide mechanisms for monitoring performance, while others may not. Since your Internet connection ultimately comes from your ISP, you should follow their recommendations when choosing equipment that bridges their network to your Ethernet network.

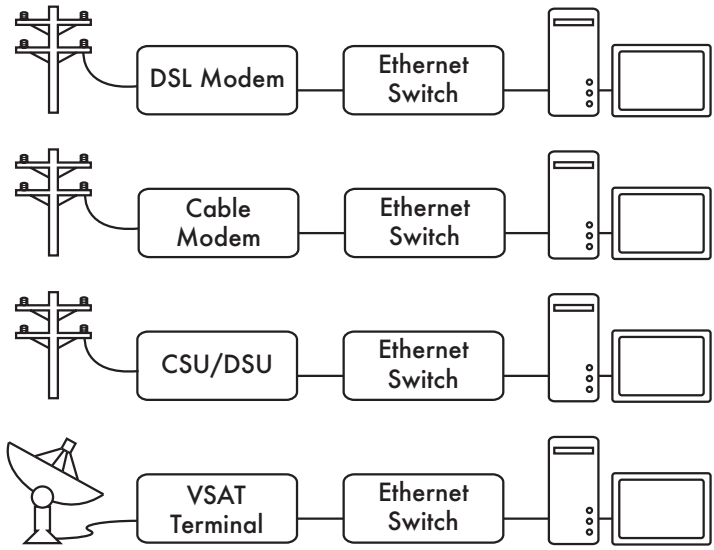


Figure 3.12: Many DSL modems, cable modems, CSU/DSUs, wireless access points, and VSAT terminals terminate at an Ethernet jack.

All components and devices should be kept in mind when analysing the network for potential bottlenecks.

Physical connectivity

Connections between computers are usually serial, that is, they transmit one bit at a time. The speed of the connection is measured in ***bits per second***, also known as ***bps***. There are eight bits in a byte, so the speed in bytes per second is eight times lower. For example, a 56,000 bps connection can transmit up to 7,000 bytes per second (Bps), which is a little less than seven kilobytes per second (7 kBps or kB/s).

The speed of the connection directly affects how fast it feels to the user. For example, the average web page is around 100 kilobytes (kB), and at 7 kBps, it takes just over 14 seconds to load. An email can be between 4 kilobytes and 2 megabytes, depending on the size of attachments, and take between half a second and five minutes to download. The information on a full CD-ROM, 680 megabytes, would take nearly 27 hours.

These examples ignore protocol overhead for the sake of simplicity. Each kind of connection necessarily adds a small amount of protocol overhead in order to maintain itself. This overhead reduces the available throughput and introduces some mandatory latency. Running TCP/IP on the connection further reduces the overall available throughput.

The Internet is a collection of many different networks, connected to each other in almost as many different ways. There is no single standard way to connect. In the early days, most networks were connected to each other by modems, but now a variety of digital and analogue technologies for interconnection exist. Each has different properties including price, range, speed, and quality, so choosing a connection method can be difficult. We will start with the oldest and slowest.

Telephone Modem

Technically, a ***modem*** is any device which converts digital signals to analogue and back, which includes almost every type of communications equipment. However, most people think of a modem as a box that connects a computer to a telephone line. This technology is old and slow, but cheap and reliable, and there are probably still more users connected to the Internet by telephone modems than any other method.

Modems come in two types: ***internal*** and ***external***. Internal modems are cards that fit inside a computer, and add a telephone port to the back panel of the computer. External modems are boxes which sit between the computer and the telephone line. Older external modems have RS-232 serial ports, usually with 25-pin D connectors, but modern external modems have USB connectors instead.

Modems send both commands and data over the same serial lines. Every modem has a slightly different set of commands, although they are all based on a standard, the ***Hayes AT command set***. Using a modem on a Windows or Macintosh computer requires driver software on the computer that knows which commands to send to it, while on Unix-based systems, you must specify the commands yourself. The supported command set is usually found in the modem's manual. Although standard AT commands will usually work, you will usually get better performance by using the right commands for the particular modem. This allows you to enable its high-speed modes and error-correction features. Some examples of advanced AT strings are provided in the **Performance Tuning** chapter, on page **177**.

There are many standards for modems that have been developed over their long life, usually to improve performance over previous models. The fastest modems today support the V.92 standard, which gives a maximum theoretical speed of 56,000 bits per second. Unfortunately, achieving this speed requires a digital telephone exchange and perfect laboratory conditions, so most connections are made at between 40,000 and 51,000 bps. On lower quality telephone lines, the maximum achievable speed may be 28,800 bps or less.

A modem connects to the public telephone network, and your computer connects to the Internet by using a modem to make a telephone call to another

modem at the Internet provider. Therefore, you pay for a telephone call for the entire time that you are online. Since telephone calls are usually expensive, modems are usually used to ***dial on demand***, in other words to connect only when users actually need access to the Internet. Computers can be configured to dial regularly to collect mail even when unattended, for example in the middle of the night when calls are cheaper or more reliable.

ISDN

Integrated Services Digital Network (***ISDN***) is the direct digital equivalent of the analogue telephone line. The line uses pure digital signaling over two wires on the public side of the network, which terminates at a network terminator (NT1) box. This is then connected to an ISDN modem or digital telephone using four wires. The devices that connect computers to ISDN lines are not really modems, since the signaling is digital on both sides, but they look like modems physically and to the computer.

ISDN requires a digital telephone exchange, and is usually more expensive to install than an analogue telephone line, but it provides higher speed. The line has two "B" channels for data, and a "D" channel for control. The "B" channels can each carry a voice conversation, or up to 64,000 bps data, much faster and more reliably than analogue lines. It is usually possible to "bond" both B channels together to achieve 128,000 bps data, but this requires two telephone calls and is therefore twice as expensive, and not all Internet providers support it.

Leased Lines

A ***leased line*** is the equivalent of a physical wire between two points. Normally, the line is owned by a telephone company and is leased to a customer. It can be thought of as a telephone line that permanently connects two fixed points, usually no more than a few kilometres apart. It is usually priced similarly to a permanent telephone call, although prices have been forced down somewhat by competing technologies.

Leased lines vary widely in speed, from 64 Kbps to 10 Mbps. Standard speeds are T1 (1.5 Mbps) and E1 (2 Mbps). They are very simple technologies, are considered very reliable, and can be very fast, so many companies use them to connect to the Internet and between their offices. Leased lines are often used to host servers because of their speed and reliability, so Internet providers usually offer reasonably large numbers of static IP addresses with a leased line connection. On the other hand, they are very expensive, and many smaller companies have switched to competing technologies such as ADSL or wireless Internet.

A leased line normally comes onto your premises as a pair of wires, much like a telephone connection. If you own both ends, you can attach a modem, tin cups,

or almost anything you like to the two ends. However, the effective range of leased lines is limited. If you connect to an Internet provider who is outside the range of what a leased line can reach, then then you may instead connect to the line provider's packet switching network, usually a ***frame relay*** network, which forwards the packets to your ISP. To do this, you will need to connect a device that understands the frame relay protocol to your end of the leased line. Frame relay is a Layer 2 (data link) network protocol, similar in purpose to Ethernet. Frame relay equipment is normally expensive, costing thousands of dollars per site. It may eventually be replaced by Ethernet.

Fibre Optic

Optical fibres are cables made from glass rather than copper. They carry information using visible light or infra-red rather than electrical potentials. They have very low signal loss and interference, even over very long distances, and are usually used for communications over distances greater than 5 km.

There are two standards for optical communication: ***Synchronous Digital Hierarchy*** (***SDH***) and Ethernet. SDH (and its precursor, ***SONET***, which is still in use in some parts of the world) were developed by telecommunications companies, while Ethernet was designed by computer companies. The two standards are completely incompatible. SDH is older, and much more widely used for long distance communications. Ethernet, particularly 10 Gigabit Ethernet over fibre (1000BASE-LX) offers lower cost and direct interconnection to Local Area Networks.

Optical fibre Internet access is also known as ***Fibre To The Home*** (***FTTH***) or ***Fibre To The Premises*** (***FTTP***). It is currently available in Japan, USA, Canada, some of Europe, Pakistan, New Zealand, and Australia. In other countries it may be available soon.

ADSL

ADSL stands for ***Asymmetric Digital Subscriber Line***, and is a popular implementation of ***DSL***, ***Digital Subscriber Line***. Digital in this case is misleading, as the line is not actually digital, but instead it is rated for much faster signaling than a conventional telephone line. This means it is usually quite close to the exchange, made from high quality copper, and any line filters have been removed. At the exchange is another compatible DSL device, which connects to the Internet using pure digital links. The maximum range of DSL is only a few kilometres, from the customer to the exchange.

DSL is a new technology that is designed to take advantage of recent developments in digital signal processing to offer reliable communications over relatively noisy lines. It takes data packets and breaks them into a number of frequency blocks or bands, including error correction data, and sends them down

the line. Interference normally only affects one or two of these bands at a time, and therefore the DSL modem at the other end can reconstruct the signal if noise damages it, using the error correction data. This makes it much more robust over potentially noisy telephone lines, and gives it a greater range than leased lines.

Asymmetric DSL is so called because the download and upload speeds are different, unlike ***SDSL*** (***Symmetric DSL***) where they are the same. It is often the case that Internet users download more than they upload. For example, receiving email and browsing web pages are both asymmetric activities. On the other hand, running a server almost always requires more upload bandwidth, so ADSL lines are usually not suitable for servers. Telephone companies sell SDSL lines for similar prices as leased lines, but ADSL lines are usually much cheaper.

The reason that ADSL lines are cheap has very little to do with the technology. The main factors forcing the price down are competition from other Internet access technologies, such as cable modems, wireless and satellite. ADSL is often regarded as a consumer product, although small businesses are increasingly replacing leased lines with ADSL for Internet access, while SDSL and leased lines are still considered premium products.

ADSL connections vary widely in speed. The original standard, ADSL1, specified a maximum download speed of just over 8,000 kilobits per second (8 Mbps), while the latest standard, ADSL2+, allows up to 24 Mbps download. These speeds are usually not achieved in practice for three reasons:

- The maximum speed of ADSL depends on the quality of the line, and particularly on the distance between the user and the exchange. The maximum speed at a distance of 5 km is 1 Mbps, and at 6 km it is usually impossible to connect.
- Internet service providers usually place a limit on the speed of each line depending on the amount that the user pays. They will charge more for a 1 Mbps connection than a 128 kbps connection.
- ADSL Internet services are usually oversubscribed by a certain amount, for example 20:1. This is called the ***contention ratio***. It is a common practices based on the idea that most users do not use their connection all the time. In this case, the provider oversells their capacity by 20 or more times.

The issue of contention is almost a universal component of Internet access, and causes significant difficulties with effective bandwidth management since part of the bandwidth between you and the Internet is entirely outside of your control. This can of course be mitigated by signing a ***Service Level Agreement*** (***SLA***) with your ISP, but this comes at a high price, when it is available at all.

ADSL lines are wildly popular in most places where they are available, which includes most capital cities. In Europe, almost all broadband connections are ADSL, far more than cable modem or wireless.

Cable modems

A ***cable modem*** is similar to an ADSL modem, but slightly less complex. Cable modems connect to the cable television network, which is made from coaxial cable rather than pairs of telephone wires. Coaxial cable has much higher quality and lower interference than telephone wires, but is much more expensive.

Cable networks also differ in architecture compared to telephone networks. Cable networks were designed for one-way transmission (broadcasting) from a central point (known as the head end) to a large number of houses. Therefore, they are arranged in a tree structure, branching out from the head end. They are shared, rather than dedicated, so cable modems have to be careful not to talk over one another.

Cable modems transmit and receive on frequency bands, like ADSL, but usually only a single wide frequency band in each direction. This band corresponding to a TV channel on the cable TV network. Coaxial cables have low noise, so the signaling protocol is much simpler than ADSL. Cable modems usually conform to a standard known as ***DOCSIS*** (***Data Over Cable Service Interface Specification***), or the European equivalent ***EuroDOCSIS***.

Cable modems can achieve very high speeds. The fastest speed offered by the latest version, DOCSIS 3.0, is 160 Mbps download and 120 Mbps upload. However, speeds are usually limited by the cable company, and charged similarly to ADSL. Like ADSL, cable modems are usually oversubscribed by a significant margin, and excessive use by some users can have a significant negative impact on the bandwidth available to others. To make matters worse, users in the same neighborhood often share the same physical network segment. This means that many users share the same collision domain, impeding performance and making it possible for users to spy on each others' traffic.

This has recently been improved in some places through the use of hybrid fibre-coax networks. In these networks, the majority of the network backbone is fibre, and local neighborhoods connect via coax to the fibre backbone in numbers of 50-200. This significantly reduces the problems with oversubscription.

Wi-Fi

Wireless Fidelity, or ***Wi-Fi***, is a network technology that uses radio waves to link multiple computers. It grew out of proprietary technologies developed by Lucent, and was standardised by the IEEE under the inspiring and memorable name of 802.11.

The most common Wi-Fi standard is ***802.11b***, which specifies a maximum data rate of 11 Mbps on each of 13 channels around 2.4 GHz. Due to protocol overhead, the maximum available throughput on 802.11b is approximately 5.5 Mbps. The 2.4 GHz frequency band is reserved for Industrial, Scientific, and Medical applications by the ITU, and is known as the ***ISM band***. The ITU permits its use at low power without any license, and therefore Wi-Fi equipment does not require a license to own or operate in most countries. However, the power restrictions limit its range to 300 metres under normal conditions, and around 20 kilometres under ideal conditions with specially adapted equipment. Also, there is no protection from interference in this band, so common household appliances (such as cordless phones and microwave ovens) can generate significant amounts of interference.

Wi-Fi equipment has become very popular in Europe and America now that it is extremely cheap, and almost every modern laptop has a Wi-Fi interface built in. Most companies run their own Wi-Fi networks to allow laptop users to access the corporate network and the Internet away from their desks without trailing cables. Many businesses, for example coffee shops, sell or give away Internet access to their customers via Wi-Fi. Many people install a Wi-Fi network at home to avoid the need to purchase and install cables around the house.

