The structure of the internet

A short history of the Internet and the World Wide Web

The internet is the largest structure of interconnected networks set up to allow computers to communicate with each other globally using standardized communication protocols.

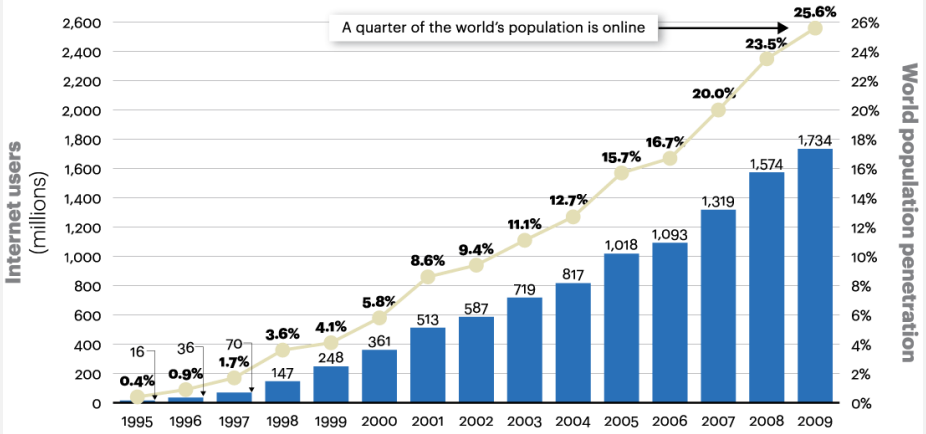
A United States defence project set up in the 1960s created ARPANET (ARPA) to enable distant departments to communicate while working on the same project without the need for physical travel. The idea of the internet was born, as the concept developed and communication technology improved.
In 1995, the Internet became fully commercialised to the public so user numbers increased (reaching 2.5 billion ~1/3 of the population in 2015).

The World Wide Web is a **collection of web pages** that **reside on computers connected to the internet** (WWW). It uses the **internet as a service to communicate the information contained within these pages**.
The concept of the WWW and **using browsers with search engines** to search for information contained within it was first **developed by Sir Tim Berners-Lee**, a British scientist working at CERN, in Geneva Switzerland.
The World Wide Web and the internet are not the same things, even today the internet may be used without the World Wide Web.

For example, during a Skype call or when using an atm. The internet is being used but not via the World Wide Web.

The World Wide Web is a collection of web pages that reside on computers connected to the internet (WWW). It uses the internet as a service to communicate the information contained within these pages.

Figure 1
Global Internet users and penetration rate (1995-2009)



Sources: Nielsen, ITU; A.T. Kearney analysis

The development of the **Web 2.0, changed from static web pages (Web 1.0) to dynamic or user-generated content** and allowed the growth of social media. Users became more active as content could be easily added and additions could be made to the internet.

The Physical Structure

Each continent is **connected by backbone cables connected by trans-continental leased lines fed across the seabeds**. **National Internet Service Providers (National ISPs)** connected directly to this backbone and distribute the internet connection to a **smaller company owned ISPs** who in turn provide access to individual homes, sites and businesses.

A leased line is a **private bidirectional** or symmetric telecommunications line between two or more locations allowing data to be transferred.

An internet service provider (ISP) is a company that distributes internet connections to homes and organisations for a subscription fee.

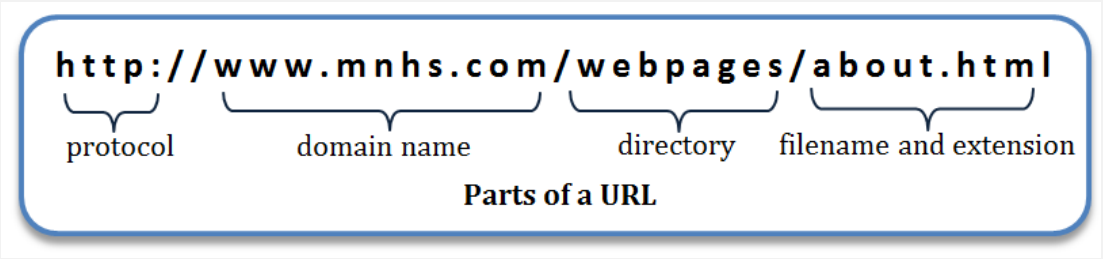
A national internet service provider (ISP) is a large telecommunications company connected to directly to the backbone of the county/continents internet connection and sells the bandwidth/connections onto third-party internet service providers.

Uniform Resource Locations (URLs)

A uniform resource locator (URL) is the full address of an Internet resource. It will **specify the location** of a resource on the internet, including the **resource name** and usually the **file type,** so that a **browser can request it** from the **website server.**

For example, https://www.domainname.com/folder/subfolder/webpage.html#element

- http = The protocol method (hypertext transfer protocol (secure))
- www.domainname.com = The host website /hostname
- webpage = Specific location or path with file extension
- element = resource (element within the file)



Parts of a URL

Internet registries and registrars

Internet registrars hold the records of all existing website names and they also have details of the domains that are available to purchase.

These are companies **act as resellers for domain names** and allow people and companies to purchase them. All **registrars must be accredited** by their **governing registry**.

Internet Registries are five global organisations governed by **the Internet Corporation for Assigned Names and Numbers (ICANN)** with worldwide databases that **hold records of all the domain names currently issued** to individuals and companies **and their details**.

Details include the registrant's:

- **Name**
- Type (**company or individual**)
- Registered **mailing address**
- The **registrar that sold** the domain name and **the date** of registry.

