Communication and networks

Table of Contents

Communication..1
Serial and parallel...1
Simplex and duplex..2
Media...2
Synchronous and asynchronous......................3
Bit rate and baud rate.....................................4
Network bandwidth..5
Latency...5
Protocols...6
Networking..6
Packets..7
Types of network devices...............................7
Topology..8
Peer-to-peer and client-server..................10
The Internet and TCP/IP..............................11
Internet services.....................................12
Protocols..12
Standards and the IETF...........................12
Application layer.....................................13
Transport layer..13
Internet layer..14
Link layer..14
Packets in packets...................................15
Network Address Translation..................16

This section is about communication of digital data. It could be studied alongside practical work using the available OS network-related commands and the use of Wireshark to examine packets.

Communication is data being sent from a transmitter to a receiver. An example might be a desktop computer sending data to a printer, over a USB cable.

We often have many devices able to send and receive data between them. This is a *network*. We might have a set of desktop machines connected by cables in a college lab or an office. This is a *local area network*, or LAN. Or it might be a wide area network (WAN), such as JANET.

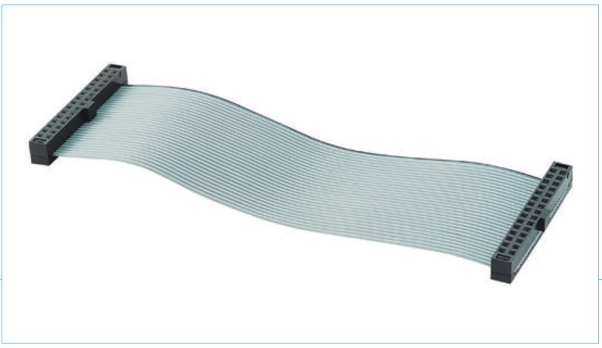
The *Internet* is a network of networks.

Communication

Serial and parallel

Serial communication is one bit at a time, sent one after another.

Parallel is several bits at the same time.



If this is over wires, serial transmission is a single wire, and we send a sequence of bits, one at a time. USB (Universal Serial Bus) is an example.

Parallel means a set of bits is sent at the same time, each on a separate wire. This might be a ribbon cable as shown.

Serial transmission is slower - only 1 bit at a time. Parallel means more hardware is needed - to process each wire.

Simplex and duplex

Simplex is one way transmission. Data can only be sent from one device (the transmitter) to another (the receiver), with no reply.

Duplex is two-way - send and reply.

Half duplex is send and receive, but not at the same time. This is like a phone call. One person speaks, then the other replies.

Full duplex is both send and receive at the same time.

Media

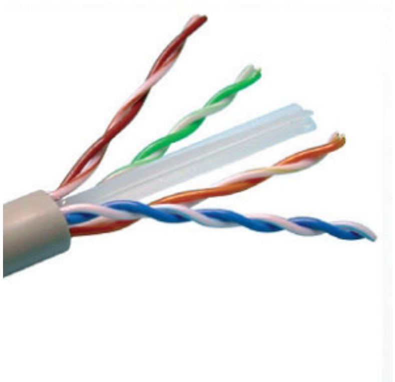
This means how devices are physically linked.

Wire

This might be *coax* or twisted pair. A coaxial cable, or coax, is a single wire, usually copper, surrounded by an earthed mesh, with the two separated by an insulator. The mesh cuts out most of the intereference from other sources.

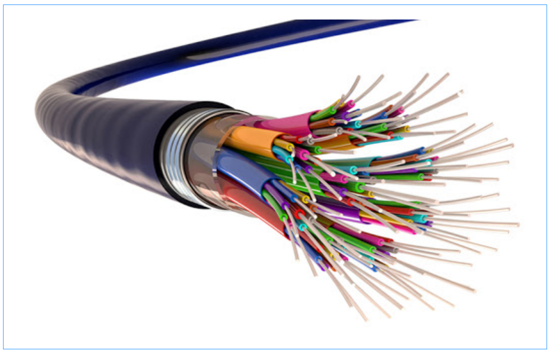
Coax cable

Twisted pair is made of pairs of wires twisted together. Any interference is picked up the same on both wires and so mostly cancels out.