Newer 802.11 standards offer higher speeds. The most popular of these new standards is 802.11g, which provides a radio rate of up to 54 Mbps (22 Mbps throughput) at 2.4 GHz, and is backwards compatible with 802.11b. There is also 802.11a, which operates around 5 GHz. However, 802.11a suffers from range problems and limitations on outdoor use, and is not nearly as ubiquitous and well supported as 802.11b/g.

Some Internet providers offer access using Wi-Fi, which is a very cheap option since the hardware costs about a hundred times less than that required for fixed wireless or microwave networks, and no licenses are required. However, the hardware can be less reliable, is much more susceptible to interference, and can be much less secure than other types of connections.

For a more in-depth look at how Wi-Fi can be adapted to provide Internet connectivity across very long distances, see Wireless Networking in the Developing World, *http://wndw.net/.*

Satellite

There are several standards for satellite Internet access, including ***Digital Video Broadcast*** (***DVB-S***), the ***Broadband Global Access Network*** (***BGAN***) and ***Very Small Aperture Terminal*** (***VSAT***).

Digital Video Broadcasting, or DVB, is a standard for digital communications via satellite. Originally designed for broadcasting television, it has been extended to allow two-way Internet communications. Several satellite providers offer Internet access via DVB, at relatively low cost. Access requires a satellite dish and a DVB-Data tuner card or external modem. Installation requires very precise positioning of the local satellite dish, and therefore must be done by a professional with the necessary tools.

DVB Internet access is available anywhere in the world, even on ships at sea. It is often used in rural areas of Europe and America, where terrestrial broadband is not yet available. It offers high bandwidth, with download speeds up to 16 Mbps, and upload up to 1 Mbps.

BGAN is designed for portable use, and the terminals are very small. It offers speeds up to 492 kbps in both directions. The terminal is not too expensive, but bandwidth prices are usually very high.

VSAT requires a fixed installation, and the dishes are larger than those used for DVB and satellite TV. It offers speeds up to 2 Mbps in each direction.

All satellites used for Internet access are geostationary, and due to their distance from the Earth, Internet access suffers from high round trip times, usually between 500 ms and one second. Depending on the location of the satellite that you use, you may find that your data is routed via the USA, Europe or Asia. The satellite may be very close to the horizon from your location, which results in lower signal strength and interference from objects on the ground, and signal quality may be affected by weather such as rain.

We will review different protocols and variables that affect satellite performance in greater detail in chapter six, **Performance Tuning**.

Fixed Microwave

Fixed microwave networks use radio frequencies between 2 and 31 GHz to connect base stations up to 20 kilometres apart. They are usually point-to-point links, which extend an existing network or connect two networks together. They use high power to achieve such high range and bandwidth, and require licenses to operate. Microwave networks are much cheaper to install than wires, especially because they do not require land ownership or access rights for the

territory they cross. They are also more physically secure, since copper and fibre optic cables can easily be damaged or stolen from the ground.

In many countries, one or more telecommunications companies compete with the state telephone operator by offering wireless "leased line" equivalents using fixed microwave networks. They offer almost the same speed and reliability as wired leased lines, at much lower cost, and are much more easily available in remote areas.

Mobile telephone operators often use fixed microwave networks to connect their rural base stations, and land telephone operators often use them to connect remote towns, villages, and isolated businesses in areas where no fixed telephone lines are available. The antennas often look like wide, thick white discs, flat on the front side and slightly curved or angled on the back side, and can be seen on almost all mobile telephone masts.

Fixed microwave equipment is usually proprietary and incompatible with equipment made by different manufacturers. It is also usually quite expensive, costing several thousand dollars for the equipment at each end.

WiMax

The WiMax standard, formally known as IEEE 802.16, is an attempt to standardise fixed microwave networks and achieve compatibility between equipment made by different manufacturers. The standard is very new, and little equipment that conforms to it is currently available. It is expected to bring down the cost of fixed microwave links, and wireless broadband access, by increasing competition.

It is often claimed that WiMAX will provide access to the Internet at up to 70 Mbps, range up to 70 miles and to a user travelling up to 70 mph, but these are not all true simultaneously. We expect that WiMAX will offer Internet access to remote and rural areas at significantly longer ranges and higher speeds than current options.

WiMAX controls access to the network by allocating time slots to individual users. This can allow more efficient use of radio spectrum than Wi-Fi, and guarantee minimum bandwidth availability, which Wi-Fi cannot do. The standard is specified in the frequency range from 2 to 66 GHz, but it seems likely that most deployments will use frequencies between 2.3 and 5 GHz.

Comparison Table

The table on the following page lists the general features of each type of connection for easy comparison. Data rates given are the rate stated by the manu-

facturer and reflects the theoretical maximum before accounting for protocol overhead.

	Round Trip Time (ms)	Cost (USD)	Quality	Range	Max Up	Max Down	Availability
Modem	100 to 160	cents / minute	High	Global	33.6 kbps	56 kbps	All but the most rural areas
ISDN	20 to 40	cents / minute	High	Global	128 kbps	128 kbps	All but the most rural areas
Leased Line	5 to 20	1000 / month	High	2 km	10 Mbps	10 Mbps	All but the most rural areas
Fibre Optic	Less than 1	50 to 1000 / month	Very High	20 km	1 Gbps	1 Gbps	Some cities
ADSL	10 to 20	20 to 200 / month	High	5 km	1 Mbps	24 Mbps	Most cities
Cable	5 to 80	20 to 200 / month	Medium to High	60+ km	120 Mbps	160 Mbps	Most cities
Wi-Fi	1 to 40	20 to 200 / month	Medium	up to 20 km	54 Mbps	54 Mbps	Some cities
Satellite	At least 500	50 to 450 / month	Medium	Global	1 Mbps	16 Mbps	Global
Fixed Microwave & WiMax	1 to 30	150 to 3000 / month	High	20 km or more	70 Mbps	70 Mbps	Some cities

Virtual connectivity

So far, we have talked about different types of physical network connections. But those physical connections can be broken into even more logical networks, providing multiple isolated virtual networks on a single physical network. The benefits of doing this include being able to restrict which hosts can communicate with each other, separate legacy networks which require control of the broadcast domain from each other, and protect machines from attack over the public Internet.

Virtual LAN (VLAN)

A ***Virtual LAN*** (***VLAN***) is a logical network that can coexist with, and be isolated from, other VLANs on the same physical medium. VLANs are normally implemented by network switching hardware, and so it makes no difference to a computer whether it is connected to a LAN or a VLAN. VLANs can be used to partition a single network switch into a number of isolated domains, effectively simulating a number of independent unconnected switches. This may reduce cost, space, and administration overheads in wiring cabinets. Switches can also be reconfigured remotely, allowing you to logically regroup computers or network ports within an organisation, without the need to physically rewire them.

In terms of bandwidth management, VLANs allow you to group computers by function, and make access policy and bandwidth shaping rules by group. Each VLAN becomes an isolated broadcast domain for the computers it contains. For example, you may create separate VLANs for public terminals, research machines, and administrative computers. Each may be allocated a different amount of bandwidth, or may have different rules in place to permit (or deny) access to particular sites and services. With VLANs in place, these computers do not need to be physically connected to the same switch. Also, computers connected to the same switch can be differentiated by the gateway according to which VLAN they are placed in.

VLAN membership can be decided in three different ways. Support for these may depend on the network switch.

- **Port-based**. A switch port is manually configured to be a member of a VLAN, so any device plugged into that physical port is automatically part of it. Some VLAN switches support ***trunking***, which sends and receives packets tagged with VLAN numbers, and also allows VLANs to extend between switches.
- **Protocol-based**. Layer three protocol information within the frame is used to determine VLAN membership. The major disadvantage of this method is that it violates the independence of the layers. For example, the switch may fail to enforce the VLAN when upgrading from IPv4 to IPv6.
- **MAC-based**. VLAN membership is based on the MAC address of the workstation. The switch has a table listing the MAC address of each machine along with the VLAN to which it belongs. This can allow VLAN policies to work, even if the same machine is plugged into different switch ports (e.g., a laptop user moving around in a building).

The format for tagging packets with the VLAN number, which allows switches and firewalls to share VLAN information, has been standardised by the IEEE as 802.1q.

Virtual Private Network (VPN)

An organisation's network normally offers some private services to its members, such as file servers and web servers. Those services are normally not accessible over the Internet for security reasons. When members of the organisation are operating outside of this network, they will not be able to access these resources. ***Virtual Private Networks*** (***VPNs***) provide a mechanism for securely connecting to such resources, and even linking entire networks together, over an untrusted public network such as the Internet. They are often used to connect remote users to an organisation's network when travelling or working from home.

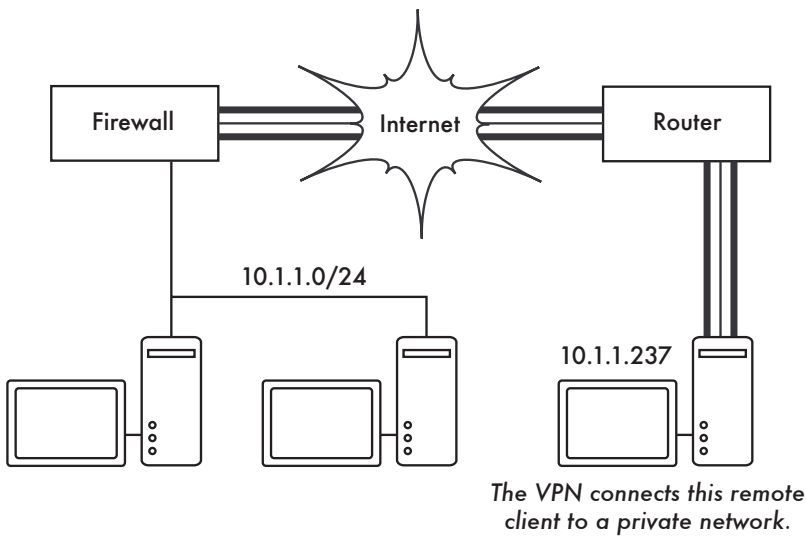


Figure 3.13: VPNs protect sensitive data when crossing untrusted networks. Eavesdroppers cannot penetrate the encrypted VPN encapsulation, represented by the heavy lines.

VPNs may create permanent, on-demand, or temporary connections between two or more networks. They can provide bridging between networks (at the Data Link Layer) or a routed connection (at the Network Layer). Data on the internal network is encrypted and transported within a ***tunnel*** to the remote side, where it is then decrypted and passed on. The encrypted tunnel protects communications from potential eavesdroppers. The establishment of a VPN is normally protected by strong authentication, such as private keys, certificates, or hardware tokens. This provides a strong guarantee to the organisation that its private resources are not being accessed by an unauthorised user.

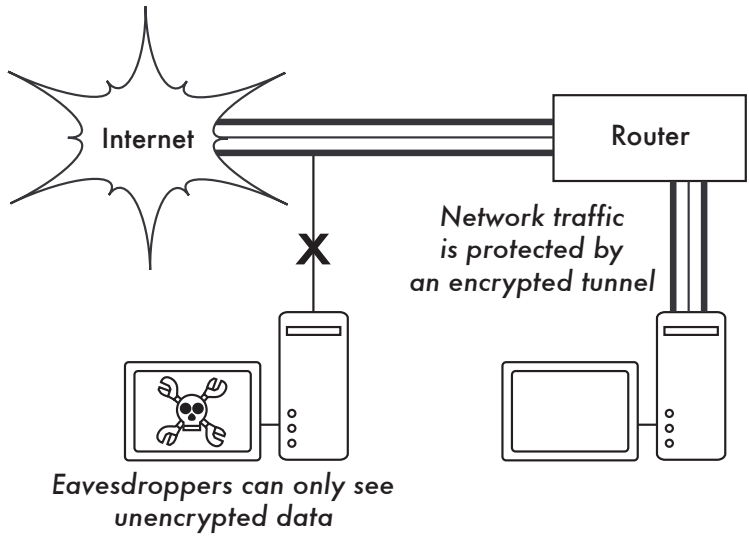


Figure 3.14: Encrypted tunnels can protect data that is not otherwise encrypted by the application itself.

While VPNs create a very flexible and secure connection between remote networks, the tunnel adds overhead to every packet sent over the public network. This makes using a VPN less efficient than accessing a service directly. Attempting to run VPNs over an already crowded Internet connection can make things even worse. One particular problem is that firewalls and bandwidth management software can only see encrypted VPN packets, and not the traffic within, which can make securing, policing, and controlling VPNs much more difficult. Unfortunately, most VPNs are nearly useless over high latency VSAT connections without extensive tweaking.

While VPNs will not improve bandwidth utilisation, they are useful when you need to provide secure access to private resources across an untrusted network, and your bandwidth is not too heavily used.

Summary

Every computer on the Internet has at least one unique IP address. This address is used by other computers to contact it, and send information to it. IP addresses are grouped into subnets of different sizes. Subnets are normally physically connected networks such as Local Area Networks (LANs), and they are connected together by routers and firewalls. There are many different technologies for connecting your network to the Internet, each with its own advantages and disadvantages. Bandwidth management is the process of controlling the information that flows between your network and the Internet.

For more information about the basics of networking, you can find some useful guides online, including Rusty Russell's Linux Networking Concepts: *http://www.netfilter.org/documentation/HOWTO/networking-concepts-HOWTO.html*

By now you should have a clear idea of how information flows between your network and the Internet. Understanding how your network and Internet connection are being used is the first step towards optimising them, improving performance and reducing your operating costs.

What is network monitoring?

Network monitoring is the use of logging and analysis tools to accurately determine traffic flows, utilisation, and other performance indicators on a network. Good monitoring tools give you both hard numbers and graphical aggregate representations of the state of the network. This helps you to visualise precisely what is happening, so you know where adjustments may be needed. These tools can help you answer critical questions, such as:

- What are the most popular services used on the network?
- Who are the heaviest network users?
- At what time of the day is the network most utilised?
- What sites do your users frequent?
- Is the amount of inbound or outbound traffic close to our available network capacity?
- Are there indications of an unusual network situation that is consuming bandwidth or causing other problems?
- Is our Internet Service Provider (ISP) providing the level of service that we are paying for? This should be answered in terms of available bandwidth, packet loss, latency, and overall availability.