The registers also allocate IP addresses and keep track of **which address(es) a domain name is associated with** as part of the Domain Name System (DNS).

Domain names and the Domain Name System (DNS)

A domain name is part of a network address which identifies it as belonging to a particular domain or the location that an internet resource resides in.

These are structured into a hierarchy of smaller domains and written as a string separated by full stops as dictated by the rules of the Domain Name System. It will include all parts of "www. Google .com"

A Domain Name System (DNS) is the Internet's system for converting alphanumeric domain names into numeric IP addresses that relate to a particular location of a web server that the domain belongs to.

Hierarchy of smaller domains

Top level Domains (TLDs to 3rd Level Domains (3LDs)

- **Generic TLDs = .com .org .edu .ac .net**
- **Country TLDs = .uk .fr .de**
- **2LDs = .co .gov .sch**
- **3LDs = .bbc .ebay .lidl**

e.g. www.ebay.com.au or www.bbc.co.uk (.ebay = 3LD, .com = Generic TLD, .au = Country TLD)

Fully Qualified Domain Name (FQDN)

A fully qualified domain name is one that includes the host server name (e.g. www, ftp, mail) depending on the type of webserver the resource is hosted upon.

This depends on the resources being requested by the URL say for a webserver we would use www.insertnamehere but for a webserver mail.insertnamehere
This depends on the type of webserver that the resource is hosted upon.

Each domain name will have **one or more equivalent IP addresses**. The **DNS catalogues** all Domain names and IP addresses in a series of global directories that **domain name servers** can access in order to find the correct IP address location for a resource.

When a web page is requested by sending data packets. A user enters a URL into the browser so packets of data are sent to request the corresponding IP address from the local DNS. If the DNS does not have the correct IP address, then the search goes up the hierarchy to another larger DNS database. The domain is linked to an IP address in the database and a data request is sent by the user's computer to that IP location to find the web page data.
A similar return journey sends data packets for that webserver resource back to the user's computer.

We use domain names using URLs that are made up of alphanumeric strings separated by **full stops and slashes** since this makes it easier for a human user to **interpret, remember and type in**. It is much harder to remember large numbers for IP addresses that **don't always follow the same memorable order**. DNS simply translate a domain name from a **URL request** into the IP address and **so the data request by the user computer can be sent to the server location of the domain.**

IP addresses

An IP address = unique addresses assigned to a network device using the Internet Protocol.

It is made up of **4 numbers** of **up to three digits**, ranging from **0 to 254,** such as 192.168.1.1

Whenever data is sent over a network, the **IP addresses of both the sender** and the r**eceive**r must be included; typically network communication takes the **form of request and response**. The destination IP is needed to ensure each packet is routed to the correct destination and the **source IP is needed so response** packets can be sent back.

An IP address is also called an **Internet Protocol** address.
A domain name is associated with a specific IP address for the web server that the website resides on.

Data Packets

Data that is to be transmitted across a network is broken down into more manageable chunks called **packets.**

The **size of the data packets may be fixed or variable** (most between 500 and 1500 bytes). Every packet contains a **header,** for metadata, and a **payload** containing the body of the data being sent. Some may also have a **trailer** containing a **Cyclical Redundancy check (CRC checksum)** that **detects errors/corruption** in transmission.

The header

- ***The sender's and the recipient's IP addresses***
- ***The protocol being used with this type of packet***
- ***The number of the packet in the sequence being sent.***

They also include a **Time To Live (TTL) or hop limit**, after which point the **data packet expires and is discarded.**

Packet - E-mail Example

Header	Sender's IP address Receiver's IP address Protocol Packet number	96 bits
Payload	Data	896 bits
Trailer	Data to show end of packet Error correction	32 bits

The Payload

This contains **the actual data being sent**. Upon, receipt, the **packets are reassembled** in the **correct order and the data is extracted**.

The Trailer

Some packets may also contain a **checksum** or a **Cyclical Redundancy Check (CRC).**

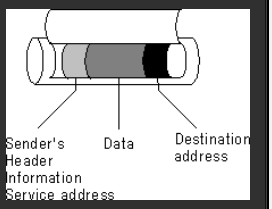
This is to **detect errors that may occur during transmission**. This is done by **attaching and creating a hash total** that is calculated from the data contained within the packets.

Commonly, the **hash total will check for the number of 1's** in the binary data in the transmission. The **CRC checksum is recalculated for each packet upon receipt** and matched to help **verify that the payload has not changed during transmission**. If the CRC totals differ then the **packet is refused with suspected data corruption** and a **new copy is requested** form the sender.

This is very similar to the parity bit used in binary. A parity bit identifies errors by checking the total number of 1's in the string – is it even or odd?

How is information sent over the internet?

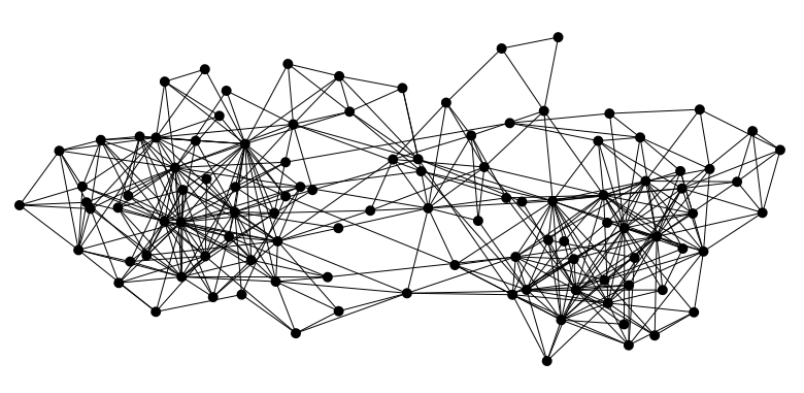
1. The user **enters a URL** into a browser.
2. The **browser extracts the domain name and sends this to the local DNS server** as a packet of data.
3. The **packet of data is extracted at the DNS server** and **queries the local DNS for the IP address of the domain by searching the DNS catalogues**.
   If no IP address is found for that domain then it goes up the hierarchy and is sent to **a larger DNS server** until the **domain is linked to an IP address in the database**.
4. The **IP address is returned as a packet of data to the users' computer** and then **the user's computer directs the request to the IP location** of the **webpage server** to request to view the page.
5. The **information on the webpage stored on the webserver** is sent back to the users' computer through **packet data** once again and the user can view the website.