4 sets of twisted pairs

Optical fibre



This consists of a glass or plastic core, down which data is sent as a sequence of light pulses. Usually many cores are bundled into a single cable.

Fibre can transmit very high bit rates over long distances, and is used for the Internet backbone.

Wireless - Bluetooth

Data is sent over radio links, usually microwave frequency.

Bluetooth is very short range - not usually more than 1 meter. One use is to connect a mouse to a PC.

Wi-Fi

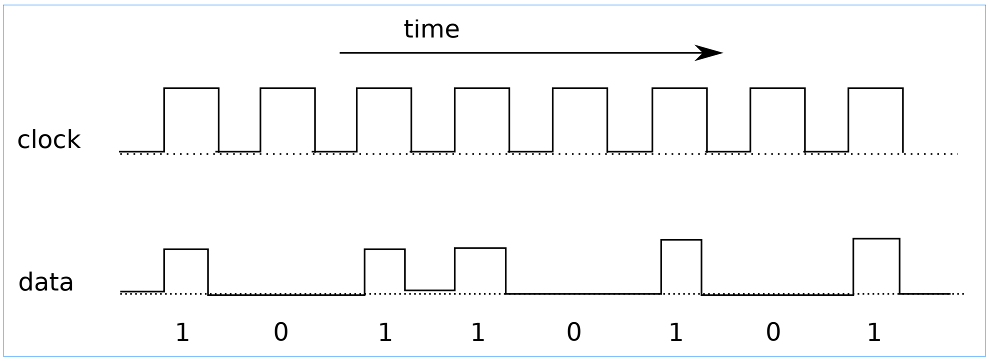
This is microwave wireless, usually used in LANs within 1 building, with a range up to around 20 meters.

Cellular networks

These include 3G, 4G and 5G technologies used for mobile phone and similar use. It uses radio masts in a *cell*, a land area a few kilometers in size. A device connects through a radio mast in the cell where it is, and hands over to a different cell if moved during a connection.

Synchronous and asynchronous

Some systems send a *clock signal* as well as the data. This is a sequence of bits at a fixed frequency - say 9600 pulses per second. The data is only valid at a clock pulse. For example:



shows how 10110101 might be transmitted.

The receiver will only read the incoming data on a clock pulse.

This is *synchronous* transmission.

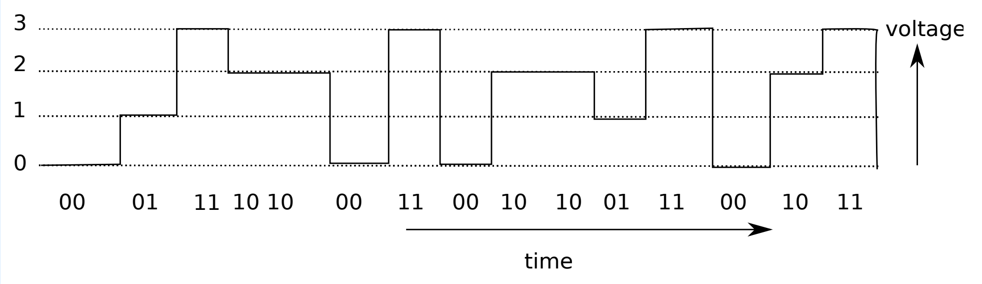
Or, no clock signal is sent. This is *asynchronous* transmission - sender and receiver are not synchronisedby a clock signal.

They must agree some transmission speed - say 9600 bits per second. And usually *start* and *stop bits* are sent. This is a sequence of one bits, to enable the receiver to synchronise its clock with that of the sender - and then read data at the correct time.