And perhaps the most important question of all:

- Do the observed traffic patterns fit our expectations?

If you cannot answer these questions, then you are likely wasting bandwidth. With good network monitoring tools in place, it is easier to troubleshoot problems, identify your biggest bandwidth users, and plan for future capacity needs.

Let's look at how a typical system administrator can make good use of network monitoring tools.

An effective network monitoring example

For the purposes of example, let's assume that we are in charge of a network that has been running for three months. It consists of 50 computers and three servers - email, web, and proxy servers. While initially things are going well, users begin to complain of slow network speeds and an increase in spam emails. As time goes on, computer performance slows to a crawl (even when not using the network), causing considerable frustration in your users.

With frequent complaints and very low computer usage, the Board is questioning the need for so much network hardware. The Board also wants evidence that the bandwidth they are paying for is actually being used. As the network administrator, you are on the receiving end of these complaints. How can you diagnose the sudden drop in network and computer performance and also justify the network hardware and bandwidth costs?

Monitoring the LAN (local traffic)

To get an idea of exactly what is causing the slow down, you should begin by looking at traffic on the local LAN. There are several advantages to monitoring local traffic:

- Troubleshooting is greatly simplified.
- Viruses can be detected and eliminated.
- Malicious users can be detected and dealt with.
- Network hardware and resources can be justified with real statistics.

Assume that all of the switches support the ***Simple Network Management Protocol*** (***SNMP***). SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. By assigning an IP address to each switch, you are able to monitor all the interfaces on that switch, and can therefore monitor the entire network from a single point. This is much easier than enabling SNMP on all computers in a network.

By using a free tool such as MRTG (page **83**), you can monitor each port on the switch and present data graphically, as an aggregate average over time. The graphs are accessible from the web, so you are able to view the graphs from any machine at anytime.

With MRTG monitoring in place, it becomes obvious that the internal LAN is swamped with far more traffic than the Internet connection can support, even when the lab is unoccupied. This is a pretty clear indication that some of the computers are infested with a network virus. After installing good anti-virus and anti-spyware software on all of the machines, the internal LAN traffic settles

down to expected levels. The machines run much more quickly, spam emails are reduced, and the users' morale quickly improves.

Monitoring the WAN (external traffic)

In addition to watching the traffic on the internal LAN, you need to demonstrate that the bandwidth the organisation is paying for is actually what they are getting from their ISP. You can achieve this by monitoring ***external traffic***.

External traffic is generally classified as anything sent over a ***Wide Area Network*** (***WAN***). Anything received from (or sent to) a network other than your internal LAN also qualifies as external traffic. The advantages of monitoring external traffic include:

- Internet bandwidth costs are justified by showing actual usage, and whether that usage agrees with your ISP's bandwidth charges.
- Future capacity needs are estimated by watching usage trends and predicting likely growth patterns.
- Intruders from the Internet are detected and filtered before they can cause problems.

Monitoring this traffic is easily done with the use of MRTG on an SNMP enabled device, such as a router. If your router does not support SNMP, then you can add a switch between your router and your ISP connection, and monitor the port traffic just as you would with an internal LAN.

Detecting Network Outages

With monitoring tools in place, you now have an accurate measurement of how much bandwidth the organisation is using. This measurement should agree with your ISP's bandwidth charges. It can also indicate the actual throughput of your connection if you are using close to your available capacity at peak times. A "flat top" graph is a fairly clear indication that you are operating at full capacity. Figure 3.15 shows flat tops in peak outbound traffic in the middle of every day except Sunday.

It is clear that your current Internet connection is overutilised at peak times, causing network lag. After presenting this information to the Board, you can make a plan for further optimising your existing connection (by upgrading your proxy server and using other techniques in this book) and estimate how soon you will need to upgrade your connection to keep up with the demand. This is also an excellent time to review your operational policy with the Board, and discuss ways to bring actual usage in line with that policy.

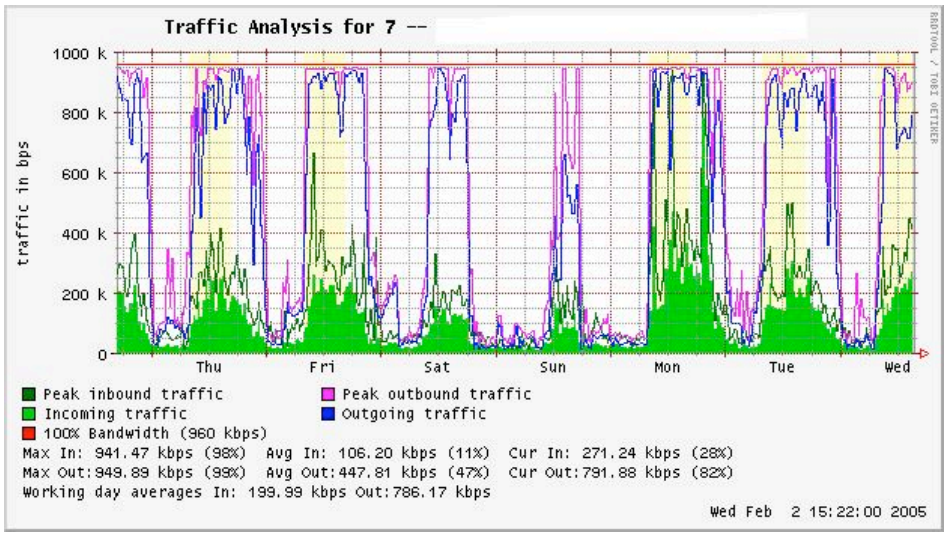


Figure 3.15: A graph with a "flat top" is one indication of overutilisation.

Later in the week, you receive an emergency phone call in the evening. Apparently, no one in the lab one can browse the web or send email. You rush to the lab and hastily reboot the proxy server, with no results. Browsing and email are still broken. You then reboot the router, but there is still no success. You continue eliminating the possible fault areas one by one until you realise that the network switch is off - a loose power cable is to blame. After applying power, the network comes to life again.

How can you troubleshoot such an outage without such time consuming trial and error? Is it possible to be notified of outages as they occur, rather than waiting for a user to complain? One way to do this is to use a program such as ***Nagios*** that continually polls network devices and notifies you of outages. Nagios will report on the availability of various machines and services, and will alert you to machines that have gone down. In addition to displaying the network status graphically on a web page, it will send notifications via SMS or email, alerting you immediately when problems arise.

With good monitoring tools in place, you are able to justify the cost of equipment and bandwidth by demonstrating how it is being used by the organisation. You are notified automatically when problems arise, and you have historical statistics of how the network devices are performing. You can check the current performance against this history to find unusual behaviour, and head off problems before they become critical. When problems do come up, it is simple to determine the source and nature of the problem. Your job is easier, the Board is satisfied, and your users are much happier.

Monitoring your network

Managing a network without monitoring is similar to driving a vehicle without a speedometer or a fuel gauge. How do you know how fast you are going? Is the car consuming fuel as efficiently as promised by the dealers? If you do an engine overhaul several months later, is the car any faster or more efficient than it was before?

Similarly, how can you pay for an electricity or water bill without seeing your monthly usage from a meter? You must have an account of your network bandwidth utilisation in order to justify the cost of services and hardware purchases, and to account for usage trends.

There are several benefits to implementing a good monitoring system for your network:

1. **Network budget and resources are justified.** Good monitoring tools can demonstrate without a doubt that the network infrastructure (bandwidth, hardware, and software) is suitable and able to handle the requirements of network users.
2. **Network intruders are detected and filtered.** By watching your network traffic, you can detect attackers and prevent access to critical internal servers and services.
3. **Network viruses are easily detected.** You can be alerted to the presence of network viruses, and take appropriate action before they consume Internet bandwidth and destabilise your network
4. **Troubleshooting of network problems is greatly simplified.** Rather than attempting "trial and error" to debug network problems, you can be instantly notified of specific problems. Some kinds of problems can even be repaired automatically.
5. **Network performance can be highly optimised.** Without effective monitoring, it is impossible to fine tune your devices and protocols to achieve the best possible performance.
6. **Capacity planning is much easier.** With solid historical performance records, you do not have to "guess" how much bandwidth you will need as your network grows.
7. **Proper network usage can be enforced.** When bandwidth is a scarce resource, the only way to be fair to all users is to ensure that the network is being used for its intended purpose.

Fortunately, network monitoring does not need to be an expensive undertaking. There are many freely available open source tools that will show you exactly

what is happening on your network in considerable detail. This chapter will help you identify many invaluable tools and how best to use them.

The dedicated monitoring server

While monitoring services can be added to an existing network server, it is often desirable to dedicate one machine (or more, if necessary) to network monitoring. Some applications (such as ***ntop***) require considerable resources to run, particularly on a busy network. But most logging and monitoring programs have modest RAM and storage requirements, typically with little CPU power required. Since open source operating systems (such as Linux or BSD) make very efficient use of hardware resources, this makes it possible to build a very capable monitoring server from recycled PC parts. There is usually no need to purchase a brand new server to relegate to monitoring duties.

The exception to this rule is in very large installations. If your network includes more than a few hundred nodes, or if you consume more than 50 Mbps of Internet bandwidth, you will likely need to split up monitoring duties between a few dedicated machines. This depends largely on exactly what you want to monitor. If you are attempting to account for all services accessed per MAC address, this will consume considerably more resources than simply measuring network flows on a switch port. But for the majority of installations, a single dedicated monitoring machine is usually enough.

While consolidating monitoring services to a single machine will streamline administration and upgrades, it can also ensure better ongoing monitoring. For example, if you install monitoring services on a web server, and that web server develops problems, then your network may not be monitored until the problem is resolved.

To a network administrator, the data collected about network performance is nearly as important as the network itself. Your monitoring should be robust and protected from service outages as well as possible. Without network statistics, you are effectively blind to problems with the network.

Where does the server fit in my network?

If you are only interested in collecting network flow statistics from a router, you can do this from just about anywhere on the LAN. This provides simple feedback about utilisation, but cannot give you comprehensive details about usage patterns. Figure 3.16 shows a typical MRTG graph generated from the Internet router. While the inbound and outbound utilisation are clear, there is no detail about which computers, users, or protocols are using bandwidth.

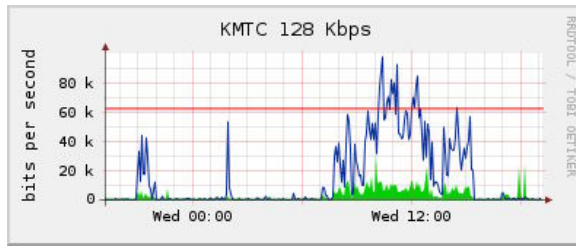


Figure 3.16: Polling the edge router can show you the overall network utilisation, but you cannot break the data down further into machines, services, and users.

For more detail, the dedicated monitoring server must have access to everything that needs to be watched. Typically, this means it must have access to the entire network. To monitor a WAN connection, such as the Internet link to your ISP, the monitoring server must be able to see the traffic passing through the edge router. To monitor a LAN, the monitoring server is typically connected to a ***monitor port*** on the switch. If multiple switches are used in an installation, the monitoring server may need a connection to all of them. That connection can either be a physical cable, or if your network switches support it, a VLAN specifically configured for monitoring traffic.

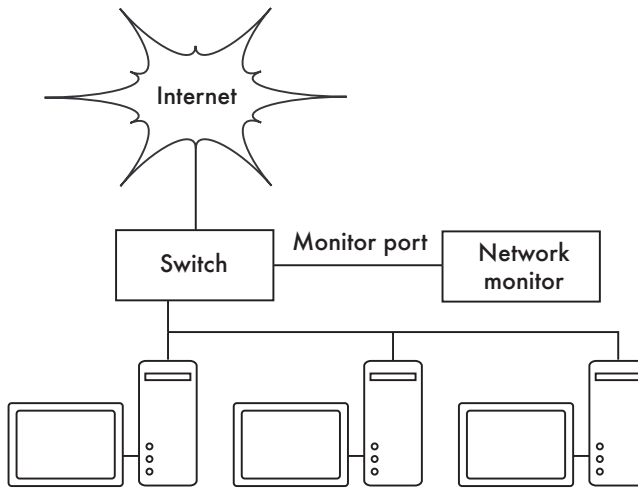


Figure 3.17: Use the monitor port on your switch to observe traffic crossing all of the network ports.

If monitor port functionality is not available on your switch, the monitoring server may be installed between your internal LAN and the Internet. While this will work, it introduces a single point of failure for the network, as the network will fail if the monitoring server develops a problem. It is also a potential performance bottleneck, if the server cannot keep up with the demands of the network.

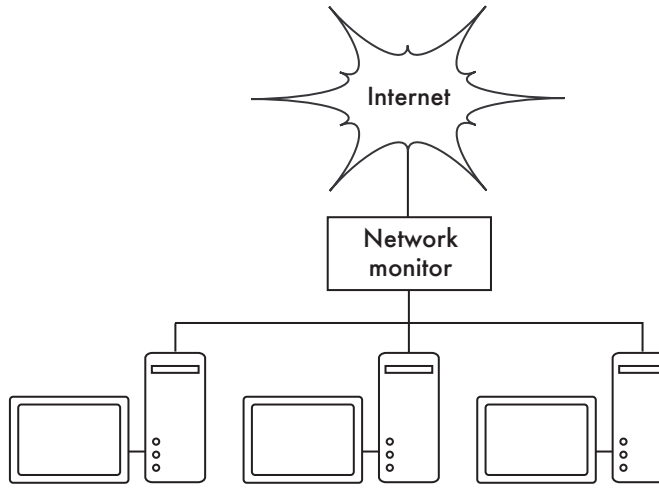


Figure 3.18: By inserting a network monitor between the LAN and your Internet connection, you can observe all network traffic.