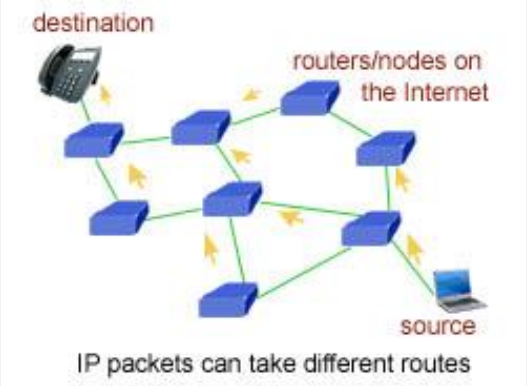


Routing packets across the internet

A large success of information transmission over the internet is through **packet switching.**
The **packets of data** are sent **along entirely different routes** between the **sender and recipient (nodes).** At the moment a packet leaves the sender's computer, the **least congested or fastest route** is taken to the recipient's.
Packets travelling across the internet will **traverse across many different nodes** (routers). Each **node/router** will **read the recipient's IP address** on the packet and **forward the packet** on the least congested or **fastest route** to its destination.
When the packets arrive **at the recipient IP,** they are **reassembled in the correct order** and **any packets that don't arrive** in time or arrive **corrupted** (detected by the **Cyclical Redundancy Check or checksum**) are requested again.

The **packet number** is contained within the **packet's header** allows the **packets to be assembled** in the correct **order** on receipt.





IP packets can take different routes

Running out of IP addresses

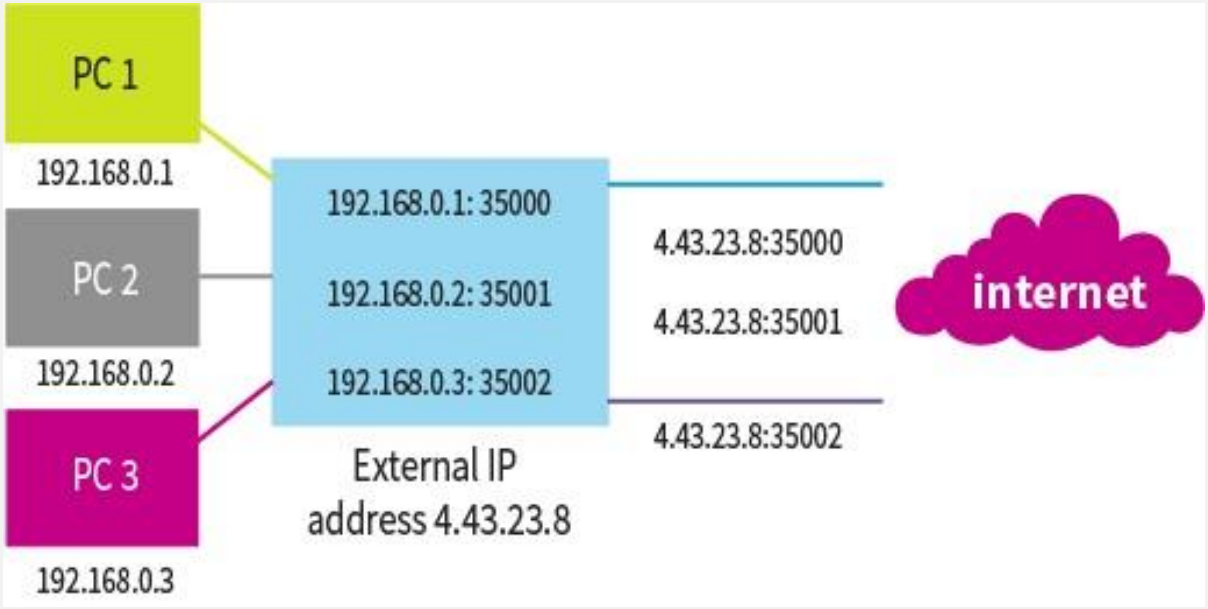
Version 4 of the internet protocol uses ~ 4 bytes (between 0 and 254). This means a single address has a maximum of 4 294 967 296 or 2^{32} unique addresses. The number of people/devices using the internet are therefore limited. Many people have multiple computers, have computers at schools and work.

Solutions are:

1) **Using Dynamic IP addresses**: Most devices do not have a fixed IP address (static IP) but instead a dynamic one. Internet service providers (ISPs) assign a dynamic IP to a device when it connects to the internet. A request is sent to a **DHCP (dynamic host configuration protocol) server**, which assigns an IP. The IP can later be recycled.
2) **Network translation (NAT):** On a local area network (LAN) internal IP addresses are used to identify network devices connected (the internal IP address range allows for more than enough devices). When connecting to the internet (WAN), these must be translated into external IP addresses. A router holds a

table of these IP addresses so that responses can be returned to the sender. This means an entire LAN like a school will share just one IP address for all connected devices.

