How does the internet work?

Introduction

The internet is a global wide area network, that connects networks throughout the world, allowing any connected device to communicate with any other connected device.

Many protocols, processes and administrative organisations combine to facilitate this global communication network, including:

- The Internet Protocol (IP, including IP addresses)
- Packet switching and routing
- The Domain Name Service (DNS)
- Universal resource identifiers (URI) / locators (URLs)

Before going any further, read this explanation of how the Internet works to provide a helpful overview of this complex subject.

Read: How does the Internet work? - Learn web development | MDN (mozilla.org)

How is the internet structured?

Specification point: *Understand the structure of the Internet.*

The internet is a world-wide network of networks, connected via routers that facilitate the transfer of data from any connected device to any other. These routers are arranged within a hierarchical organisation, with packets passed up the hierarchy and transferred across networks before descending back down the hierarchy to their destination.

At the top of this hierarchy is the “internet backbone” which provides the core that all other network and internet service providers connect to.

The majority of the internet backbone’s physical infrastructure is comprised of underground and under-sea fibreoptic cables, owned and maintained by large telecommunications companies.

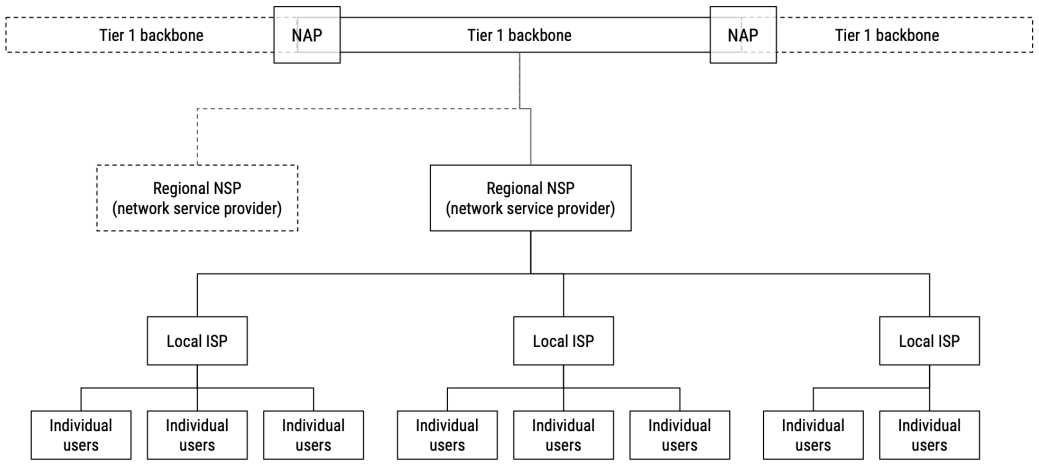

Figure 1 - The hierarchical structure of the Internet

Tier 1 networks

Each of the interconnected networks that form the internet operate at different “tiers”.

Tier 1 networks form the internet’s backbone. These are very large networks, capable of handling huge amounts of bandwidth and are operated by large telecommunications companies, known as Network Service Providers (NSPs).

Tier 1 NSPs own most undersea fibreoptic cables as well as much subterranean fibre cable within their home nations (and often in other countries too).

Tier 1 NSPs can reach every other network on the internet without having to pay a fee. This doesn’t mean that they are directly connected to every other network but rather that each Tier 1 network can exchange traffic with other Tier 1 network without charge, in an arrangement known as *peering*. This mutually beneficial arrangement ensures efficient transfer of traffic for all of a Tier 1 network’s customers, even if it involves opening up its network to its competitors.

There are a surprisingly small number of Tier 1 NSPs, mostly based in the US. Only one is operated by a UK company, Liberty Global (who own Virgin Media). AT&T, Verizon and T-Mobile US are all American Tier 1 NSPs whom you may have heard of.

Activity: Explore the undersea tables that are owned and operated by Tier 1 (and some Tier 2) organization.

