Internet

The Internet is what holds the Web together, and it is driven by protocols

history ip tcp udp dns

The Internet

What is the difference between the Internet and the Web? Surprisingly, many people don't know

The Internet is the hardware and software infrastructure which joins the world's computers

The Web is an application which runs on the Internet platform

However, it is arguably the most important application, and it tends to soak up all the others

History

Before the Internet, local area networks used protocols like ethernet, which is still used to connect nodes, i.e. hubs, bridges, switches, routers

Then Packet Switching was invented and they gradually all got joined up

Today, there are huge numbers of computers around the world, connected by wires, fibres, radio, and all sorts, with a big variety of transmission protocols

What is it that turns this bunch of connected computers into an internet?

Ethernet

Computers were originally joined in pairs by wires

A couple of simple early protocols survived a long time, as *serial* and *parallel* ports on the back of PCs

Many protocols were developed for joining several computers together via LANs = Local Area Networks, but the superior ones (Cambridge Ring, Token Ring,...) were out-marketed by the Ethernet

See the next few slides for more Ethernet details

Basic details

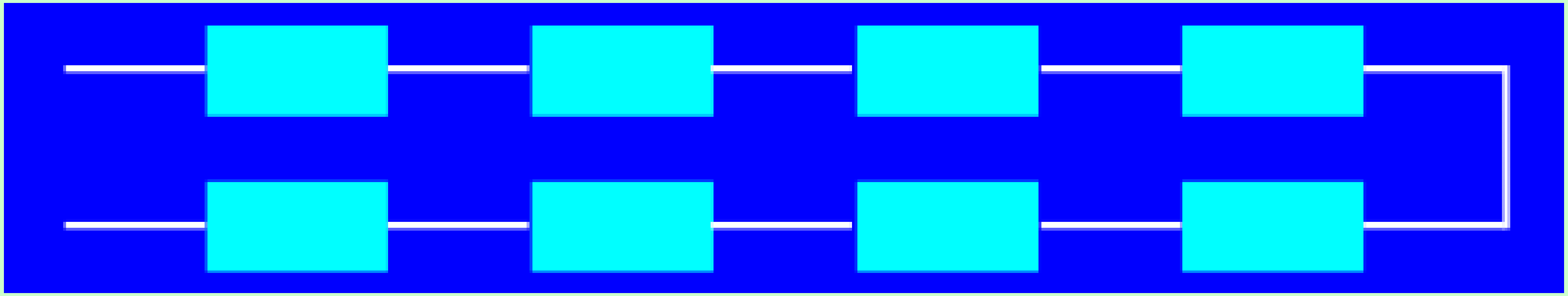
The Ethernet protocol joins computers by wires ("twisted pair" or shielded) or optical fibres or radio

The transmission speed is the speed of light in glass or copper or air, 66% to 99% of SOL-in-vacuum

The transmission frequency is 10Mb or 100Mb or 1Gb or … in bits per second

Layout

Originally, an ethernet would go up one side of a corridor and down the other, joining offices



Frames

A computer *broadcasts* a *frame* onto the "wire" and only one of these frames happens at once

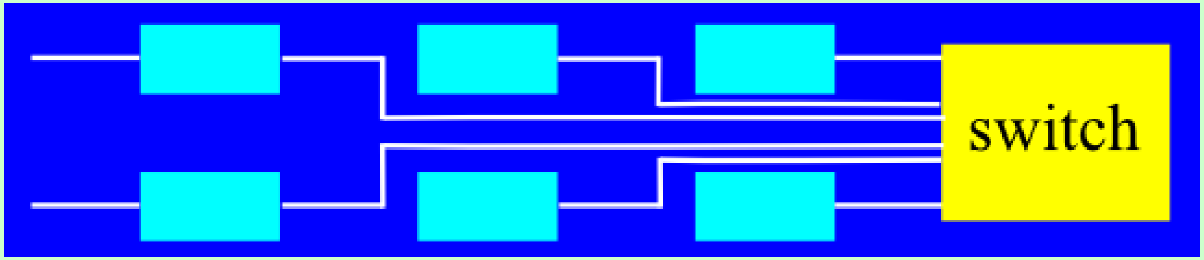
The recipient recognises its own address, and reads the frame, while all other computers ignore it

All the computers share the bandwidth (10Mb, ...), there is no central control, just collisions where two computers try to start a frame at the same time