NA**T works by changing the TCP/UDP port numbers of the source IP address.** Each device's external IP s the same with the exception of the port number on the end of the string. NAT uses the port numbers on receipt to also route the packets to their original IP on the LAN.



Modern routers use NAT

3) **Internet protocol version 6** uses **16 bytes per address rather than 4 (giving a maximum range of 2^{128}).** This allows for a larger number of devices to be connected to the internet – however, roll out is very slow since devices using version 4 don't work with version 6. This means a lot of the internet's infrastructure would need replacing.

Circuit switching Vs Packet switching

Devices that use circuit switching will send data packets along the same route for the duration of the connection. The network route path is calculated and setup in advanced. When a link fails, a new circuit has to be set up before further communications can take place.
The most common example of circuit switching today is for telephone calls. There is a direct link between devices. When a caller dials a number, various switches in telephone exchanges set up a path between the caller an the recipient. The circuit connection is set up for the entire duration including periods of inactivity (silence).
Telephone calls must use circuit switching as the packets need to arrive in order with very little latency so speech is continuous.

On the other hand, packet switching (used by the internet), means that data packets can be sent along separate routes to the recipient destination depending on the congestion/distance. With a network as large as the internet, there are so many different routes between nodes to the destination to be utilised. If one lien is broken then another may be used and the direct link does not have to be established and every link may be used by multiple connections.

Packet switching	Circuit switching
Packets can take independent routes so determine congestion and speed in real time so packets arrive as fast as possible.	All packets will arrive in order so perfect for continuous communication like telephone calls (all take the same route).
Packet switching makes connections very resilient as if one link goes down another may be used so communication is not broken.	Communication may be faster once connections are established due to lower traffic.
Bandwidth can be used more efficiently as nor part of the network is reserved and every link can be shared by multiple connections.	The connection is private and independent so less likely to have data collisions/interference – fewer corruption or packet loss. May also be more secure.
	Bandwidth can be wasted as once the connection is established, no other communications can use that route for the duration of the link. –can lead to slower services.
	A route across the network that is not already reserved needs to be set up in advanced – this can cause delay. E.g. phone engaged
Packets may not arrive in order so not useful for telephone communication. A protocol called the sliding window is used to reassemble packets in their correct order by looking at their packet number in the header.	If any part of the route is broken then a new circuit needs to be set up for communication to continue.

#**Packet switching** = a method of communicating packets of data across a network on which other similar communications are happening simultaneously.

What is a protocol?

A protocol is a **set of rules defining common methods** for **data communication**. These rules **need to be standard** across all devices in order for them **to communicate** with each other.

- **HTTP (hypertext transfer protocol)** has become the standard protocol for **web browsers to render web pages**.
- **TCP/IP** is also used worldwide and **enables communication with any other computer connected to the internet**, regardless of its location.

Handshaking is needed for communication to begin. When a communication link is set up, both devices need to agree on the set of protocols to use. No data can be sent until the handshake has occurred. It is possible that the devices don't agree on a protocol method so a link is not established.

Physical and logical considerations

When a choice of protocol has to be made, there are many aspects to consider, both **physical and logical**.
Physical aspects include:

Logical aspects	Physical aspects
Address formatting (Sender's IP, Recipient's IP, Port number)	
Time to live or hop limit	Ethernet or wireless (802.11x)
Bit rate (speed of transmission)	Fibre (optical) or copper (electrical)
Error detection (cyclical redundancy checksum)	Serial or parallel transmission (one bit of data at a time or in blocks/bytes at the same time)
Packet size	**Duplex mode** = both can communicate bidirectional but full duplex occurs simultaneously and half-duplex means they must wait (one direction at a time).
Ordering of packets	Synchronous/asynchronous (transmit and receive at same / different speeds)
Routing	Synchronous/asynchronous (transmit and receive at same / different speeds)
Encryption an digital signatures	
Compression	

The TCP/IP protocol stack

There are **many different tasks that need to be carried** out when **sending a packet over a network**. How will the packet get there? How will the packet be formatted? Does the packet need to arrive in a specific order? What happens when an error is introduced during transmission? Owing to all the work that needs to be done, it **is not practical to place all of it into a single protocol**, **so a number of protocols are chained together and known as a protocol stack**.

The Transmission Control Protocol or internet protocol (TCP/IP) protocol stack is a set of **networking protocols (rules)** that work together as **four connected layers**, **passing incoming and outgoing data packets up** and down the layers of **network communication**.

There are four layers:

The TCP/IP stack contains multiple protocols but is named after two of the protocols within it.

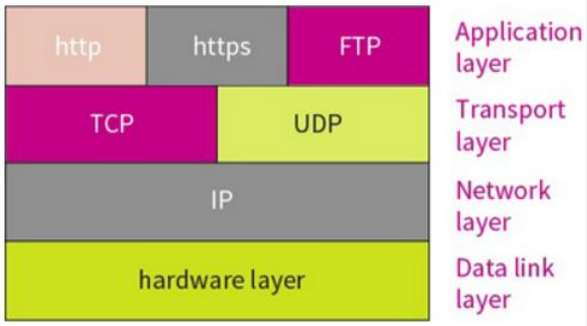


Figure 9.12 TCP/IP protocol stack.