Bit rate and baud rate

In a simple system we might just send two allowed voltage levels, with 0 volts meaning a 0 bit, and 1 volt meaning a 1 bit.

But we might have something more complex. For example, we might have 4 allowed voltages - 0,1,2 and 3 volts. Then each pulse can transmit 2 bits, with 0 = 00, 1 = 01, 2 = 10 and 3 = 11. For example:



which shows 00 10 11 10 10 00 11 00 10 10 01 11 00 10 11 being sent.

In this example, we can send 1 of 4 *symbols* at a time - the 4 symbols are 00, 01 , 10 and 11.

The baud rate is the number of symbols sent per second.

The *bit rate is the number of signal transitions per second* (voltage changes per second).

So

baud rate = bit rate X number of possible symbols.

We might not be sending voltages at all - this might be wireless or optical, for example. If the medium is wireless, we might be sending 4 different frequencies. The same ideas apply.

Network bandwidth

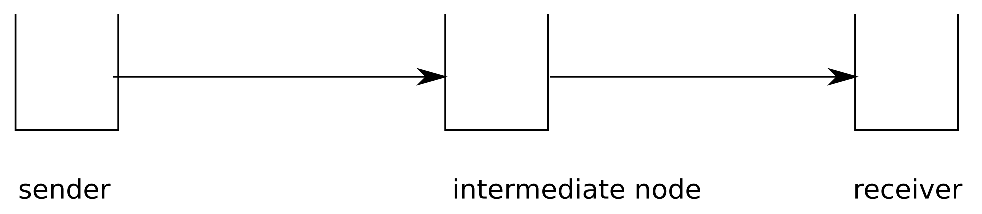
This means the highest bit rate which can be used.

In electronics it means something different. Signals can be sent at different frequencies, and the bandwidth is the difference between the lowest and highest usable frequencies.

Latency

Latency is the time delay between data being sent and received.

This is not the same as bandwidth. For example there might be some device between sender and receiver:



Data might be sent at high speed between devices, but be delayed at the intermediate node.

Latency using a satellite might be several seconds. Over a LAN it might be 10 ms.

Email has very variable latency. On some systems, an email sent to a server becomes available in seconds. On other systems it may not appear for hours.

Protocols

A *protocol* is a set of *rules which sender and receiver agree*, so they can transmit data.

USB has a protocol. HTTP is hypertext transfer protocol, and is how web pages are requested and sent. FTP is file transfer protocol - for the transfer of files. SMTP is simple mail transfer protocol, a way of sending emails.

Test

1. What are the advantages and disadvantages of serial over parallel transmission?

2. What is the difference between half and full duplex?

3. What is the difference between bit rate and latency?

Networking

A network is a set of inter-connected digital devices. There are different types, with ranges of a few meters to the whole planet.

A Personal Area Network (*PAN*) is very short range. An example might be a mouse connected to a desktop PC with Bluetooth.

A Local Area Network (*LAN*) is usually in a single building. A LAN might be in a person's home, or a set of computer labs in a college, or an office building. The devices on a LAN (the nodes) would be desktop PCs, laptops, printers and a gateway to connect the LAN to the Internet. LAN nodes are usually connected by cable or Wi-Fi.

A Metropolitan Area Network (*MAN*) is a set of LANs in a city area, usually interconnected by fibre and wireless links.

A Wide Area Network (*WAN*) interconnects a set of cities across a region or country. An example is JANET, the Joint Academic Network, which connects colleges and universities in the UK.

The *Internet* is a global network of networks, using the TCP/IP protocols. The word Internet comes from *inter-networking*.

This section is mainly abouts LANs.

Packets

Network transmission involves a sequence of data communications - usually not just between a sender and receiver through a single cable connected between them, There may be several intermediate devices.

On most digital networks, the data is split into *packets*. A packet contains a header and a payload. The header will contain the addresses of the sender and the destination. These addresses might be MAC addresses (media access control - unique to the hardware network interface) or the IP address. Network devices use the address to send the packet to the required destination.