A better solution is to use a simple network hub (not a switch) which connects the monitoring machine to the internal LAN, external router, and the monitoring machine. While this does still introduce an additional point of failure to the network (since the entire network will be unreachable if the hub dies), hubs are generally considered to be much more reliable than routers. They are also very easily replaced should they fail.

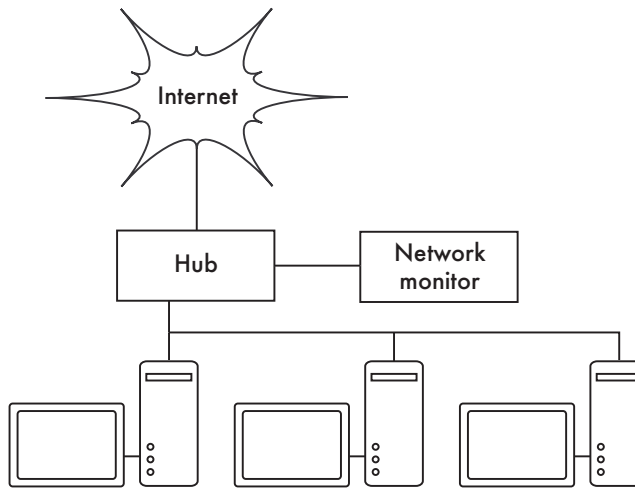


Figure 3.19: If your switch does not provide monitor port functionality, you can insert a network hub between your Internet router and the LAN, and connect the monitoring server to the hub.

Once your monitoring server is in place, you are ready to start collecting data.

What to monitor

It is possible to plot just about any network event and watch its value on a graph over time. Since every network is slightly different, you will have to decide what information is important in order to gauge the performance of your network.

Here are some important indicators that many network administrators will typically track.

Switch statistics

- Bandwidth usage per switch port
- Bandwidth usage broken down by protocol
- Bandwidth usage broken down by MAC address
- Broadcasts as a percentage of total packets
- Packet loss and error rate
- Wireless signal, noise, and data rate

Internet statistics

- Internet bandwidth use by host and protocol
- Proxy server cache hits
- Top 100 sites accessed
- DNS requests
- Number of inbound emails / spam emails / email bounces
- Outbound email queue size
- Availability of critical services (web servers, email servers, etc.).
- Ping times and packet loss rates to your ISP
- Status of backups

System health statistics

- Memory usage
- Swap file usage
- Process count / zombie processes
- System load

- Uninterruptible Power Supply (UPS) voltage and load
- Temperature, fan speed, and system voltages
- Disk SMART status
- RAID array status

You should use this list as a suggestion of where to begin. As your network matures, you will likely find new key indicators of network performance, and you should of course track those as well. There are many freely available tools that will show you as much detail as you like about what is happening on your network. You should consider monitoring the availability of any resource where unavailability would adversely affect your network users.

For example, your users may dial into modems on your site to gain remote access to your network. If all the modems are used, or if any are faulty, then users will be denied access and will probably complain. You can predict and avoid such problems by monitoring the number of available modems, and provisioning extra capacity before you run out.

Don't forget to monitor the monitoring machine itself, for example its CPU usage and disk space, in order to receive advance warning if it becomes overloaded or faulty. A monitoring machine that is low on resources can affect your ability to monitor the network effectively.

How to select tools to monitor the network

In order to properly and effectively monitor a network, you must first select a monitoring tool. When evaluating tools, you should consider which of the following properties are most important to you. Good monitoring tools should be:

- **Appropriate.** Just as you do not measure the contents of a gas tank with a thermometer, you should use the proper monitoring tool for the job. Some tools (such as ***tcpdump***) can give you enormous amounts of detail, but very little idea of trends over time. Others (such as ***MRTG***) give you great historical detail of network flows, but no details about a particular protocol or user. Your monitoring solution will include several tools, each performing a particular monitoring task.
- **Affordable.** There are many monitoring programs that are totally free for download on the Internet and require no payments or license fees. If your project has a limited budget, these can be ideal. Even if you have considerable budget to spend on monitoring tools, be sure to compare the functionality of proprietary solutions with the free or open source equivalent.
- **Lightweight.** This means the tool must not be processor or memory exhaustive, especially if you are unable to dedicate a machine to monitoring. It must

also be able to perform for long periods without taking up much hard disk space. Some programs may require huge amounts of space to function, while others may need little or no storage at all. Be sure that the tool you choose fits the hardware on which it is used.

- **Flexible**. Your tools should adapt to your networking environment, not the other way around. Since every network is different, your monitoring tools should be able to be configured to accurately display the data you need to track.
- **Graphical.** The best programs plot graphs or have colorful displays to allow for easy and quick understanding of network activities. For example, red will indicate an alert or error while green will indicate no problems. This turns data into information as opposed to a meaningless string of numbers and percentages. Most modern programs will do both.
- **Supported.** Ensure that the program is well supported by its authors (or the community at large) with updates and patches. Such programs are more secure and more reliable in the long run, since developers are constantly reviewing functionality and addressing problems.
- **Data retentive.** The program should be able to log and retain data over long periods (e.g. two or three years). This will help you discover trends in a network environment by comparing values in year A to values in year B.
- **User friendly and feature rich.** The best programs are easy to use and present data in an easy to understand way. The program should also have features such as web accessibility and even web administration, so that people with modest computer skills can help monitor the network.

While not all tools will meet every one of these criteria, it is important to keep these points in mind when deciding which tools you will use to monitor your network.

Types of monitoring tools

We will now look at several different classes of monitoring tools. ***Spot check*** tools are designed for troubleshooting and normally run interactively for short periods of time. A program such as **`ping`** may be considered an active spot check tool, since it generates traffic by polling a particular machine. Passive spot check tools include ***protocol analysers***, which inspect every packet on the network and provide complete detail about any network conversation (including source and destination addresses, protocol information, and even application data). ***Trending*** tools perform unattended monitoring over long periods, and typically plot the results on a graph. ***Realtime monitoring*** tools perform similar monitoring, but notify administrators immediately if they detect a problem. ***Log analysis*** tools summarise the logs of various other programs, and may notify an administrator when problems arise. ***Intrusion detection***

tools watch for undesirable or unexpected network traffic, and take appropriate action (typically denying access and/or notifying a network administrator). Finally, ***benchmarking*** tools estimate the maximum performance of a service or network connection.

The class of tool to use depends on the question you need to answer.

- **How do I troubleshoot an immediate problem?** If your logs or graphs don't indicate the source of the problem (page **83**), use a spot check tool (page **74**).
- **How do I estimate the performance of my connection?** Use a benchmarking tool (page **89**).
- **How do I recognise a virus or denial of service attack?** Look at bandwidth consumption trends (page **83**), watch your email and web server logs (page **80**), or use an intrusion detection tool or spot check tool (page **74**). Also see the discussion on pages **170** and **174**.
- **How can I be automatically notified when outages occur?** Use a realtime monitoring tool (page **87**).
- **How do I ensure that bandwidth is used for suitable services/purposes?** Use an intrusion detection tool (page **87**) or protocol analyser (page **78**.) It is also useful to install a trending tool to monitor protocols used broken down by bandwidth (or even IP or MAC address) over time (page **83**.)
- **When should I upgrade my bandwidth?** Use a trending tool (page **83**) in conjunction with a protocol analyser (page **78**). When you are sure that your bandwidth is being used appropriately, a trending tool will show you how much you have consumed in the past, and you can predict how much you will need in the future.

Walking around the lab

Do not discount the effectiveness of simply walking into a lab environment. If you wish to have an understanding of the social habits of users in an educational facility, visit the lab during the busiest times of the day. While walking around different areas of the lab, I have been surprised at what I have found. In areas where there is little physical policing, students often browse personal sites, play games, listen to music, and perform other recreational activities. In the areas where more rigid physical policing has been performed through notices and by lab consultants, the atmosphere tends to much more work oriented.

Being available to (and noticed by) your users can often improve network performance significantly. This is nearly always easier to implement than any technical solution.

Spot check tools

There may be times when you suspect someone is misappropriating network resources. Perhaps someone has found a way around the proxy, and is downloading music or movies. ***Spot check tools*** can help you locate the source of the problem. Spot check tools give you a quick view of what is happening on your connection in real time. Some tools (such as ***ntop***) can run continually, while others (such as ***wireshark***) are usually only run when needed.

ping

Probably the most commonly used spot check tool is the ubiquitous ***ping*** command. Just about every operating system (including Windows, Mac OS X, and of course Linux and BSD) includes a version of this utility. It uses ICMP packets to attempt to contact a specified host, and tells you how long it takes to get a response.

Knowing what to ping is just as important as knowing how to ping. If you find that you cannot connect to a particular service in your web browser (say, *http://yahoo.com/*), you could try to ping it:

```
$ ping yahoo.com
PING yahoo.com (66.94.234.13): 56 data bytes
64 bytes from 66.94.234.13: icmp_seq=0 ttl=57 time=29.375 ms
64 bytes from 66.94.234.13: icmp_seq=1 ttl=56 time=35.467 ms
64 bytes from 66.94.234.13: icmp_seq=2 ttl=56 time=34.158 ms
^C
--- yahoo.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 29.375/33.000/35.467/2.618 ms
```

Hit control-C when you are finished collecting data. If packets take a long time to come back, there may be network congestion. If return ping packets have an unusually low ***Time To Live*** (***TTL***), you may have routing problems between your machine and the remote end. But what if the ping doesn't return any data at all? Some Internet hosts do not respond to pings for security reasons, so it may help to try another address on the Internet which is known to respond to pings. However, if you are pinging a host name instead of an IP address, you may be running into DNS problems.

To check this, try pinging an IP address on the Internet, instead of a domain name. If that works, then you have a problem with your DNS server. If you don't get a response from the IP address, try to ping your default router:

```
$ ping 216.231.38.1
PING 216.231.38.1 (216.231.38.1): 56 data bytes
64 bytes from 216.231.38.1: icmp_seq=0 ttl=126 time=12.991 ms
64 bytes from 216.231.38.1: icmp_seq=1 ttl=126 time=14.869 ms
```

```
64 bytes from 216.231.38.1: icmp_seq=2 ttl=126 time=13.897 ms
^C
--- 216.231.38.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 12.991/13.919/14.869/0.767 ms
```

If you can't ping your default router, then chances are you won't be able to get to the Internet either. If you can't even ping other IP addresses on your local LAN, then it's time to check your connection. If you're using Ethernet, is it plugged in? If you're using wireless, are you connected to the proper wireless network, and is it in range?

Network debugging with ping is a bit of an art, but it is useful to learn. Since you will likely find ping on just about any machine you will work on, it's a good idea to learn how to use it well.

traceroute and mtr

As with ping, ***traceroute*** is found on most operating systems (it's called ***tracert*** in some versions of Microsoft Windows). By running traceroute, you can find the location of problems between your computer and any point on the Internet:

```
$ traceroute -n google.com
traceroute to google.com (72.14.207.99), 64 hops max, 40 byte packets
 1  10.15.6.1  4.322 ms  1.763 ms  1.731 ms
 2  216.231.38.1  36.187 ms  14.648 ms  13.561 ms
 3  69.17.83.233  14.197 ms  13.256 ms  13.267 ms
 4  69.17.83.150  32.478 ms  29.545 ms  27.494 ms
 5  198.32.176.31  40.788 ms  28.160 ms  28.115 ms
 6  66.249.94.14  28.601 ms  29.913 ms  28.811 ms
 7  172.16.236.8  2328.809 ms  2528.944 ms  2428.719 ms
 8  * * *
```

The **-n** switch tells traceroute not to bother resolving names in DNS, and makes the trace run more quickly. You can see that at hop seven, the round trip time shoots up to more than two seconds, while packets seem to be discarded at hop eight. This might indicate a problem at that point in the network. If this part of the network is in your control, it might be worth starting your trouble-shooting effort there. Note that traceroute tools can only indicate the path forward from the local machine to the remote end, but cannot display the actual return path.

My TraceRoute (***mtr***) is a handy program that combines ping and traceroute into a single tool. It is available at *http://www.bitwizard.nl/mtr/*. By running mtr, you can see an ongoing average of round trip times and packet loss to a single host, instead of the momentary snapshot that ping and traceroute provide.

```
                                My traceroute  [v0.69]
tesla.rob.swn (0.0.0.0)                (tos=0x0 psize=64 bitpatSun Jan  8 20:01:26 2006
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                     Packets               Pings
 Host                              Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. gremlin.rob.swn                 0.0%     4    1.9   2.0   1.7   2.6   0.4
 2. er1.sea1.speakeasy.net          0.0%     4   15.5  14.0  12.7  15.5   1.3
 3. 220.ge-0-1-0.cr2.sea1.speakeasy.  0.0%     4   11.0  11.7  10.7  14.0   1.6
 4. fe-0-3-0.cr2.sfo1.speakeasy.net   0.0%     4   36.0  34.7  28.7  38.1   4.1
 5. bas1-m.pao.yahoo.com            0.0%     4   27.9  29.6  27.9  33.0   2.4
 6. so-1-1-0.pat1.dce.yahoo.com     0.0%     4   89.7  91.0  89.7  93.0   1.4
 7. ae1.p400.msr1.dcn.yahoo.com     0.0%     4   91.2  93.1  90.8  99.2   4.1
 8. ge5-2.bas1-m.dcn.yahoo.com      0.0%     4   89.3  91.0  89.3  93.4   1.9
 9. w2.rc.vip.dcn.yahoo.com         0.0%     3   91.2  93.1  90.8  99.2   4.1
```

The data will be continuously updated and averaged over time. As with ping, you should hit control-C when you are finished looking at the data. Note that you must have root privileges to run mtr.

ntop

Ntop is probably one of the most useful traffic monitoring programs around. It is essentially a network protocol analyser with an intuitive built-in web interface that delivers a wealth of information.