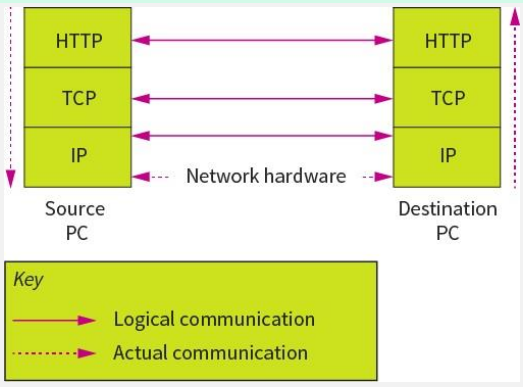
At **each layer there is a choice of which protocol to use (excluding the bottom hardware layer)**. Each choice provides different properties for the packet being sent. As the packets pass through the different layers, **different protocol methods may be used**. The packet is unwrapped, or encapsulated in an envelope containing new packet data as it descends the layers and is unwrapped again at the receiving end.

Each protocol in the stack offers different services, but always formats packets so that they can be understood by other layers.
The service access point (SAP) is where protocols will communicate with each other. This is generic so that any protocol can be swapped out **without affecting other protocols in the stack**.

This way, the application chooses the service it required, without having to tailor the data for a specific protocol. **Protocols at both ends of the network must match** for the packet to be understood (this is usually a guarantee from the handshake process).

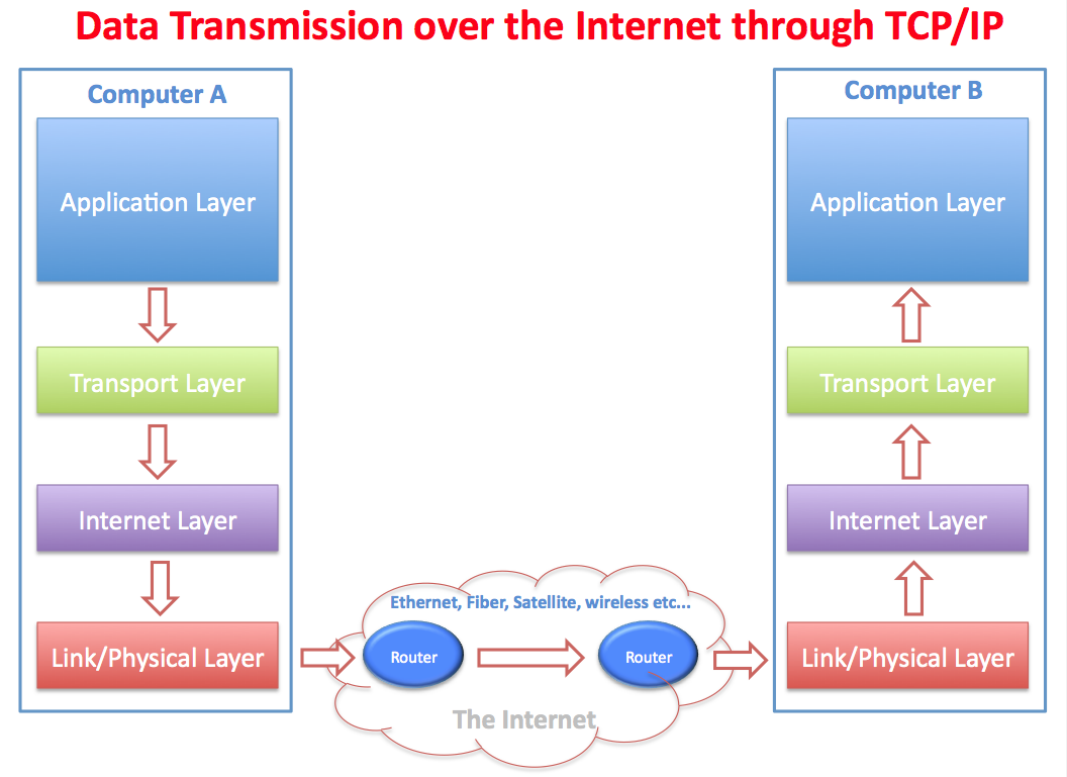
Each **protocol considers that it is communicating directly with its counterpart at the destination end along a logical communication line**. It does so by **adding data to the packet header (metadata) for its counterpart to read**. In reality, **data is added**, to the **header of a data packet**, **by each protocol** in the **layer as it passes down the stack**.

At the destination, the data packet is passed up the protocol stack with each layer removing its part of the header.



Summary: Various protocols with different roles in a stack. In each layer, a different protocol is used and the data packet is encapsulated in an envelope containing new packet data as it descends down the layers. It is unwrapped at the receiving end with different part of the header information being removed as it ascends up each layer.

The 4 layers involved in the transmission control protocol stack



The application layer

This applies the appropriate protocol to the data being transferred.

The application layer is the **first layer so sits on top of the stack** and the **protocols used are very specific** to the **applications.** It uses protocols specific to the applications in use.

e.g. HTTP, hypertext transfer protocol is used to transfer HTML web pages; while FTP, file transfer protocol, is used to send simple files over a network, an email protocol (POP) is applied to email data.

At this layer, **the transfer protocol to be used is dependent on the application that is running**. Having a choice means that **new protocols can be added w**ithout having to **change the existing structure**.

The writers of software that need their programs to have networking features may create a protocol for the application layer that decides the **structure of the data and how it will be presented.** The application layer contains the **most protocols so is the most flexible** layer in the stack.

The **protocols are not concerned with how the data will be transmitted from A to B** but what the data contains. An application can create its own protocol to be used in the application layer but **most make use of existing ones**.