If a large amount of data is being sent, it may be split into a sequence of smaller packets. Traffic conditions on the net may change as these packets are being sent. As a result routers may chose different routes for these packets.

This may mean the packets arrive out of sequence. But they contain sequence numbers, so the receiver can re-assemble them into the correct sequence.

The alternative to a packet-switched network is a circuit-switched network. An example is the normal telephone network. Dialling a number causes the network to connect a circuit from sender to destination. This circuit remains connected until the call ends.

Types of network devices

Switch

A switch connects devices, routing packets from one device to its destination device, on the basis of its MAC address. To do this, it learns the addresses of each device plugged into a port, and stores these in a table. A switch is within a single network.

Hub

A hub is similar to a switch, but sends packets from one device to all other devices. Hubs have mostly been replaced by switches.

Router

A router connects 2 or more networks together. It has a table of addresses, but unlike a switch, these are IP not MAC addresses. A home network has a router, so that the set of devices on the home network can use the Internet at the same time, with packets being routed to the correct device.

On the Internet backbone, large scale routers route traffic over fibre.

Gateway

A gateway connects two networks using different protocols.

Wireless access point

A wireless access point (WAP) has aerials or antennae to send and receive mircrowave, and enables devices to link to a LAN using Wi-Fi

Modems

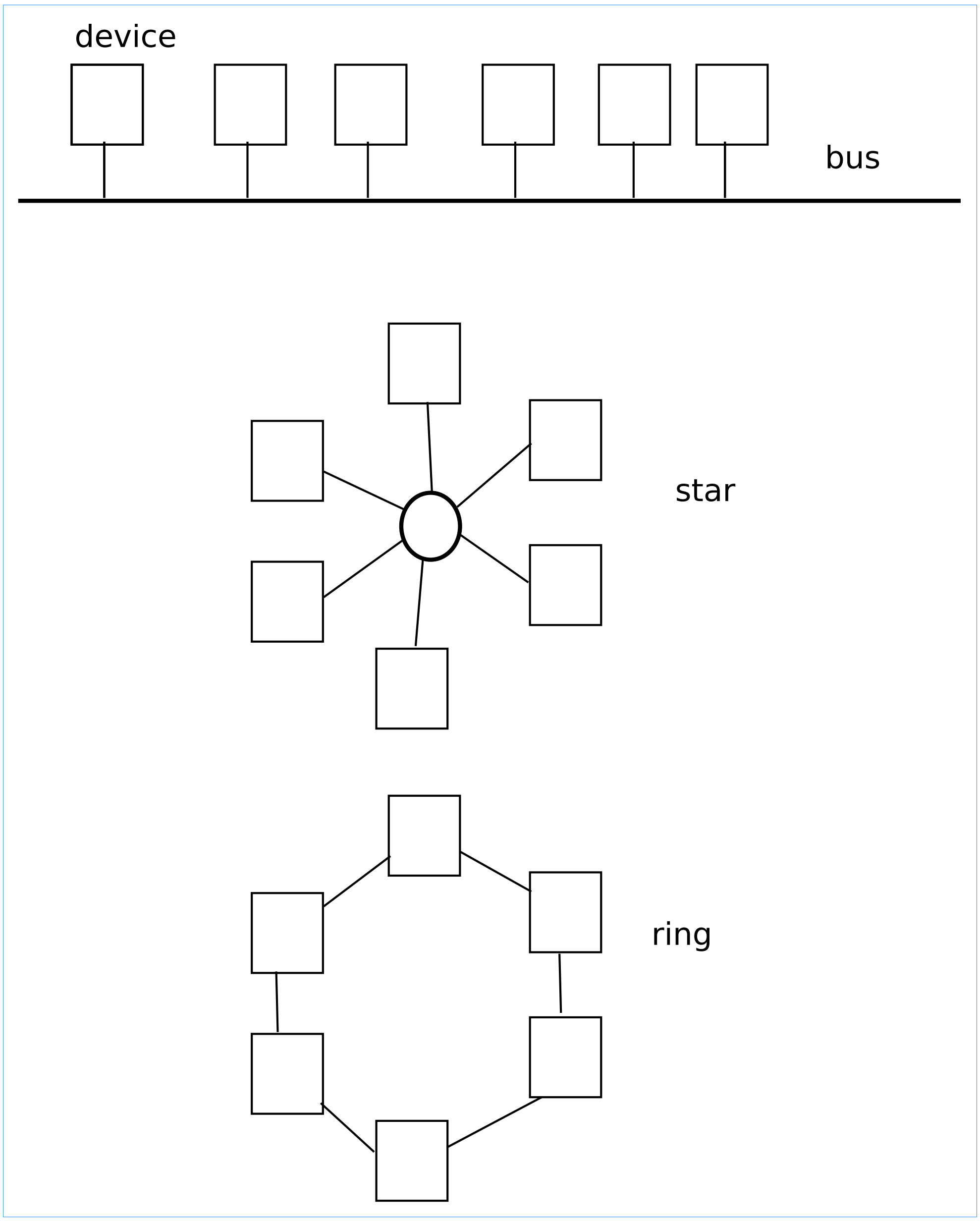
A modem converts digital signals to analog form, and the reverse, in order to send LAN data over a medium such as telephone line or radio. Home networks usually have a modem to connect through a telephone landline to the Internet. They may also have powerline ethernet, which puts diital data onto AC power lines.