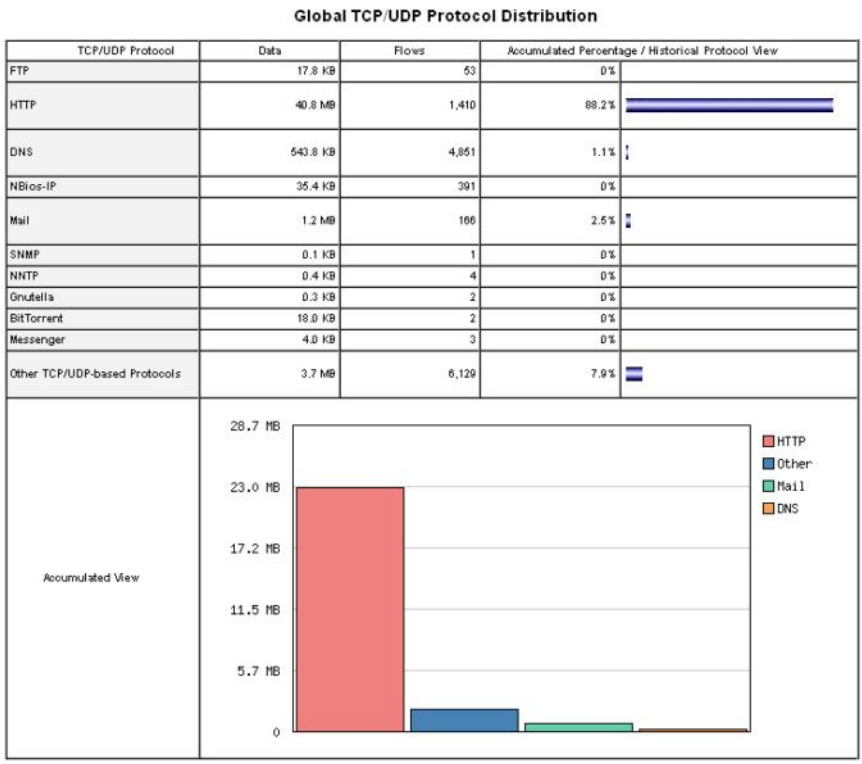


Figure 3.20: ntop displays a wealth of information about how your network is utilised by various clients and servers.

Some of its more useful features include:

- Traffic display can be sorted by various criteria (source, destination, protocol, MAC address, etc.).
- Traffic statistics grouped by protocol and port number
- An IP traffic matrix which shows connections between machines
- Network flows for routers or switches that support the NetFlow protocol
- Host operating system identification
- P2P traffic identification
- Numerous graphical charts
- Perl, PHP, and Python API

Ntop is available from *http://www.ntop.org/* and is available for most operating systems. It is often included in many of the popular Linux distributions, including RedHat, Debian, and Ubuntu. While it can be left running to collect historical data, ntop can be fairly CPU intensive, depending on the amount of traffic observed. If you are going to run it for long periods you should monitor the CPU utilisation of the monitoring machine.

The main disadvantage of ntop is that it does not provide instantaneous information, only long-term totals and averages. This can make it difficult to use to diagnose a problem that starts suddenly.

iptraf

IPTraf is a lightweight but powerful LAN monitor. It has an ncurses interface and runs in a command shell. IPTraf takes a moment to measure observed traffic, and then displays various network statistics including TCP and UDP connections, ICMP and OSPF information, traffic flows, IP checksum errors, and more. It is a simple to use program that uses minimal system resources.

While it does not keep historical data, it is very useful for displaying an instantaneous usage report. IPTraf is available from *http://iptraf.seul.org/.*

tcpdump

Tcpdump (*http://www.tcpdump.org/*) is a command-line tool for monitoring network traffic. It does not have all the bells and whistles of wireshark (page **78**) but it does use fewer resources. Tcpdump can capture and display all network protocol information down to the link layer. It can show all of the packet headers and data received, or just the packets that match particular criteria. Packets captured with tcpdump can be loaded into wireshark for visual analysis and further diagnostics. This is very useful if you wish to monitor an interface on a

remote system and bring the file back to your local machine for analysis. The tcpdump tool is available as a standard tool in Unix derivatives (Linux, BSD, and Mac OS X). There is also a Windows port called ***WinDump*** available at *http://www.winpcap.org/windump/.*

Wireshark (Ethereal)

Wireshark (formerly known as Ethereal) is a free network protocol analyser for Unix and Windows, and is billed as "The World's Most Popular Network Protocol Analyser." It allows you to examine data from a live network or from a capture file on disk, and interactively browse and sort the captured data. Both summary and detailed information is available for each packet, including the full header and data portions. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.

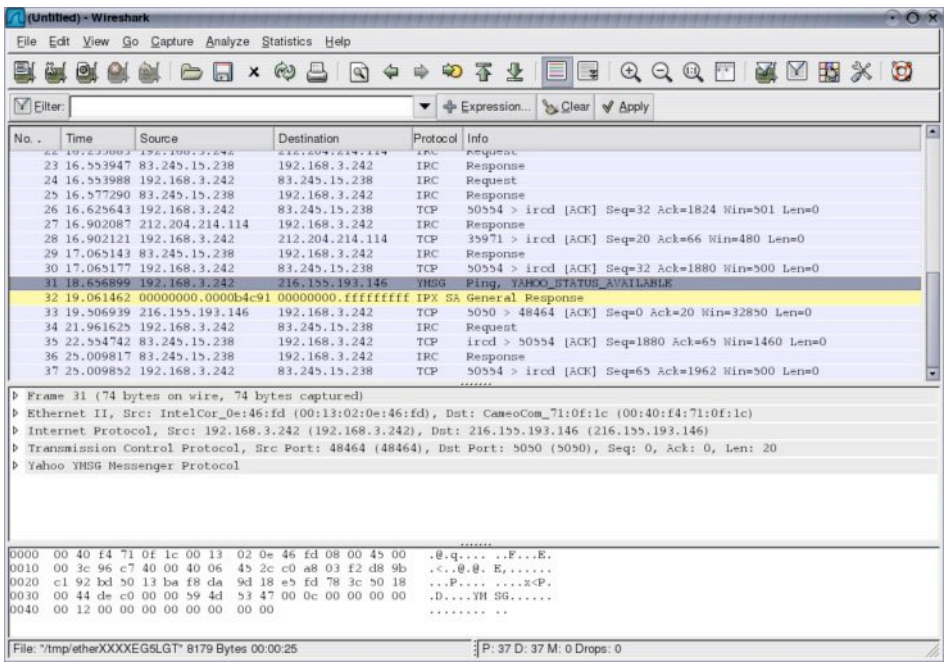


Figure 3.21: Wireshark (formerly Ethereal) is a powerful network protocol analyser that can show you as much detail as you like about any packet that crosses the wire.

It can be daunting to use for first time users or those that are not familiar with the OSI layers. It is typically used to isolate and analyse specific traffic to or from an IP address, but it can be also used as a general purpose fault finding tool. For example, a machine infected with a network worm or virus can be identified by looking for the machine that is send out the same sort of TCPIP packets to large groups of IP addresses.

Wireshark is available from *http://www.wireshark.org/.* For examples of how to use wireshark to debug specific network problems, see chapter five, **Troubleshooting**.

ngrep

Ngrep provides most of GNU grep's pattern matching features, but applies them to network traffic. It currently recognises IPv4 and IPv6, TCP, UDP, ICMP, IGMP, PPP, SLIP, FDDI, Token Ring, and much more. As it makes extensive use of regular expression matches, it is a tool suited to advanced users or those that have a good knowledge of regular expressions.

But you don't necessarily need to be a regex expert to be able to make basic use of ngrep. For example, to view all packets that contain the string GET (presumably HTTP requests), try this:

```
# ngrep -q GET
```

Pattern matches can be constrained further to match particular protocols, ports, or other criteria using BPF filters. This is the filter language used by common packet sniffing tools, such as tcpdump and snoop. To view GET or POST strings sent to destination port 80, use this command line:

```
# ngrep -q 'GET|POST' port 80
```

By using ngrep creatively, you can detect anything from virus activity to spam email. You can download ngrep at *http://ngrep.sourceforge.net/.*

EtherApe

EtherApe (*http://etherape.sourceforge.net/*) displays a graphical representation of network traffic. Hosts and links change size depending on the amount of traffic sent and received. The colors change to represent the protocol most used. As with wireshark and tcpdump, data can be captured "off the wire" from a live network connection or read from a tcpdump capture file.

EtherApe doesn't show quite as much detail as ntop, but its resource requirements are much lighter.

Argus

Argus (*http://qosient.com/argus/*) stands for ***Audit Record Generation and Utilization System***. Argus is also the name of the mythological Greek god who had hundreds of eyes.

From the Argus website:

> *Argus generates flow statistics such as connectivity, capacity, demand, loss, delay, and jitter on a per transaction basis. Argus can be used to analyse and report on the contents of packet capture files or it can run as a continuous monitor, examining data from a live interface; generating an audit log of all the network activity seen in the packet stream. Argus can be deployed to monitor individual end-systems, or an entire enterprises network activity. As a continuous monitor, Argus provides both push and pull data handling models, to allow flexible strategies for collecting network audit data. Argus data clients support a range of operations, such as sorting, aggregation, archival and reporting.*

Argus consists of two parts: a master collector that reads packets from a network device, and a client that connects to the master and displays the usage statistics. Argus runs on BSD, Linux, and most other UNIX systems.

NeTraMet

NeTraMet (*http://www.auckland.ac.nz/net/NeTraMet/*) is another popular flow analysis tool. Like Argus, NeTraMet consists of two parts: a collector that gathers statistics via SNMP, and a manager that specifies which flows should be watched. Flows are specified using a simple programming language that define the addresses used on either end, and can include Ethernet, IP, protocol information, or other identifiers. NeTraMet runs on DOS and most UNIX systems, including Linux and BSD.

Log analysers

Information that has been derived from logging and trending tools can be very useful in determining the expected future usage patterns of your network. While some spot check tools can be used over long periods to capture information, many of them will generate a lot of data. Managing the storage and archiving of this data can be tricky.

Often, simple data derived from log files can be just as useful in determining where potential problem areas lie. Log analysers and trending tools tend to require much less disk space and CPU power than spot check tools. Here are several popular and freely available logging and trending tools.

Firewall logs

Your firewall can generate a potentially huge amount of statistical information. This can be used to compile a list of top bandwidth users (or abusers), and identify users that find holes in your firewall rules to run bandwidth intense applications such as peer-to-peer programs. If you are running a commercial fire-

wall, it usually includes a reporting feature to notify you of abnormal traffic patterns. If you are running an open source firewall under Linux or BSD, there are any number of log file reporters available. Some of these include:

- IPTables log analyser, *http://www.gege.org/iptables/*
- ADMLogger, *http://aaron.marasco.com/linux.html*
- adcfw-log, *http://adcfw-log.sourceforge.net/*
- logwatch, *http://www.logwatch.org/*

To use a log watcher with your firewall, you will need to configure your firewall to log the packets you want to watch. For example, you could configure your firewall to log packets bound for port 445, a port used by the Sasser worm. Analysis of activity on this port may show an attacker's IP address, from which you would want to block traffic.

Proxy (cache) logs

Proxy logs contain extensive details about the browsing habits of your users. By analysing this information, you can find users who make inappropriate use of network resources, and identify sites that may benefit from mirroring (page **144**). These logs can also help pinpoint potential problem areas on your network, such as viruses. While we will discuss Squid cache logs in particular, these techniques will work with many different kinds of cache servers.

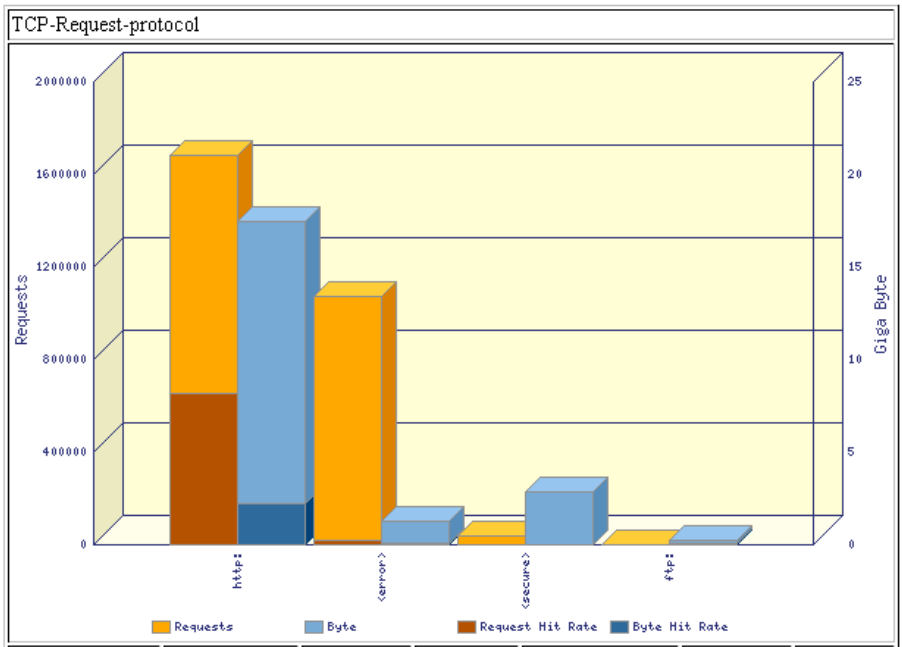
Unless otherwise specified, Squid log files use a standard web server log format. Popular log file analysers such as ***Analog*** (*http://www.analog.cx/*), ***Webalizer*** (*http://www.mrunix.net/webalizer/*), and ***AWStats*** (*http://awstats.sourceforge.net/*) will read Squid log files and produce detailed reports on the data collected. The Squid project site has a page full of links to compatible log file analysers at *http://www.squid-cache.org/Scripts/*.

One popular analyser is ***Calamaris***, *http://cord.de/tools/squid/calamaris/.* Calamaris produces impressive html reports and graphs of proxy utilisation. It is written in Perl, and requires gdlib to be installed to display graphs. Calamaris, perl, and gdlib are available on most Linux distributions as standard packages. Calamaris is shown in Figure 3.22.