Collisions

A computer detects a collision (what it reads from the wire is not what it wrote) and backs off for a random amount of time, which 'doubles' with each attempt

Each computer is connected by an ethernet to a hub, bridge, switch, router, or another computer



Now the bandwidth isn't shared (except for hubs)

MAC addresses

Every Ethernet card has a factory id number (the ethernet or *MAC* = *Media Access Control* address)

"Address" is misleading because it has nothing to do with where the computer is, it only works locally

A frame has "to" and "from" MAC addresses in it, and a body; although the body can be arbitrary data, it is usually an IP packet

Longer distance connections are made by Ethernet-like protocols (but higher bandwidth) including transatlantic cables, satellite links etc

Nodes

A *node* is anything which is connected up by the internet – a hub, bridge, switch, router or computer

The words hub, bridge, switch, router have fairly simple, precise, logical distinctions between them

In practice, not all devices are that easily classified, and people tend to mis-use the words anyway

Nevertheless, for anyone wanting hardware expertise, it is worth knowing about them

See the next few slides for more Ethernet details

Hubs

a *hub* re-transmits ***analogue*** signals, is purely local (***no*** connection to "the rest of the world"), and doesn't need to know where to send data

Hub is the modern name for *repeater* or *concentrator*. It has several wires or fibres plugged into it. It takes the signal on each, and re-transmits it on the others. It does this in a purely analogue way, without reading any of the digital information. So it is usually a very small and cheap box (though other devices like switches have now become small and cheap too).

Hub features

The wires connected to a hub still form a single logical Ethernet, so only one transmission can be in progress at once

This network can have a star shape, making it easier to connect many computers

A hub can amplify the signal and/or improve its quality, which may mean the wires can be longer or the number of computers connected can be greater

Bridges

A *bridge* (see wikipedia entry) is a more sophisticated type of hub, and is relatively rare now that switches have become cheap

It monitors the signals digitally, and so can read the message headers, but otherwise retransmits messages in analogue form without storing them

It gradually builds up a table giving the id of each computer and which segment it is on, which it finds out when the computer first transmits

Bridge features

The segments of a bridge must all be of the same type (or at least run at the same transmission frequency)

Before finding out about the computers, the bridge uses broadcasting, like a pure hub, forming a single logical Ethernet

after finding out about the computers, when two computers on the same segment communicate, the communication is not transmitted on the other segments, so the segments become as efficient as if they were separate Ethernets

Switches

A *switch* re-transmits ***frames***, has exactly ***one*** connection to "the rest of the world", and needs only a local table to know where to send data

A switch reads and stores a whole message (frame) before deciding whether or where to re-transmit it.

Switch features

3m

The segments can be separate Ethernets

The segments can be completely different, e.g. slow Ethernet over wires and fast Ethernet over fibre

A switch only has a table of local computers, which it builds up over time, and so can only cope with one 'global' link to the rest of the Internet

Routers

A *router* re-transmits ***IP packets***, has ***two or more*** connections to "the rest of the world", and therefore needs a global table (requested from other routers) to know where to send data

A router not only understands frames, but also the IP packets being carried in them. It deals with local ids (MAC addresses) and also IP addresses. It communicates with other routers (e.g. well known national or international ones) to gather information about which (ranges of) IP addresses should be sent where.

Router features

A router can connect different Ethernets, not just segments of a single Ethernet

It has two or more links to "the rest of the world"

It is responsible for congestion control and other high-level issues

It needs quite a lot of effort to set up and maintain one

History of Packet Switching

There is a general rule in the history of technology that the fame goes not to the original inventor, but to the runner-up who made the idea popular or commercial, or just stole it (exercise: who were the real inventors of the light bulb, telephone, radio, television, computer, ...)

Packet switching is often said to have been invented by (D)ARPA = *(Defense) Advanced Research Projects Agency*, with variable routing in order to to make networks robust against nuclear attack

See the next few slides for packet switching details

Non-nuclear

The nuclear motivation is almost certainly wrong

It probably arises from the fact that in the USA, military sources of research funding are so important that military applications are regularly invented for projects to attract grants

The ***real*** motivation for variable routing was almost certainly to cope with the terrible unreliability of hardware and networks in those days

Non-USA

More importantly, attributing the invention to ARPA gives the false impression that the Internet started in the USA and spread to other countries