Physical devices often combine several of these functionalities. For example home networks usually have a device which is a modem, gateway, router and WAP.

Topology

This means the shape or pattern in which network nodes are connected. This is usually for a LAN.

There are 3 standard topologies - *star bus* and *ring*:



The ring topology was used in IBM Token Ring networks from 1985, and now largely replaced by Ethernet cabling in a star.

In a bus topology all devices are connected onto a single cable. This is simple, but has the problem of a collision, when 2 stations try to send at the same time. This is fixed by re-transmitting the data, but this slows the process. Because of this the bus topology has become less popular than star topolgies with a switch.

For a star, each device is connected to a central item, probably a switch. The switch only sends data to its destination. The only issue is if two stations try to send to each other at the same time, but modern cabling is full duplex, so collisions do not occur at all.

There is a difference between *physical and logical topologies*. Physical means how they are actually connected by cables. Logical topology means how data packets move between devices.

For example in a physical star topology, if the central item is a switch, this routes packets from source to destination only. So this is as if each device is only connected to the other device it is sending data to.

Peer-to-peer and client-server

In *client-server*, one device has special status, as a server, with a client usually meaning a device an individual user is using.

For example on the world-wide web, a server sends requested web pages over the Internet to clients using a tablet or mobile phone or desktop PC.

On a LAN, a server provides storage to users on client devices. Users' files are stored on the server, so that a user can use any network device. The server must provide security, requiring users to logon to gain access to their files.

The term server can mean either hardware of software. A hardware server is a computer used to run server software. It may not have keyboard or screen, since no-one uses it directly.