Activity: Look at this list of Tier 1 networks

Tier 2 and 3 networks

A Tier 2 network engages in peering with other Tier 2 networks but also pays for access to Tier 1 NSPs to transfer data to some parts of the internet[1]. Tier 2 NPSs often provide national network services, though possibly not international. Tier 3 networks are operated by smaller ISPs who pay Tier 2 networks to transfer data outside of their own network. In both Tier 2 and Tier 3 networks, the charges involved in passing data to an NSP further up the hierarchy are then passed onto individuals and businesses who subscribe to their services.

Most consumer and business-focused ISPs are Tier 2 NSPs, including BT and Vodafone or Tier 3 networks.

As data is passed between different nations, packets will be passed up to a national router and then transferred internationally along a Tier 1 or 2 network and then passed back down the routing hierarchy to its destination.

Internet Exchanges Points (IXs)

Tier 1, 2 and 3 networks interconnect at physical locations around the world known as Internet Exchange Points (IXs). Internet Exchanges are often established as a collaboration between ISPs who wish to mutually benefit from being able to access each other’s networks quickly and cheaply and, in doing so, provide better service to their customers. LINX (London Internet Exchange) was the first IX in the UK, forming in 1994 as a collaboration between the leading ISPs in the UK at the time, namely Pipex, BT, Demon Internet, EUnet GB and JANET (who run the academic network that provides UK universities with internet connectivity)[2]. Now, LINX operates a number of IXs around the UK into which over 950 different networks (Tier 1, 2 or 3) connect. Any of the connecting networks can then make direct connections to any other, providing an efficient transfer of data between networks.

---

[1] Tier 2 network - Wikipedia

[2] [ARCHIVED CONTENT] Objects (nationalarchives.gov.uk)

Figure 2 - The switch used at the first London Internet Exchange

Watch: An overview of the Internet and IXs (The Euro-IX video): https://youtu.be/yJJHukw9Lyc

Summary: The structure of the Internet

- Individuals (consumers and businesses) connect to local Internet Service Providers (ISPs, operating Tier-2 or Tier-3 networks), such as BT, TalkTalk, Sky, etc.
- These local ISPs themselves then connect to each other and to larger regional ISPs (mostly Tier-2 networks), from whom they might pay for bandwidth, at Internet Exchanges (IXs).
- IXs are physical buildings dotted around the country where ISP's networks can connect to one another and to larger, regional ISPs if the traffic needs to pass to another network that is not connected to that IX directly. From here, regional ISPs will connect to each other and larger national network service providers (NSPs) or else directly to the internet backbone[3].

Summary: Internet Architecture - Client, Internet Service Provider, Regional ISP and Backbone (generalnote.com)

[3] Internet backbone - Wikipedia

How is data sent across the internet?

Specification points:

- *Understand the role of packet switching and routers.*
- *Know the main components of a packet.*
- *Define:*
  - *Router*
  - *Gateway*
- *Consider where any why they are used.*
- *Explain how routing is achieved across the internet*

A typical local area network (LAN) will use Ethernet to connect devices and exchange data. Data is encapsulated in Ethernet *frames* which are sent between devices. Each frame contains the MAC address of the destination, and this used by the destination device to identify that the packet is intended for them.

This works well for devices on the same LAN, but what about devices across two different networks?

Let's imagine we have two networks, A and B, which are connected via some specialist devices or hosts that perform the role of "gateways" to the networks. This connection is possibly managed by a different link-layer protocol such as PPP (Point-to-Point Protocol), rather than the Ethernet used within the LAN.

Now imagine that host X on Network A (let's call is A.X) wanted to send data to device Y on Network B (B.Y). Using MAC addresses wouldn't work as, whilst A.X could be told the MAC address of B.Y, there would be no device on Network A that has that destination MAC address and thus the frame would be dropped – no host within network A would listen to and respond to it.

To solve this problem, another address needs to be added to the data being sent. Rather than determining which piece of network hardware should respond to the frame, this address is used to determine its ultimate destination, quite possible within another network. This inter-network address is known as an IP address and is fundamental to the operation of the Internet.