HTTP (extra info) (port 80)

Hypertext transfer protocol allows **resource**s, such as images or text files, to **be transmitted** over the network. Each item is **identified using a URL**, a **uniform resource locator**, and is **transmitted over port 80.**

HTML pages, which are stored as text files, are downloaded using HTTP along with any resources identified within it. The web browser then interprets the HTML code and displays the web page.

HTTPS (extra info) (port 443)

Hypertext transfer protocol secure (HTTPS) is the secured version of normal HTTP. Before communications can start, the **data stream must be encrypted**. HTTPS works over port 443 and uses **public–private 128-bit encryption** to secure the data being transmitted.

Certificates are issued by trusted third party webservers to identify that server. Certificate includes the public key for that webserver and is used for verification purposes.

The transport layer (with TCP)

1. This layer uses the **Transmission Control Protocol (TCP**) to e**stablish the end-to-end connection** between the sender and recipient.
2. It is ONLY AT THIS POINT THAT **data is split into individual packets.** The first data to be **added to the header** is at this stage- including the **packet number and a total number of packets** or the sequence to be transmitted.
3. It also **labels the port number** which the packets will **route through**. Port number is dependent on the application protocol being used that is dependent on the application.

4. If any packets are lost or corrupted during transmission, then **the transport layer will request for these packets to be resent. Receipt of packets is also acknowledged. CHECKSUM (CRC) added**

Note needed in my spec but this layer is also needed for the network address translation (NAT) of routers.
e.g. The common port used for HTTP protocol (decided in application layer above) is port 80, it is called upon by the destination browser.

Likewise, at the recipient end of this layer, tall this data is removed and packets are reassembled in order.

The transport layer (with UDP)

Universal datagram protocol (UDP) provides an **unreliable connection channel** with **minimum level features**. A less sophisticated transport layer protocol. Commonly called fire-and-forget.

- **No error checking**
- **No ordering of packets**

This **means less data is sent** and it is a **better use of bandwidth** for applications that do not regard about the loss of data. **Much less processing** is involved too. These applications just **need data fast**.

The Network layer

This also known as the **internet layer or the IP layer**. It consists of just one protocol **called Internet Protocol**. IP uses packet switching (considering no single path as a reliable route but instead sending packets along routes through different nodes).

1. It **provides end-to-end routing paths** for network communication by adding the **source** and the **destination IP addresses** (albeit through **packet switching**)
2. It **adds the IP address of the destination** to the previously added port (by the TCP above) to form a socket.
3. Routers **operate on the network layer** and will use the destination **IP address to forward on the packets**

The addition of the **IP addresses to the port number** forms **a socket,** *e.g. 42.205.110.140:80 (added onto :80)*

This is similar to the addition of a person's name to a street address on an envelope to direct the letter to the correct person in the building. The **socket specifics which device the packet must be sent** to and the **application being used on that device.**

A socket is the endpoint of a two-way communication link. It consists of the destination IP address bounded to a port number, so will specify which device the packet must be sent to and the application being used on that device.

A socket is bound to a port number **so that the TCP layer (of receiving end) can identify the application** that data is destined **to be sent to**. i.e. so the TCP knows which Port the application requires. For the application layer above it.

The link layer

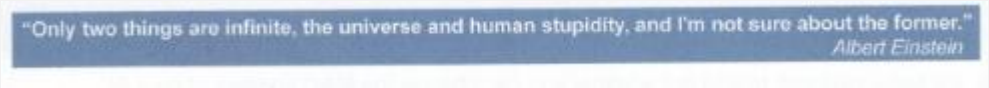
1. The **physical connection between network nodes**.
2. It adds the **unique MAC address to identify the network interface cards** of the **source and destination** hardware.

This means that once the packet has reached the correct network using the IP address, it can then locate the corresponding piece of hardware.

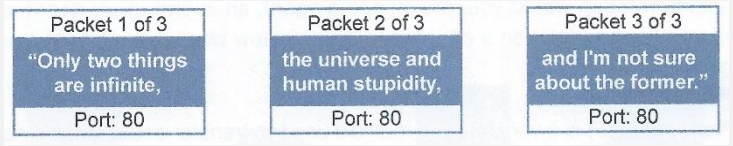
NOTE:
The destination will be the next piece of hardware the packet is to be sent to. Unless two computers are on the same network, the MAC address will change to the next router the packet is to be sent to.

==========

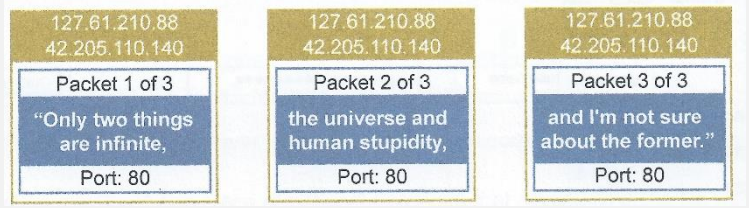
Summarising the sending process



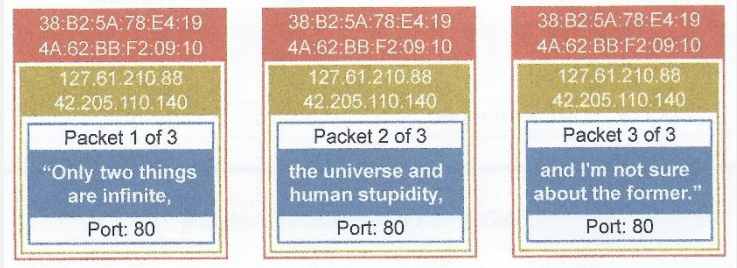
Application layer



TCP/IP layer



Network Layer



Link Layer

Summary of the recipient process

At the receiving end, the MAC address is stripped off by the link layer, which passes the packets to the network layer. The IP addresses are then removed by the network layer which passes them on to the transport layer to remove the port numbers and reassemble the packets in the correct order and check there is no data corruption or missing packets. The resulting data is passed on to the application which presents the data to the user.