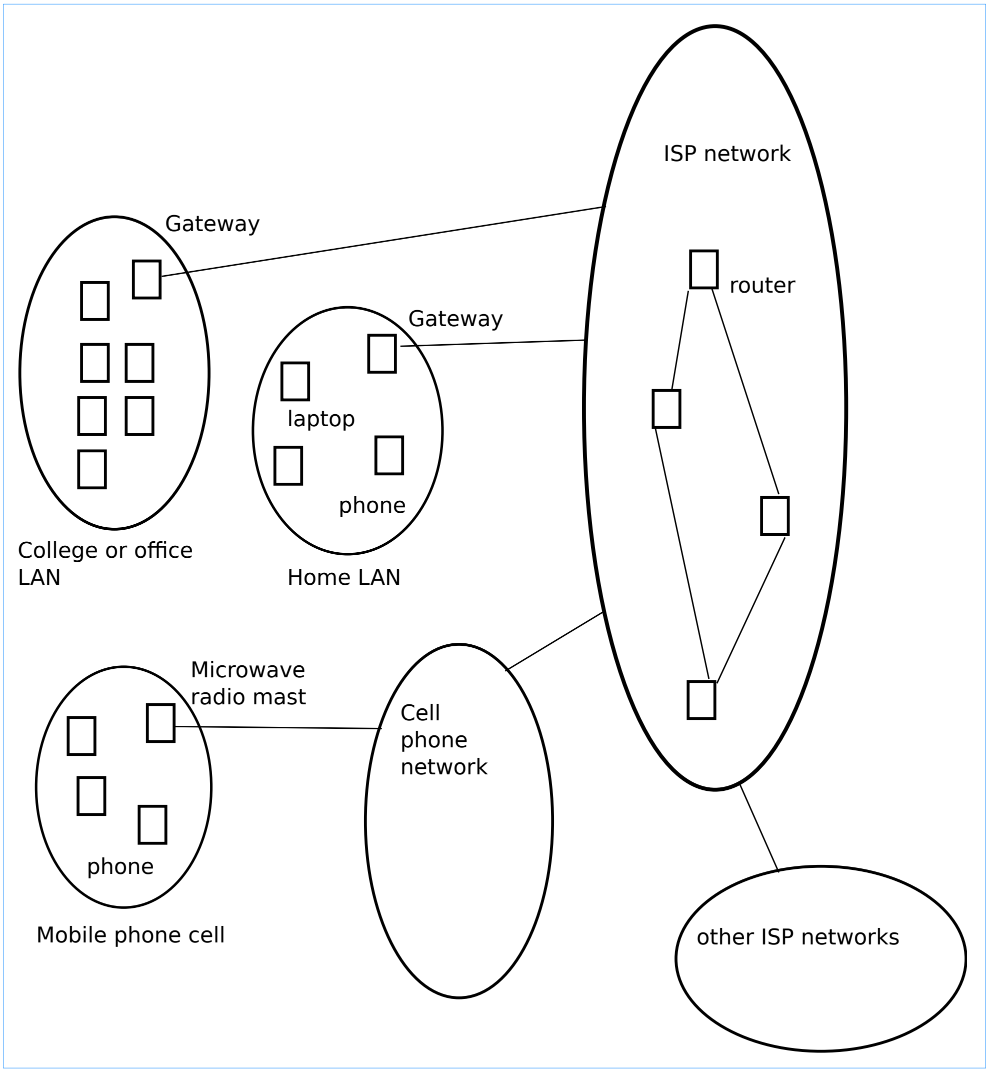
There are different kinds of server software. A program to send web pages is an http server, since it uses the http protocol. Apache is the most common version. A database server provides access to a database. MySQL is an example. A server will usually run several pieces of server software. A common set would be an Apache web server, a MySQL database server, and a PHP interpreter to execute PHP scripts on the server.

On a *peer-to-peer* network, there are no servers. A user normally uses the same device, and users' files are stored on that device.

The Internet and TCP/IP

The *Internet is a set of LANs connected together*. It started as connections between LANs at universities, together with ARPANET, a network of the US Department of Defence.

This indicates an outline structure:



An Internet Service Provider, ISP, is an organisation, mostly commercial, which provides facilities for Internet access.

In a college or office LAN, devices might be connected by Ethernet to a gateway which connects to an ISP network. A home LAN might be connected by Wi-Fi through a gateway, and this might include a mobile cell phone.

Or a cell phone might be linked on 3G, 4G or 5G through a cell radio mast, routed through the cell phone network to the ISP network.

The ISP networks are connected.

Internet services

The 'world wide web' is one of these services, displaying web pages.

But that is just service available over the Internet. Others include file transfer and email.