Watch: The Internet Protocol (Ben Eater): https://youtu.be/rPoalUa4m8E (9:33)

The structure of IP packets

Specification point: *Know the main components of a packet.*

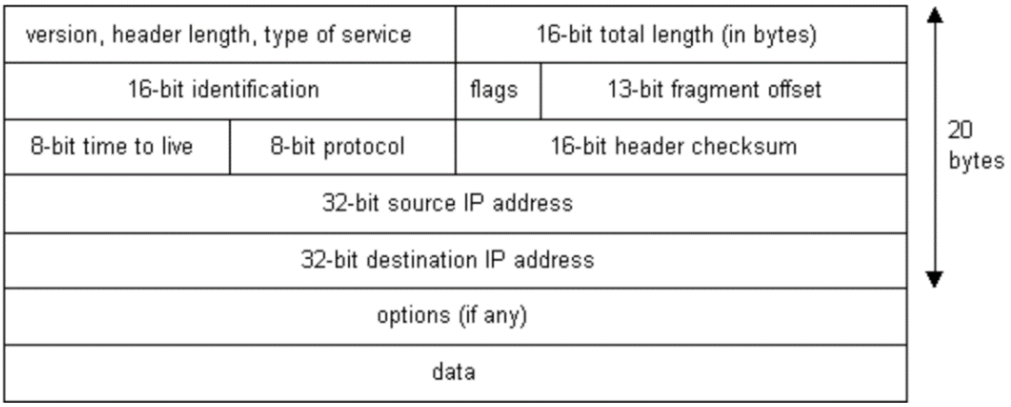

Figure 3 - The structure of an IPv4 packet[4]

[4] https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper_files/ruswp_diag7.gif

Internet Protocol packets carry TCP and UDP packets to their destination. Each IP packet begins with a header that contains, amongst other things, the IP addresses of the packet's source and destination hosts, a "time to live" (TTL) value and a checksum.

The source and destination IP addresses are used by routers to determine which path the packet should be delivered along. They are also used by the ARP protocol to deliver the packet to the correct host once within the destination LAN.

The TTL value is used to ensure that packets do not get caught in a never-ending cyclical route as they travel from router to router along their journey. When the IP packet is created, the TTL value is set to some positive value (different operating systems have different defaults; 64 is typical for macOS, 128 for Windows XP/10 and Server[5]). As the packet is received by a router, its TTL value is decremented before it is passed on. If a router receives a packet with a TTL value of 1 is dropped and the packet is discarded.

The IP header checksum is used to ensure that a packet has not been corrupted upon receipt. If it has, it is discarded by the receiving router[6].

Following the header, the original TCP/UDP packet is appended, forming the total IP packet.

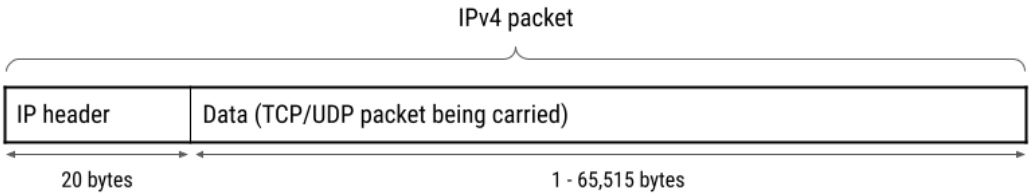

Figure 4 - A more generalised diagram of an IPv4 packet

IP addresses

Watch: Computerphile: I.P. Addresses https://youtu.be/L6bDA5FK6gs

Every device on the internet has a unique IP address. A device's IP address is used to transfer data (in the form of *IP packets*) from one device to another across the internet via devices that help direct a packet across the network known as *routers*.

IP addresses are numbers (a bit like a phone number for a computer) and come in two forms; IPv4 and IPv6. [7]Their role and function are essentially the same but they differ in their length, i.e., the number of bits that they use to represent their unique number.