Some people are frustrated by the use of the Unix epoch time format used in the Squid logs. This perl snippet:

```
tail -f /var/log/squid/access.log | perl -pe 's/^\d+/localtime $&/e;'
```

...will convert the timestamps into a human readable format.



protocol	request	%	hit-%	Byte	%	hit-%
http:	1683656	60.28	38.86	17860M	80.07	12.64
<error>	1074321	38.46	1.47	1333979K	5.84	1.95
<secure>	35144	1.26	0.00	3002505K	13.15	0.00
ftp:	151	0.01	33.77	214765K	0.94	2.16
Sum	2793272	100.00	23.99	22305M	100.00	10.25

Figure 3.22: Calamaris will analyse your Squid logs to give you informative reports about your users' web traffic patterns.

Another popular log analyser is ***Sawmill*** (*http://www.sawmill.net/*). It is commercial software, which among other things means that it comes with technical support. Sawmill can produce more detailed and professional-looking reports than many open source alternatives. It also has some interesting advanced features, such as geographic location reports. Sawmill runs on Unix and Windows. It requires a commercial license based on the type and amount of data to be analysed.

In the same way that there are programs for analysing web and cache server logs, there are programs for analysing mail server logs. Using these, it is possible to better understand the demands that email places on your network. The open source program ***Isoqlog*** (*http://www.enderunix.org/isoqlog/*) understands the log files generated by four commonly-used UNIX mail servers (qmail, postfix, sendmail and exim), and generates statistics from these.

Trending tools

Trending tools are used to see how your network is used over a long period of time. They work by periodically monitoring your network activity, and displaying a summary in a human-readable form (such as a graph). Unlike log analysers, trending tools collect data as well as analyse and reporting on it.

Below are some examples of trending tools. Some of them need to be used in conjunction with each other, as they are not stand-alone programs.

MRTG

The ***Multi Router Traffic Grapher*** (***MRTG***) monitors the traffic load on network links using SNMP. MRTG generates graphs that provide a visual representation of inbound and outbound traffic. These are typically displayed on a web page. MRTG is available from *http://oss.oetiker.ch/mrtg/*.

MRTG can be a little confusing to set up, especially if you are not familiar with SNMP. But once it is installed, MRTG requires virtually no maintenance, unless you change something on the system that is being monitored (such as its IP address).

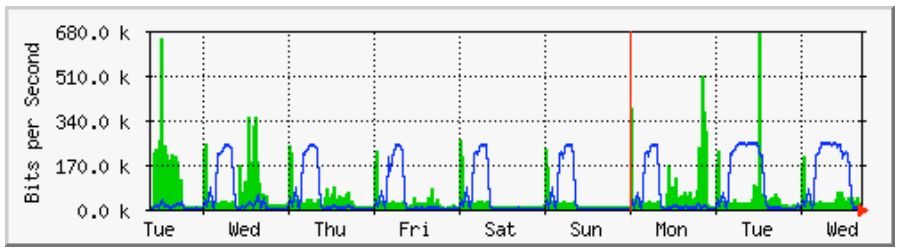


Figure 3.23: MRTG is probably the most widely installed network flow grapher.

RRDtool

RRD is short for ***Round Robin Database***. RRD is a database that stores information in a very compact way that does not expand over time. ***RRDtool*** refers to a suite of tools that allow you to create and modify RRD databases, as well as generate useful graphs to present the data. It is used to keep track of time-series data (such as network bandwidth, machine room temperature, or server load average) and can display that data as an average over time.

Note that RRDtool itself does not contact network devices to retrieve data. It is merely a database manipulation tool. You can use a simple wrapper script (typically in shell or Perl) to do that work for you. RRDtool is also used by many full featured front-ends that present you with a friendly web interface for

configuration and display. RRD graphs give you more control over display options and the number of items available on a graph as compared to MRTG.

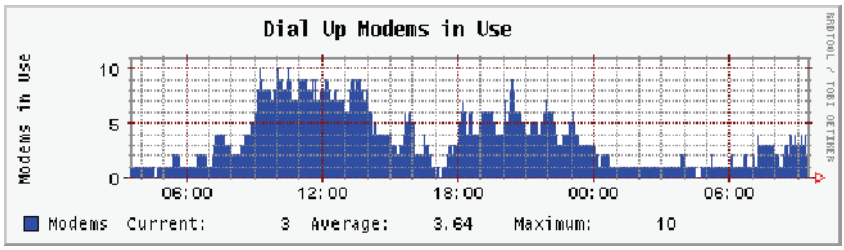


Figure 3.24: RRDtool gives you a lot of flexibility in how your collected network data may be displayed.

RRDtool is included in virtually all modern Linux distributions, and can be downloaded from *http://oss.oetiker.ch/rrdtool/.*

Cacti

Cacti (*http://www.cacti.net/*) is one such front-end for RRDtool. It stores all of the necessary information to create graphs in a MySQL database. The front-end is written in PHP. Cacti does the work of maintaining graphs, data sources, and handles the actual data gathering. There is support for SNMP devices, and custom scripts can easily be written to poll virtually any conceivable network event.

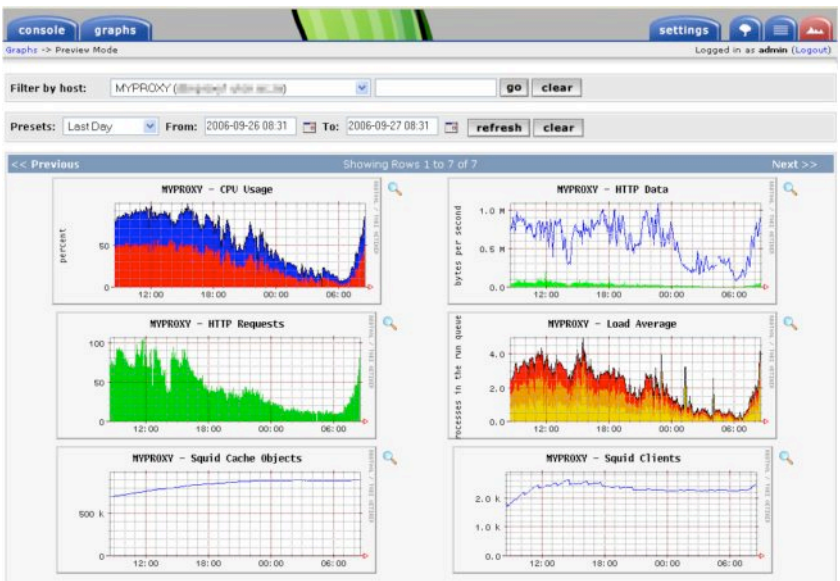


Figure 3.25: Cacti can manage the polling of your network devices, and can build very complex and informative visualisations of network behaviour.

Cacti can be somewhat confusing to configure, but once you work through the documentation and examples, it can yield very impressive graphs. There are hundreds of templates for various systems available on the cacti website, and the code is under rapid development.

NetFlow

NetFlow is a protocol for collecting IP traffic information invented by Cisco. From the Cisco website:

> *Cisco IOS NetFlow efficiently provides a key set of services for IP applications, including network traffic accounting, usage-based network billing, network planning, security, Denial of Service monitoring capabilities, and network monitoring. NetFlow provides valuable information about network users and applications, peak usage times, and traffic routing.*

Cisco routers can generate NetFlow information which is available from the router in the form of UDP packets. NetFlow is also less CPU-intensive on Cisco routers than using SNMP. It also provides more granular information than SNMP, letting you get a more detailed picture of port and protocol usage.

This information is collected by a NetFlow collector that stores and presents the data as an aggregate over time. By analysing flow data, one can build a picture of traffic flow and traffic volume in a network or on a connection. There are several commercial and free NetFlow collectors available. Ntop (page **76**) is one free tool that can act as a NetFlow collector and probe. Another is Flowc, which is described in detail on page **86**.

It can also be desirable to use Netflow as a spot check tool, by just looking at a quick snapshot of data during a network crisis. Think of NetFlow as an alternative to SNMP for Cisco devices. For more information about NetFlow, see *http://en.wikipedia.org/wiki/Netflow* .

SmokePing

SmokePing (*http://oss.oetiker.ch/smokeping/*) is a deluxe latency measurement tool written in Perl. It can measure, store and display latency, latency distribution and packet loss all on a single graph. SmokePing uses RRDtool for data storage, and can draw very informative graphs that present up to the minute information on the state of your network connection.

It is very useful to run SmokePing on a host with good connectivity to your entire network. Over time, trends are revealed that can point to all sorts of network problems. Combined with MRTG (page **83**) or Cacti (page **84**), you can observe the effect that network congestion has on packet loss and latency. SmokePing can optionally send alerts when certain conditions are met, such as

when excessive packet loss is seen on a link for an extended period of time. An example of SmokePing in action is shown in Figure 3.26.

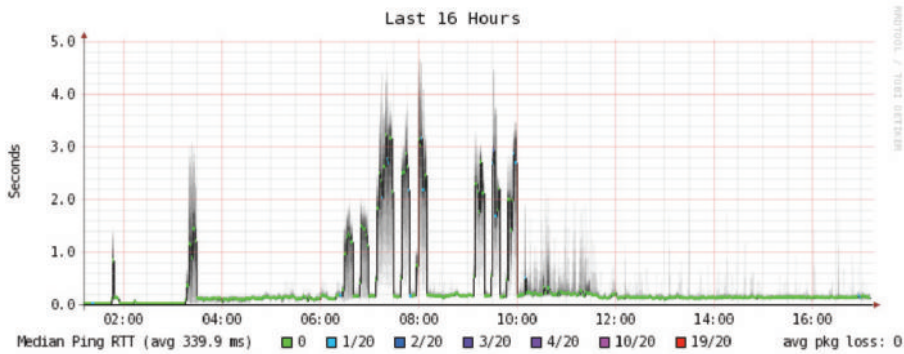


Figure 3.26: SmokePing can simultaneously display packet loss and latency spreads in a single graph.

Flowc

Flowc (*http://netacad.kiev.ua/flowc/*) is an open source NetFlow collector (see NetFlow above). It is lightweight and easy to configure. Flowc uses a MySQL database to store aggregated traffic information. Therefore, it is possible to generate your own reports from the data using SQL, or use the included report generators. The built-in report generators produce reports in HTML, plain text or a graphical format.

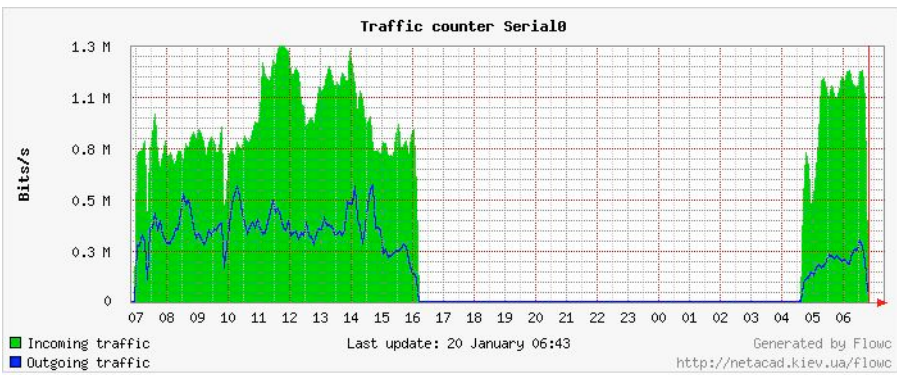


Figure 3.27: A typical flow chart generated by Flowc.

The large gap in data probably indicates a network outage. Trending tools typically will not notify you of outages, but merely log the occurrence. To be notified when network problems occur, use a realtime monitoring tool such as Nagios (page **88**).

Realtime tools

It is desirable to find out when people are trying to break into your network, or when some part of the network has failed. Because no system administrator can be monitoring a network all the time, there are programs that constantly monitor the status of the network and can send alerts when notable events occur. The following are some open source tools that can help perform this task.

Snort

Snort (*http://www.snort.org/*) is a packet sniffer and logger which can be used as a lightweight network intrusion detection system. It features rule-based logging and can perform protocol analysis, content searching, and packet matching. It can be used to detect a variety of attacks and probes, such as stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and many other kinds of anomalous traffic patterns. Snort has a real-time alert capability that can notify administrators about problems as they occur with a variety of methods.

Installing and running Snort is not trivial, and depending on the amount of network traffic, will likely require a dedicated monitoring machine with considerable resources. Fortunately, Snort is very well documented and has a strong user community. By implementing a comprehensive Snort rule set, you can identify unexpected behaviour that would otherwise mysteriously eat up your Internet bandwidth.

See *http://snort.org/docs/* for an extensive list of installation and configuration resources.

Apache: mod_security

ModSecurity (*http://www.modsecurity.org/*) is an open source intrusion detection and prevention engine for web applications. This kind of security tool is also known as a ***web application firewall***. ModSecurity increases web application security by protecting web applications from known and unknown attacks. It can be used on its own, or as a module in the Apache web server (*http://www.apache.org/*).

There are several sources for updated mod_security rules that help protect against the latest security exploits. One excellent resource is GotRoot, which maintains a huge and frequently updated repository of rules:

http://gotroot.com/tiki-index.php?page=mod_security+rules

Web application security is important in defending against attacks on your web server, which could result in the theft of valuable or personal data, or in the

server being used to launch attacks or send spam to other Internet users. As well as being damaging to the Internet as a whole, such intrusions can seriously reduce your available bandwidth.

Nagios

Nagios (*http://nagios.org/*) is a program that monitors hosts and services on your network, notifying you immediately when problems arise. It can send notifications via email, SMS, or by running a script, and will send notifications to the relevant person or group depending on the nature of the problem. Nagios runs on Linux or BSD, and provides a web interface to show up-to-the-minute system status.