Protocols

A protocol is some kind of agreement or standard or mutual understanding between sender and receiver to enable the communication to happen.

There are now thousands of different protocols in use.

A protocol *model* is a way of thinking about protocols, to structure and organise them, to classify them.

There are 2 common protocol models - OSI and TCP/IP.

OSI is a theoretical model.

TCP/IP is a protocol suite - a set of protocol types, organised into 4 'layers'. The Internet has developed using TCP/IP.

The layers are

1. The application layer
2. The transport layer
3. The Internet layer
4. The link layer

Standards and the IETF

The Internet Engineering Task Force IETF is a loose volutary organistion of computer scentists and engineers who agree standards and similar ideas concenring the Internet. It cannot enforce anything - but since this is about communication, anyone who does not comply with an IETF-agreed standard is left isolated.

This links to the IETF website.

The IETF publishes RFCs - requests for comments, which discuss topics and announce agreed standards.

Application layer

The application layer protocols each relate to some application type. For example, the application might be about transferring files across the Internet. That would use the FTP protocol. Or it might be a web browser, intended to fetch and display web pages. This would use the http protocol. Or if the application is an email client, this would use POP to receive and SMTP to send emails to a mail server.

These applications are not concerned about *how* the data is sent or received. They simply use this service provided by the other 3 layers.

Transport layer

These are protocols about how data is sent and received in general terms - so nothing about, for example, if it goes over fibre or twisted pair.

The main protocols are TCP and UDP.

TCP is the Transmission Control Protocol. It is *connection-oriented*. If a client wishes to communicate with a server, the following sequence happens

1. The client requests a connection, and the server confirms
2. Client and server send and receive.
3. Client requests termination, server confirms.

Any of these might fail, but if it does, no acknowledgement is received, so the sender knows there has been a failure, and can repeat the send.

This makes TCP reliable.

UDP is the the User Datagram Protocol. This is *connection-less*. The sender simply send the data - called a *datagram*. There is no acknowledging reply. This makes UDP faster but unreliable.

UDP is used for simple query-response situations. One example is NTP, Network Time Protocol, which a device can use to check its time setting. The device sends a request to a time server. This might reply with the current time. If it does not reply, the device can simply use a different time server.

Internet layer

The Internet layer protocols are about connecting networks together. The Internet is about inter-networking.

The main protocol is IP - the Internet Protocol. This puts the data to be sent into a packet, with a header which includes the sender and destination IP address. This enables routers to direct the packet as required.

There are 2 versions of IP - IPv4 and IPv6. IPv4 is the earlier version, and has IP addresses 4 bytes long - which limits the number of possible distinct addresses to around 4 billion, a limit which may be reached soon. IPv6 has 6 byte long addresses. IPv4 is still the most commonly used.

The maximum size of an IPv4 packet is 64 kB. This means data has to be split between several packets, called fragments, each carrying a sequence number. Packets are routed dynamically, so different packets may arrive out of sequence. The receiver can re-assemble the data in to correct order - but packets may be lost.

IP is connection-less - so there is no guarantee of correct delivery. The transport layer might use TCP to establish a connection and confirm delivery.

Link layer

The link layer is concerned with communication *within a network*. So a computer on a college LAN, for example, fetching a web page from a remote server, must first have that sent to a gateway, possibly over twisted pair. The link layer does this.

One protocol is ARP, address resolution protocol, which relates hardware MAC addresses to IP addresses. Others include his relating to the actual connection, such as Ethernet over coax cable or twisted pair, or Wi-Fi.

Packets in packets

The protocol layers work together, with the packet produced by one layer being treated as the data for the packet produced by the next layer.

As an example - suppose someone types a web address into the address bar of a web browser and hits RETURN.