Packet switching was actually invented at Arpanet in the USA, and the NPL Network in the UK, ***at the same time in 1968***, and the USA and Europe freely shared ideas after that as networking developed

Growth

The USA network, the European network, and other networks, and the links between them, grew simultaneously until people started thinking of it as a single global network around the mid 1980s

USA and Europe sources compete over the history of the evolution of the internet - e.g. the first long distance connection was in the USA, but the first connection between two countries was in Europe

Wikipedia articles are now reasonable, but still tend to oscillate a bit as USA or European authors update them

Protocols

Protocols are communication conventions, and they stack up on top of each other in layers, and names for the higher level protocols go on the front of computer names as prefixes with a colon, e.g.

ip, tcp, udp, dns (low level)
http:, https: hypertext transfer (i.e. web)
ftp:, ftps: file transfer
ldap:, ldaps: lightweight directory access
smtp:, imap:, imaps:, pop3:, pop3s: email
ssh: secure remote login

The IP Protocol

The Internet is formed out of *nodes*, i.e. computers and hubs, bridges, switches and routers by a world-wide agreement about the format of data

IP = Internet Protocol is a "logical" protocol which sits on top of each "physical" protocol such as Ethernet (but there can be many layers, not just 2)

IP Packets

The unit of IP is the packet, a variable sized block of data (up to 64k) with a header (20B or more) containing a sender and receiver etc

The Internet works by packet switching and not frame broadcasting, which doesn't scale up

Store and Forward

Each 'physical' protocol defines how it transmits IP packets, e.g. Ethernet sends one in several frames

IP has a convention for partial packets, so once a packet has been split, it doesn't have to be recombined until it reaches its final destination

Packets are moved using "store-and-forward": a packet is stored in a buffer in memory in a node, then forwarded in the right direction using switching or routing tables

Unreliability

IP provides no reliability:

packets may not arrive,
or may arrive twice (retries),
or may arrive out of order

Links are (now) quite reliable but congestion is still handled by deleting excess packets, because any 'better' way would add to the congestion

IP was designed for ftp etc., not voice or video, so buffers are too big, delay is too high (& loss is too low)

IP Addresses

IP relies on every computer on the Internet having an IP address, used in routing tables

An IP address (IPv4=version 4) is 32 bits, usually written as 4 decimal numbers, each between 0 and 255, e.g. 137.222.102.187

IP addresses are allocated hierarchically in blocks, e.g. the UK gets a large block, gives a medium subblock to UOB, which gives a small block to CS

The world is running out of IP addresses, several interim tricks are being used, and IPv6 is coming

Getting Connected

So, how do you get connected to the Internet?

The best option is to buy a broadband or leased line connection from an ISP = *Internet Service Provider*, plus high (upstream) bandwidth, plus a *static* IP address (or small block of addresses)

The 2nd inferior option is for your ISP to give you a *dynamic* IP address, where the IP address may change each time you connect (and typically a low upstream bandwidth)

Running Servers

With a static IP address, there is no problem - just run your own server

With a dynamic IP address, the 2nd best option is to run your own server and use a service like noip.com or dyndns.com to broadcast changes to your IP address.

The worst option is to pay an ISP (or use a free ISP) to give you disk space, and a share of a server, but you then have ***no*** control over server configuration.

The TCP Protocol

TCP = Transmission Control Protocol is the most important general purpose protocol on top of IP

Many of the higher level application protocols sit on top of TCP

It is often called TCP/IP (read "TCP over IP") since TCP could in theory run on top of other protocols, but it never does, so it is a senseless name (and every protocol runs over something else, so why aren't all protocols described that way?)

What TCP Does

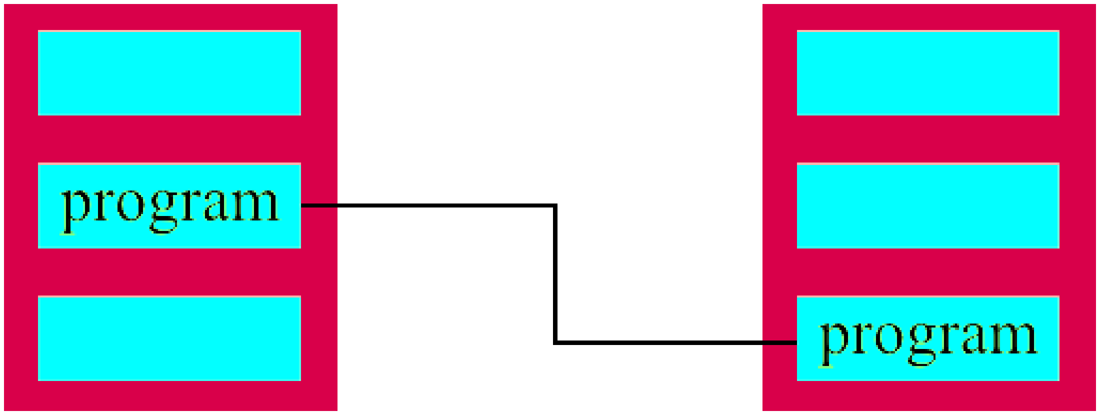
TCP makes a long lasting connection (session) between two computers, adding *reliability*, using acknowledgements, timeouts, retries