Since routers operate on the network layer, source and destination MAC addresses are changed at each router node.
Packets, therefore, move up and down the lower layers in the stack as they pass through each router or switch between the client and the server.

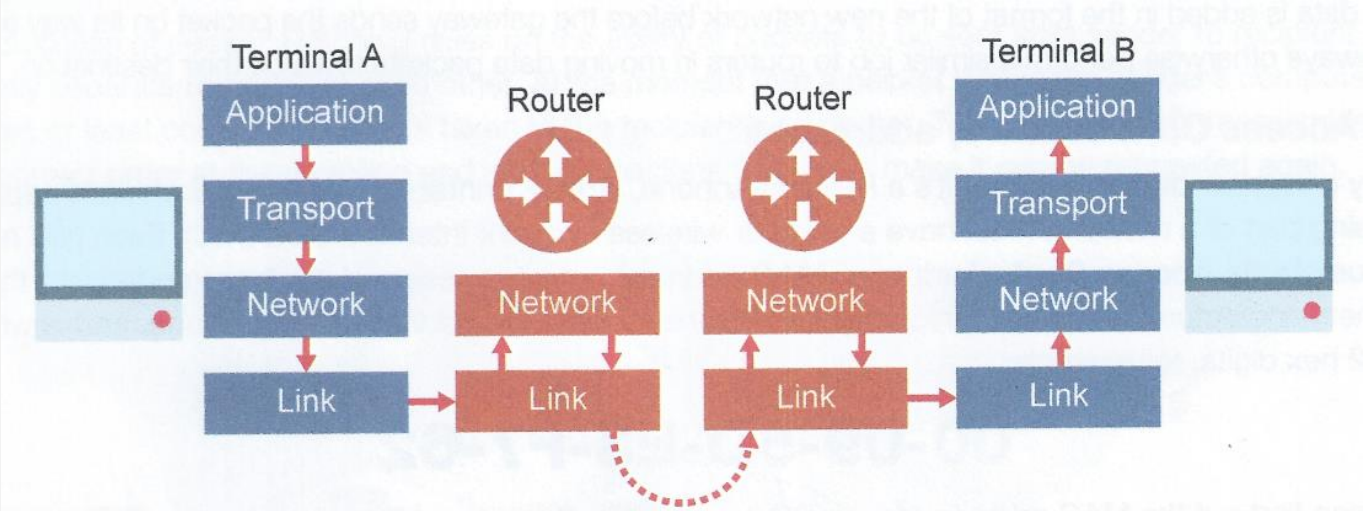


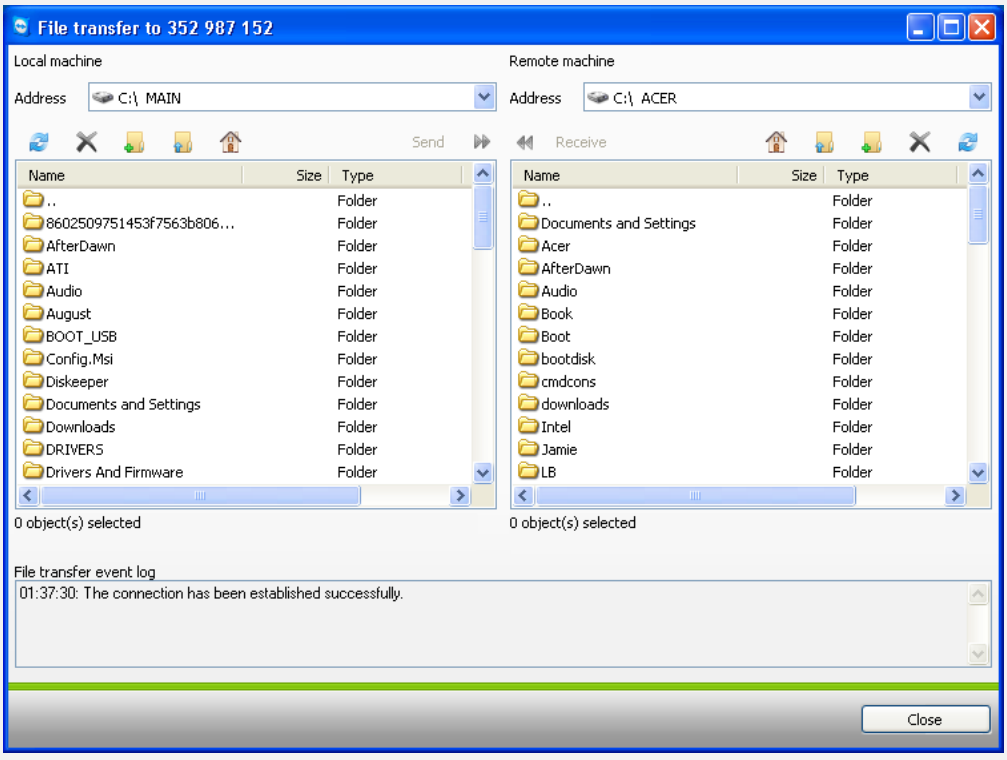
Figure 22.1 The TCP/IP protocol stack

Transferring files using FTP (port 21)

File Transfer Protocol (FTP) is a very efficient method used to transfer data across a network, often the internet. **FTP functions** as a **high-level protocol in the Application Layer** using the appropriate software.