What happens in outline is

1. The web browser constructs an http packet. This contains the server web address, and the http command - GET

2. That data is the payload for a TCP protocol packet, which has a header of port numbers. The destination port will probably be 80, because http servers normally use port 80

3. That is the payload for an IP packet. That adds a header of IP address of destination and source.

4. That is the data for a link layer packet, which has a header including the hardware MAC addresses of the sending machine, and the next device in the route, which might be the first gateway router. As the packet moves across the Internet from router to router, this link layer packet is split apart, and the MAC addresses replaced by those for the next link:

http protocol				URL and GET
TCP protocol			port numbers	URL and GET
IP protocol		IP addresses	port numbers	URL and GET
link layer	MAC addresses	IP addresses	port numbers	URL and GET

The idea here is that the http packet is the payload within the TCP packet, and so on.

In fact TCP is a connection-based protocol. So to start with the browser sends TCP connect packets to the server to set up a connection. If that works, it sends the http GET as shown. The server replies sending back the requested web page (or 404s or whatever).

If the webpage has links to JavaScript script files, or CSS style sheets, or images, the browser will send more GET requests for these before it can display the page. At some later point, the TCP connection will end.

The web is not simple.

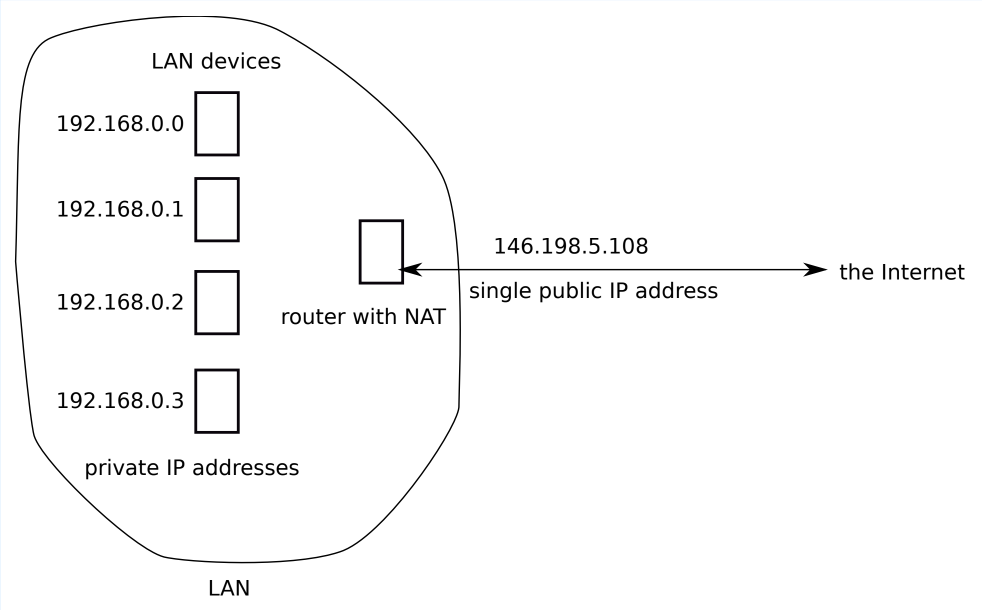
Network Address Translation

In fact there are two kinds of IP address:

Public IP addresses. These are unique - only one device can have a given IP address. They are usually assigned by an ISP, from blocks allocated from IANA through regional authorities.

Private IP addresses. These belong to the devices on a LAN. For example one set of private IP addresses is 192.168.00.00 to 192.168.255.255. There are no public IP addresses in this range. But, on different LANs, many devices might have the same private IP address.

LAN gateways include a process called network address translation, or NAT. This maps the LAN-side private addresses to a single public address onto the Internet:



Each of the LAN devices will send and receive packets to different servers on the web at the same time, and must each recieve the correct replies, even though they all share the same public IP address.

One way the NAT router can do this is by modifying the port numbers of the TCP and UDP packets it passes on.

Test

1. What is the difference between a hub and a switch?

2. Outline the 4 TCP/IP protocoal layers

3. What in a connection-based protocol? Contrast them with a connection-less protocol.

4. Why is NAT used?