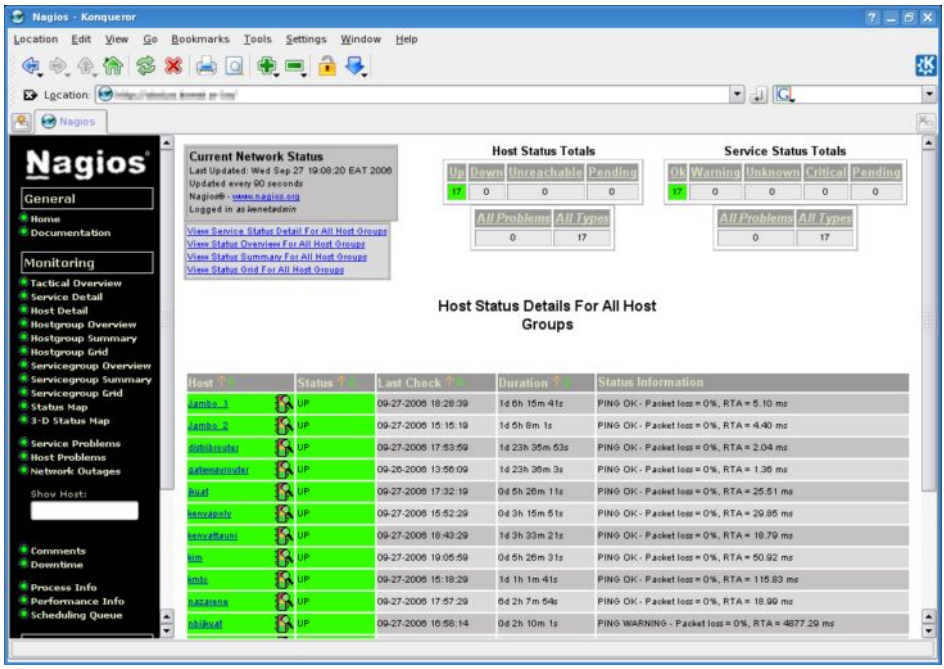


Figure 3.28: Nagios keeps you informed the moment a network fault or service outage occurs.

Nagios is extensible, and can monitor the status of virtually any network event. It performs checks by running small scripts at regular intervals, and checks the results against an expected response. This can yield much more sophisticated checks than a simple network probe. For example, ping (page **74**) may tell you that a machine is up, and nmap may report that a TCP port responds to requests, but Nagios can actually retrieve a web page or make a database request, and verify that the response is not an error.

Nagios can even notify you when bandwidth usage, packet loss, machine room temperature, or other network health indicator crosses a particular threshold. This can give you advance warning of network problems, often allowing you to respond to the problem before users have a chance to complain.

Zabbix

Zabbix (*http://www.zabbix.org/*) is an open source realtime monitoring tool that is something of a hybrid between Cacti and Nagios. It uses a SQL database for data storage, has its own graph rendering package, and performs all of the functions you would expect from a modern realtime monitor (such as SNMP polling and instant notification of error conditions). Zabbix is released under the GNU General Public License.

Benchmarking

How fast can the network go? What is the actual usable capacity of a particular network link? You can get a very good estimate of your throughput capacity by flooding the link with traffic and measuring how long it takes to transfer the data.

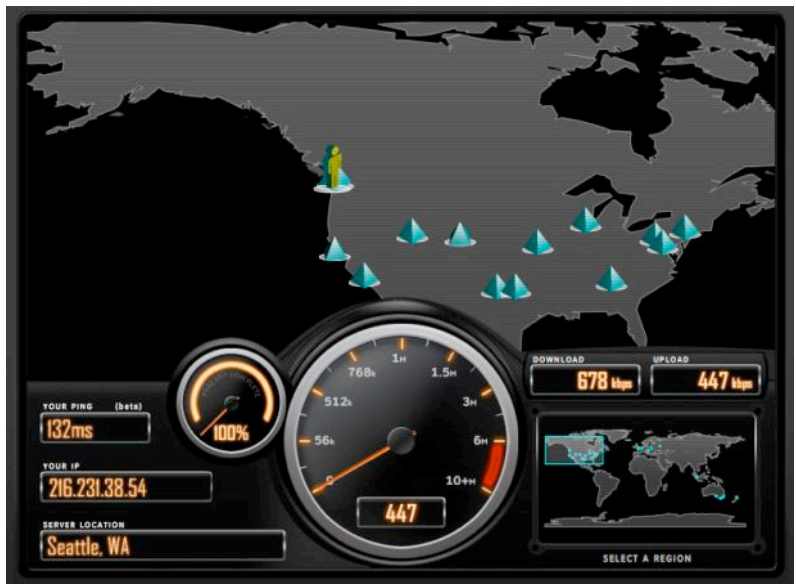


Figure 3.29: Tools such as this one from SpeedTest.net are pretty, but don't always give you an accurate picture of network performance.

While there are web pages available that will perform a "speed test" in your browser (such as *http://www.dslreports.com/stest* or *http://speedtest.net/*), these tests are increasingly inaccurate as you get further from the testing source. Even worse, they do not allow you to test the speed of a given link, but

only the speed of your link to a particular site on the Internet. Here are three tools that will allow you to perform throughput testing on your own networks.

- **ttcp** (*http://ftp.arl.mil/ftp/pub/ttcp/*). Now a standard part of most Unix-like systems, ttcp is a simple network performance testing tool. One instance is run on either side of the link you want to test. The first node runs in receive mode, and the other transmits:

```
node_a$ ttcp -r -s

node_b$ ttcp -t -s node_a
ttcp-t: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp -> node_a
ttcp-t: socket
ttcp-t: connect
ttcp-t: 16777216 bytes in 249.14 real seconds = 65.76 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 124.57, calls/sec = 8.22
ttcp-t: 0.0user 0.2sys 4:09real 0% 0i+0d 0maxrss 0+0pf 7533+0csw
```

  After collecting data in one direction, you should reverse the transmit and receive partners to test the link in the other direction. It can test UDP as well as TCP streams, and can alter various TCP parameters and buffer lengths to give the network a good workout. It can even use a user-supplied data stream instead of sending random data. Remember that the speed readout is in kilobytes, not kilobits. Multiply the result by 8 to find the speed in kilobits per second.

  The only real disadvantage to ttcp is that it hasn't been developed in years. Fortunately, the code has been released in the public domain and is freely available. Like ping and traceroute, ttcp is found as a standard tool on many systems.

- **iperf** (*http://dast.nlanr.net/Projects/Iperf/*). Much like ttcp, iperf is a commandline tool for estimating the throughput of a network connection. It supports many of the same features as ttcp, but uses a "client" and "server" model instead of a "receive" and "transmit" pair. To run iperf, launch a server on one side and a client on the other:

```
node_a$ iperf -s

node_b$ iperf -c node_a
------------------------------------------------------------
Client connecting to node_a, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  5] local 10.15.6.1 port 1212 connected with 10.15.6.23 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-11.3 sec   768 KBytes   558 Kbits/sec
```

  The server side will continue to listen and accept client connections on port 5001 until you hit control-C to kill it. This can make it handy when running multiple test runs from a variety of locations.

The biggest difference between ttcp and iperf is that iperf is under active development, and has many new features (including IPv6 support). This makes it a good choice as a performance tool when building new networks.

- **bing** (*http://www.freenix.fr/freenix/logiciels/bing.html*). Rather than flood a connection with data and see how long the transfer takes to complete, Bing attempts to estimate the available throughput of a point-to-point connection by analysing round trip times for various sized ICMP packets. While it is not always as accurate as a flood test, it can provide a good estimate without transmitting a large number of bytes.

  Since bing works using standard ICMP echo requests, so it can estimate available bandwidth without the need to run a special client on the other end, and can even attempt to estimate the throughput of links outside your network. Since it uses relatively little bandwidth, bing can give you a rough idea of network performance without running up the charges that a flood test would certainly incur.

What is normal?

If you are looking for a definitive answer as to what your traffic patterns **should** look like, you are going to be disappointed. There is no absolute right answer to this question, but given some work you can determine what is normal for your network. While every environment is different, some of the factors that can influence the appearance of your traffic patterns are:

- The capacity of your Internet connection
- The number of users that have access to the network
- The social policy (byte charging, quotas, honor system, etc.).
- The number, types, and level of services offered
- The health of the network (presence of viruses, excessive broadcasts, routing loops, open email relays, denial of service attacks, etc.).
- The competence of your computer users
- The location and configuration of control structures (firewalls, proxy servers, caches, and so on)

This is not a definitive list, but should give you an idea of how a wide range of factors can affect your bandwidth patterns. With this in mind, let's look at the topic of baselines.

Establishing a baseline

Since every environment is different, you need to determine for yourself what your traffic patterns look like under normal situations. This is useful because it

allows you to identify changes over time, either sudden or gradual. These changes may in turn indicate a problem, or a potential future problem, with your network.

For example, suppose that your network grinds to a halt, and you are not sure of the cause. Fortunately, you have decided to keep a graph of broadcasts as a percentage of the overall network traffic. If this graph shows a sudden increase in the amount of broadcast traffic, it may mean that your network has been infected with a virus. Without an idea of what is "normal" for your network (a baseline), you would not be able to see that the number of broadcasts had increased, only that it was relatively high, which may not indicate a problem.

Baseline graphs and figures are also useful when analysing the effects of changes made to the network. It is often very useful to experiment with such changes by trying different possible values. Knowing what the baseline looks like will show you whether your changes have improved matters, or made them worse.

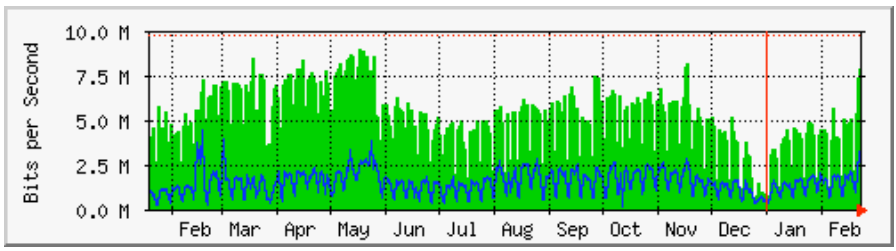


Figure 3.30: By collecting data over a long period of time, you can predict the growth of your network and make changes before problems develop.

In Figure 3.30, we can see the effect the implementation of delay pools has made on Internet utilisation around the period of May. If we did not keep a graph of the line utilisation, we would never know what the effect of the change over the long term was. When watching a total traffic graph after making changes, don't assume that just because the graph does not change radically that your efforts were wasted. You might have removed frivolous usage from your line only to have it replaced by genuine legitimate traffic. You could then combine this baseline with others, say the top 100 sites accessed or the average utilisation by your top twenty users, to determine if habits have simply changed. As we will see later, MRTG, RRDtool, and Cacti are excellent tools you can use to keep a baseline.

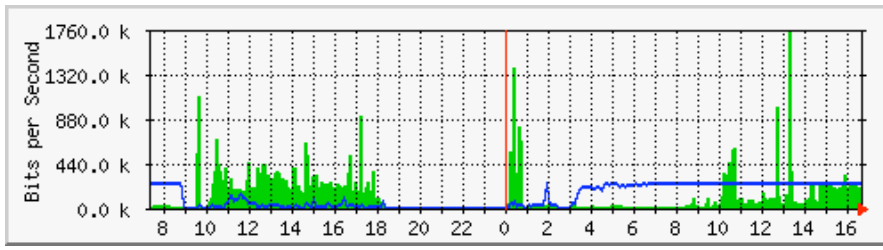


Figure 3.31: The traffic trend at Aidworld logged over a single day.

Figure 3.31 shows traffic on an Aidworld firewall over a period of 24 hours. There is nothing apparently wrong with this graph, but users were complaining about slow Internet access.

Figure 3.32 shows that the upload bandwidth use (dark area) was higher during working hours on the last day than on previous days. A period of heavy upload usage started every morning at 03:00, and was normally finished by 09:00, but on the last day it was still running at 16:30. Further investigation revealed a problem with the backup software, which ran at 03:00 every day.

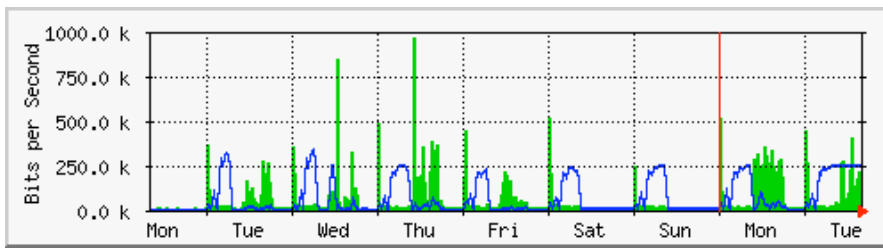


Figure 3.32: The same network logged over an entire week reveals a problem with backups, which caused unexpected congestion for network users.

Figure 3.33 shows measurements of latency on the same connection as measured by a program called SmokePing. The position of the dots shows the average latency, while the gray smoke indicates the distribution of latency (jitter). The color of the dots indicates the number of lost packets. This graph over a period of four hours does not help to identify whether there are any problems on the network.

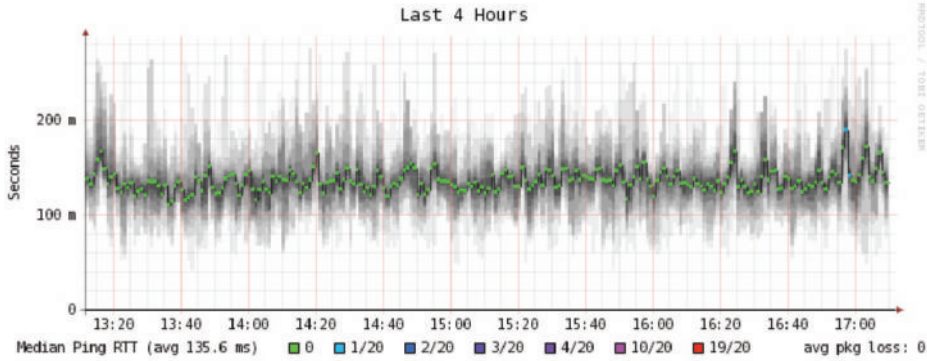


Figure 3.33: Four hours of jitter and packet loss.