The user is presented with a **file management screen that displays the data structure** for both the **local and remote** computer. The user can view the files and folders on both machines and **transfer data to or from by simply dragging** and dropping items.

FTP sites may also be used by companies that wish to **distribute updates to customer machines**. Most FTP sites require a **username and password to authenticate the user**, but some sites can be configured to allow for **anonymous use.**



What is a protocol – General definition

#A protocol is a set of rules and formats that govern data communication methods over networks between computers.

There are many properties of a transmission that a protocol can define. Common ones include:

1. **Packet size**
2. **Transmission speed**
3. **Packet sequence controls (numbering)**
4. **Address formatting (Sender's IP, Recipient's IP, Port number)**
5. **Time to Live or hop limit for a packet**
6. **Error detection types**

Popular protocols include: File Transfer Protocol (FTP), TCP/IP, User Datagram Protocol (UDP), Hypertext Transfer Protocol (HTTP), Post Office Protocol (POP3), Internet Message Access Protocol (IMAP), Simple Mail Transfer Protocol (SMTP).

Each protocol employed during the transmission of data adds information to the packet's header or footer. It is crucial that each computer not only foll**ows a set of rules, but ensures that both sides use the same ones.** In Computer Science, a set of rules that governs communication is known as a protocol.

In Particular, TCP/IP = The internet protocol suite:

The **internet protocol suite** consists of four layers that use methods and structures for **encapsulating** data packets with a **header, payload and trailer.**

- The header's metadata/tags are: the **sender and recipient's IP address**, the **sequence** of the packet in the transmission, the **port number** to be routed through and the **Time to Live or hop limit**.
- The trailer holds the **Cyclical redundancy checksum** to detect corruption of data in transmission
- The **protocol being used**
- The protocol also determines the **transmission speed**

The role of a mail server in sending and retrieving email

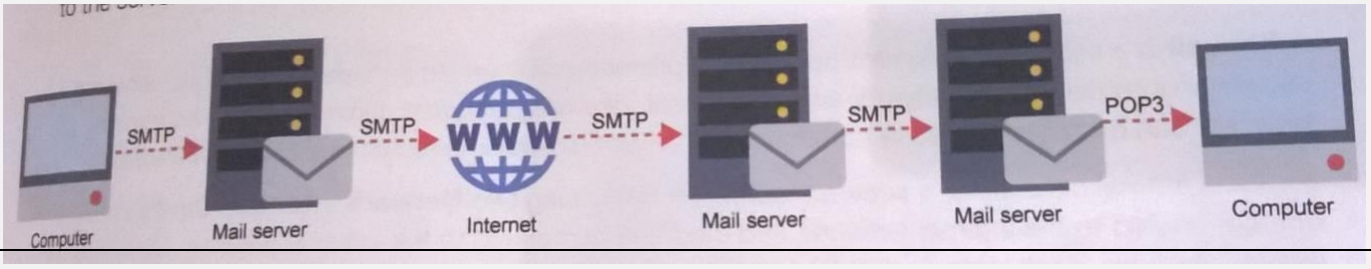
A mail server acts as a virtual post office for all incoming and outgoing emails. These servers route mail according to its database of local network user's email addresses as it comes and goes, and store it until it can be retrieved.

Post Office Protocol (v3) (POP3) is responsible for retrieving emails from a mail server that temporarily stores your incoming email. When emails are retrieved from the mail server by POP3, they are transferred to your local computer (desktop/phone/tablet/laptop), and deleted from the mail server.

This means that applications that allow users to access email through POP3, find that their emails don't synchronise the same emails on all devices.

Internet Message Access Protocol (IMAP) is another email protocol that is more commonly used today. It is designed to keep emails stored on the server, thus maintaining synchronicity between devices.

Simple Mail Transfer Protocol (SMTP) = is used to transfer outgoing mail from one server to another or from an email client to the server when sending an email.



The role protocol methods

We need protocols so that each computer knows exactly how data should be sent over the network, and when it arrives the computer will know exactly how to process it.

- **TCP/IP = the basic communication language or protocol of the Internet. This governs the method that the data is sent over the internet and is applied in the transport layer (either uses TCP or UDP).**
- **TCP = provides a steam like communication with error detection, ordering of packets and confirmation that packets have been received. Also adds port number to identify application being used.**
- **Hypertext Transfer Protocol (HTTP) = Hypertext Transfer Protocol, HTTP, is a high-level protocol applied in the application layer that is used by the World Wide Web (WWW)/assigned by browsers. This protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.**
- **Hypertext Transfer Protocol (HTTPS) Hypertext Transfer Protocol secure (using end to end encryption. This communicates over port 443. Certificates and the secure sockets layer (SSL) protocol are used to secure communication.** Encryption = **public-private 128-bit encryption to secure the data being transmitted.**
- **File Transfer Protocol (FTP) = Uses port 21, a method of communication that allows files to be transmitted.**
- **User Datagram Protocol (UDP) = a fire-and-forget protocol with minimal protections (no error checking/no ordering of packets or sending of lost packets). Applied in transport layer for data that needs to be sent fast without reliability.**
- **Post Office Protocol (POP3) = a protocol used for retrieving email from a mail server that temporarily stores incoming mail.**
- **Internet Message Access Protocol (IMAP) = Internet Message Access Protocol, used for retrieving incoming email from a mail server and is also designed to keep emails on a server to ensure synchronicity between devices.**
- **Simple Mail Transfer Protocol (SMTP) = a protocol used to transfer outgoing emails from one mail server to another (or form a client to a mail server) when sending emails.**