It adds congestion control, by monitoring ack timings, and discourages anti-social behaviour, by slowing down and speeding up appropriately

It is highly tuned for optimum overall efficiency of the Internet, but not the efficiency of individual connections, nor does it cater for the real-time streaming of sound or video

Port Numbers

IP provides a way for ***computers*** to communicate, but it is really ***programs*** that need to communicate:



TCP includes a *port number*, as well as an IP address, i.e. an index number into a table of programs (like having numbered connectors on each computer)

Standard Port Numbers

Port numbers from 0 to 1023 are reserved for standard services, 1024+ can be used for anything

For a server program to use 0-1023, it must have root or administrator privileges

Examples are (don't memorise!)

```
http:80 https:443 ftp:22 ftps:990
ldap:389 ldaps:636
smtp:25 pop3:110 pop3s:995
imap:143 imaps:993
ssh:23 dns:53 doom:666
```

The UDP Protocol

UDP = *User Datagram Protocol* is an alternative protocol, instead of TCP, over IP

It is very simple, and in fact it is essentially just raw IP, plus the same port numbers as TCP

It sends a single unreliable one-shot message in a single IP packet, and it is meant for simple lookups, acknowledgements, timeouts, and so on

It is now being used for some sound and video streaming applications, which is not a good idea

Streaming

The problem with UDP streaming is that it spoils the overall internet efficiency effect of TCP

There's a theory which says two of the parameters throughput, delay, loss determine the third (see Davies, Holyer and Thompson) and you can reliably support both ftp-like traffic (maximum throughput) and sound/video streaming traffic (minimum delay)

The theory can be used in hardware using tiny buffers

It is tragic that the telephone network hasn't been merged with the Internet

The DNS Protocol

Computers can use just IP addresses alone, but since communications are started by humans, a readable version is needed, called domain names

For example, there is a name **www.cs.bris.ac.uk** corresponding to the IP address of our web server

There are aliases, so **cs.bris.ac.uk** is an alias for **compsci.bristol.ac.uk**

The first part of a full name is (an alias for) a computer, e.g. **www.cs...** is an alias for **sal.cs...**

Domains

Suffixes are domains representing local networks with their own sets of IP addresses, e.g.

```
cs.bris.ac.uk, bris.ac.uk, ac.uk, uk
```

Once, domain names had geographical meaning, and each country had a two-letter suffix, like **uk**

USA was an exception, so **.com** and **.edu** are the USA equivalent of **.co.uk**, and **.ac.uk**, but geography has diminished, so owning **holyer.com** doesn't make my family an American company!

Domain Name Service

DNS = *Domain Name Service* is the protocol for looking up a domain name to find its IP address

Use whois, dig, nslookup, winipcfg, ipconfig, and equivalent web sites to use dns services manually

There are many registrars, so for best info, first find the right registrar, then look up their database

(Multi-word domain names with no hyphens are potentially ambiguous; see some examples)

Domain Name Servers

A "server" means both the computer and the program running on it which provides a service

A DNS server is a (computer or) program which looks up domain names and gives back IP addresses and provides other related services

Each LAN needs a DNS server, so every computer can convert names to numbers

For distant domains, the local DNS server uses a *registry*, i.e. a national DNS server

Getting a Domain Name

You can buy a domain name very cheaply (e.g. at ukreg.com), .co.uk names are cheaper than .com

When you get a domain name, you specify an IP address that it refers to (if you change it, it takes up to a day to filter through the net)

The IP address refers to your computer, or a forwarding service (or a DNS)

Note: you can transfer ownership of an ISP account or domain name, without the ISP doing it