The next graph (figure 3.34) shows the same data over a period of 16 hours. This indicates that the values in the graph above are close to the normal level (baseline), but that there were significant increases in latency at several times during the early morning, up to 30 times the baseline value. This indicates that additional monitoring should be performed during these early morning periods to establish the cause of the high latency, which is probably heavy traffic of some kind.

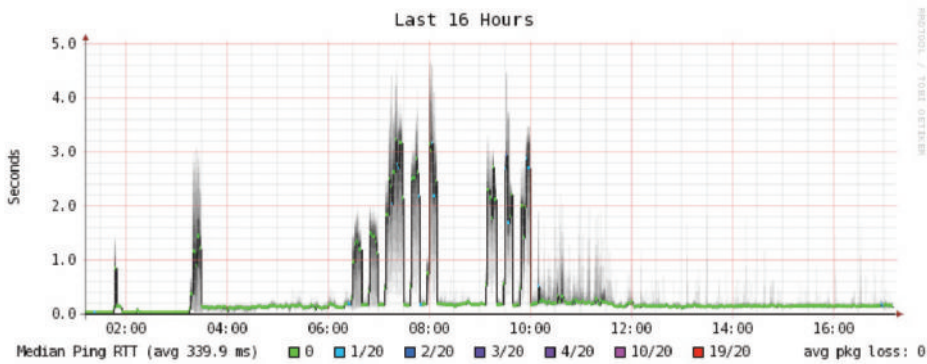


Figure 3.34: A higher spread of jitter is revealed in the 16 hour log.

Figure 3.35 shows that Tuesday was significantly worse than Sunday or Monday for latency, especially during the early morning period. This might indicate that something has changed on the network.

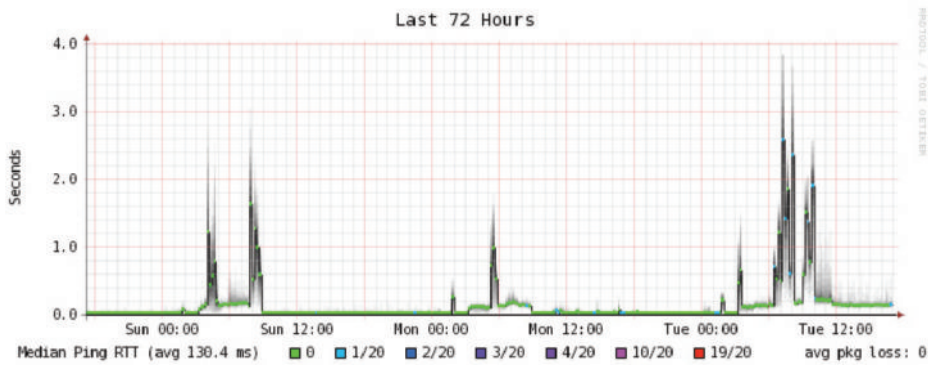


Figure 3.35: Zooming out to the week long view reveals a definite repetition of increased latency and packet loss in the early morning hours.

How do I interpret the traffic graph?

In a basic network flow graph (such as that generated by the network monitor MRTG), the green area indicates ***inbound traffic***, while the blue line indicates ***outbound traffic***. Inbound traffic is traffic that originates from another network (typically the Internet) and is addressed to a computer inside your network. Outbound traffic is traffic that originates from your network, and is addressed to a computer somewhere on the Internet. Depending on what sort of network environment you have, the graph will help you understand how your network is actually being used. For example, monitoring of servers usually reveals larger amounts of outbound traffic as the servers respond to requests (such as sending mail or serving web pages), while monitoring client machines might reveal higher amounts of inbound traffic to the machines as they receive data from the servers.

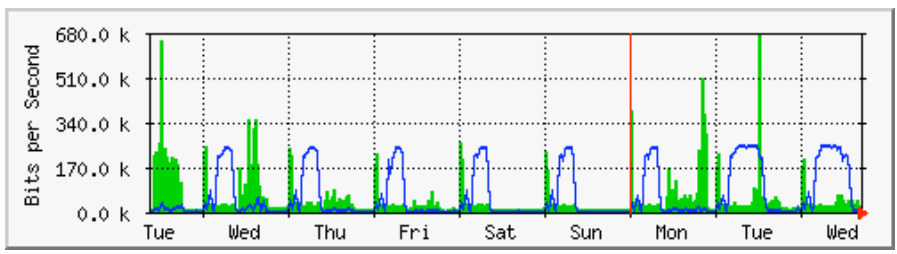


Figure 3.36: The classic network flow graph. The dark area represents inbound traffic, while the line represents outbound traffic. The repeating arcs of outbound traffic show when the nightly backups have run.

Traffic patterns will vary with what you are monitoring. A router will normally show more incoming traffic than outgoing traffic as users download data from the Internet. An excess of outbound bandwidth that is not transmitted by your network servers may indicate a peer-to-peer client, unauthorised server, or

even a virus on one or more of your clients. There are no set metrics that indicate what outgoing traffic to incoming traffic should look like. It is up to you to establish a baseline to understand what normal network traffic patterns look like on your network.

Detecting network overload

Figure 3.37 shows traffic on an overloaded Internet connection.

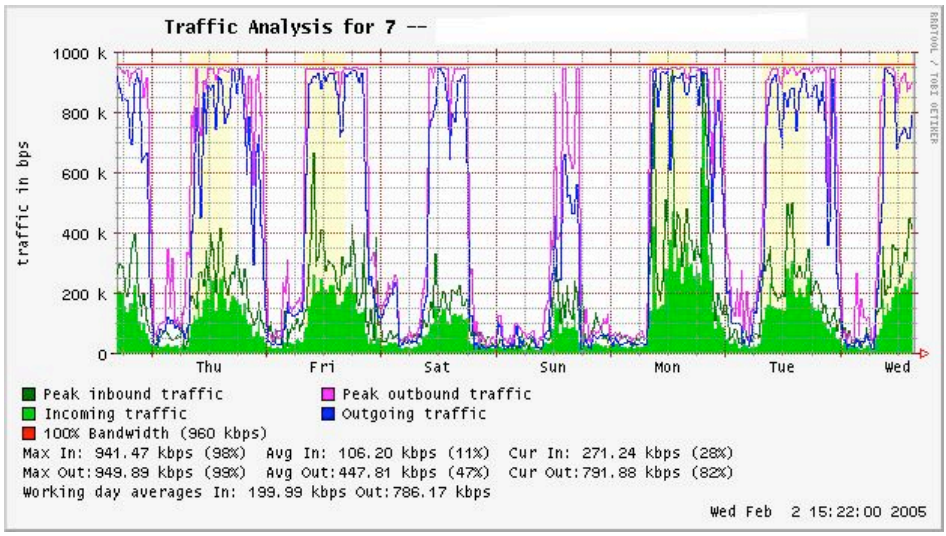


Figure 3.37: Flat-topped graphs indicate that a line is using the maximum available bandwidth, and is overutilised during these times.

The most apparent sign of overloading is the flat tops on outbound traffic during the middle of every day. Flat tops may indicate overloading, even if they are well below the maximum theoretical capacity of the link. In this case it may indicate that you are not getting as much bandwidth from your service provider as you expect.

Measuring 95th percentile

The 95th percentile is a widely used mathematical calculation to evaluate regular and sustained utilisation of a network pipe. Its value shows the highest consumption of traffic for a given period. Calculating the 95th percentile means that 95% of the time the usage is below a certain amount, and 5% of the time usage is above that amount. The 95th percentile is a good value to use to show the bandwidth that is actually used at least 95% of the time.

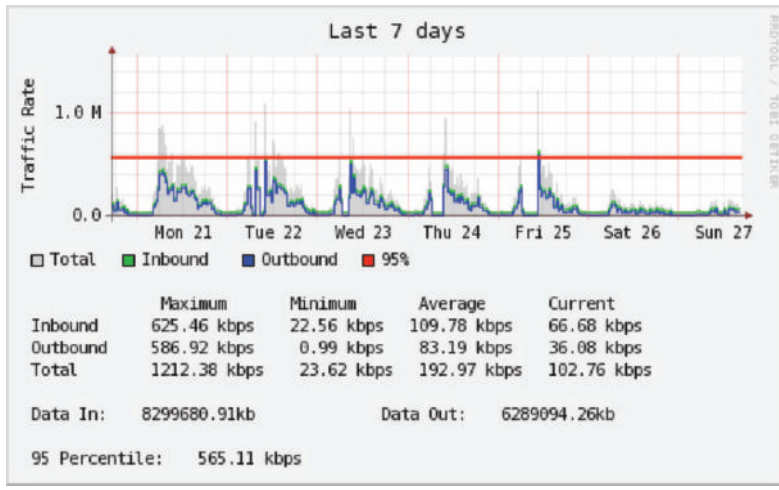


Figure 3.38: The horizontal line shows the 95th percentile amount.

MRTG and Cacti will calculate the 95th Percentile for you. This is a sample graph of a 960 Kbps connection. The 95th percentile came to 945 Kbps after discarding the highest 5% of traffic.

Monitoring RAM and CPU usage

By definition, servers provide critical services that should always be available. Servers receive and respond to client machine requests, providing access to services that are the whole point of having a network in the first place. Therefore, servers must have sufficient hardware capabilities to accommodate the work load. This means they must have adequate RAM, storage, and processing power to accommodate the number of client requests. Otherwise, the server will take longer to respond, or in the worst case, may be incapable of responding at all. Since hardware resources are finite, it is important to keep track of how system resources are being used. If a core server (such as a proxy server or email server) is overwhelmed by requests, access times become slow. This is often perceived by users as a network problem.

There are several programs that can be used to monitor resources on a server. The simplest method on a Windows machine is to access the Task Manager using the **Ctrl Alt + Del** keys, and then click on the Performance tab. On a Linux or BSD box, you can type **`top`** in a terminal window. To keep historical logs of such performance, MRTG or RRDtool (page **83**) can also be used.

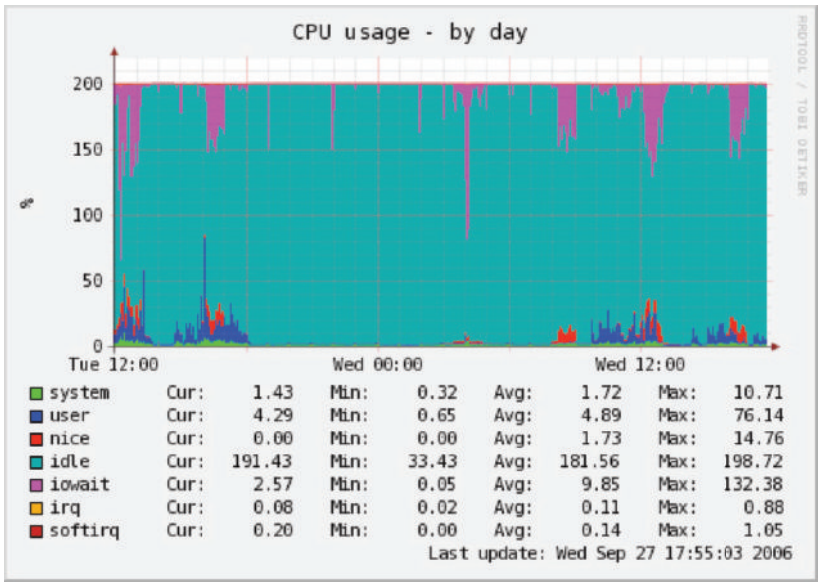


Figure 3.39: RRDtool can show arbitrary data, such as memory and CPU usage, expressed as an average over time.

Mail servers require adequate space, as some people may prefer to leave their email messages on the server for long periods of time. The messages can accumulate and fill the hard disk, especially if quotas are not in use. If the disk or partition used for mail storage fills up, the mail server cannot receive mail. If that disk is also used by the system, all kinds of system problems may occur as the operating system runs out of swap space and temporary storage.

File servers need to be monitored, even if they have large disks. Users will find a way to fill any size disk more quickly than you might think. Disk usage can be enforced through the use of quotas, or by simply monitoring usage and telling people when they are using too much. Nagios (page **88**) can notify you when disk usage, CPU utilisation, or other system resources cross a critical threshold.

If a machine becomes unresponsive or slow, and measurements show that a system resource is being heavily used, this may be an indication that an upgrade is required. If processor usage constantly exceeds 60% of the total, it may be time to upgrade the processor. Slow speeds could also be as a result of insufficient RAM. Be sure to check the overall usage of CPU, RAM, and disk space before deciding to upgrade a particular component.

A simple way to check whether a machine has insufficient RAM is to look at the hard disk light. When the light is on constantly, it usually means that the machine is constantly swapping large amounts of data to and from the disk. This is known as ***thrashing***, and is extremely bad for performance. It can usually be

fixed by investigating which process is using the most RAM, and killing or re-configuring that process. Failing that, the system needs more RAM.

You should always determine whether it is more cost effective to upgrade an individual component or purchase a whole new machine. Some computers are difficult or impossible to upgrade, and it often costs more to replace individual components than to replace the entire system. Since the availability of parts and systems varies widely around the world, be sure to weigh the cost of parts vs. whole systems, including shipping and taxes, when determining the cost of upgrading.

Resources

- JANET Bandwidth Management Review, *http://www.ja.net/services/network-services/bmas/papers/review/BMAS_Bandwidth_Management_Review.htm*
- Linux Advanced Routing and Traffic Control HOWTO, *http:///lartc.org/*
- Linux security and admin software, *http://www.linux.org/apps/all/Networking/Security_/_Admin.html*
- Network monitoring implementation guides and tutorials, *http://wiki.debian.org/Network_Monitoring*
- Optimising Internet Bandwidth in Developing Country Higher Education, *http://www.inasp.info/pubs/bandwidth/index.html*
- Planet Malaysia blog on bandwidth management, *http://planetmy.com/blog/?p=148*
- Rusty Russell's Linux Networking Concepts, *http://www.netfilter.org/documentation/HOWTO/networking-concepts-HOWTO.html*
- *TCP/IP Illustrated, Volume 1: The Protocols*, Addison Wesley, 1994
- Wireless Networking in the Developing World, *http://wndw.net/*