IPv4 uses 32-bit numbers and are expressed (for humans to read) as four decimal numbers, separated by full stops. Each decimal number is between 0 and 255 and is represented in binary as 8 bits. For example:

IPv4 address (decimal): `172.144.43.2`

Address in binary: `10101100 10010000 00101011 00000010` (32-bits total length)

IPv6 uses 128-bit numbers, represented as eight groups of up to four hexadecimal digits, separated by colons. Each hexadecimal digit represents four bits, so 8*4*4 = 128 bits in total length.

IPv6 address: `2001:db8:3333:4444:5555:6666:7777:888`

[5] https://packetpushers.net/ip-time-to-live-and-hop-limit-basics/
[6] https://en.wikipedia.org/wiki/Internet_checksum
[7] IPv4 and IPv6 address formats - IBM Documentation

The representation of IPv6 addresses can be made more compact by omitting any zeros at the start of a segment (value between commas) begins with or comprises only 0s. For example, `2001:0db8:0001:0000:0000:0ab9:C0A8:0102` can be expressed as `2001:db8:1::ab9:C0A8:102.`

What is the significance of all this? By utilizing more bits to represent each IPv6 address, there are many more unique addresses available for each device. Indeed, there are (in the order of[8]) 2^{128} unique addresses versus 2^{32} unique addresses available within IPv4.

Extra detail: Mapping between IP and Ethernet (Ben Eater) (10 minutes)

Routing IP packets

Watch: Routing video: https://youtu.be/gQtgtKtvRdo (7 minutes)

Let's go back to Networks A and B with their devices A.X and B.Y. For X.A to communicate with B.Y, it needs the destination IP address of B.X. This will tell A.X where it can find B.Y across the internet.

When A.X prepares to send its data to B.Y it will determine from the IP address of B.Y that B.Y is not on the same network (or sub-network) as A.X. It will do this using a process known as *subnet masking*, which will be explained later.

Having determined that B.Y is external to A.X's local network, A.X will encapsulate its IP packet (destined for B.Y) with an Ethernet frame destined for A.X's local *router*[9]. The router will then look at the IP packet that it has received and determine the best path to send it along to help it reach its destination. To enable this, the router might have multiple connections to other routers, each connected one of many *interfaces*.

When a router receives a packet, it must determine:

- Should it drop the packet (TTL = 1)?
- Should it forward the packet on?
- If it should forward the packet, where should it send it?

The router uses information stored in its *routing* and *forwarding tables* to answer these questions, along with routing protocols such as Link State Protocol, Border Gateway Protocol (BGP), Distance Vector and shortest path algorithms such as Dijkstra's.

The router will store information about which routers it is connected to and the shortest path for certain destinations within its *routing table*. It will use a *forwarding table* to ensure packets are sent out on the appropriate interface to reach the best router for handling the packet. All of this is based on the destination IP address contained within the IP packet's header.

Of course, not every router on the Internet can know about the location of every other host and router - with IPv4 there would be over 4 billion destinations to store. To overcome this, IP addresses are organised into ranges and routers store information about the best path to reach each network. When a router receives a packet, it uses *longest prefix matching* to determine closest matching route to send a packet along towards its ultimate destination.

---

[8] *In the order of*: In reality, some IPv6 address ranges (and IPv4 for that matter) are reserved, reducing the actual number of possible addresses from the theoretical maximum permitted by the number of bits used.

[9] When connecting a LAN to an ISP, it's more likely that the packet is transferred from the sending device to a gateway than a router. A gateway is a device that acts an interface between two physical-layer networks, for example connecting an Ethernet network to a PPP network. This is why most operating systems will refer to the "default gateway" within their network settings, rather than "default router".

For example, an IP packet destined for 142.250.179.228 could be sent along any of three potential routes based on the matching prefixes in the table below, however the route with the longest matching prefix (142.250.179.0/24) will ultimately be selected and the packet will be forwarded on the router's outbound interface #1. Prior to passing on the router, the TTL value within the IP packet is decremented.

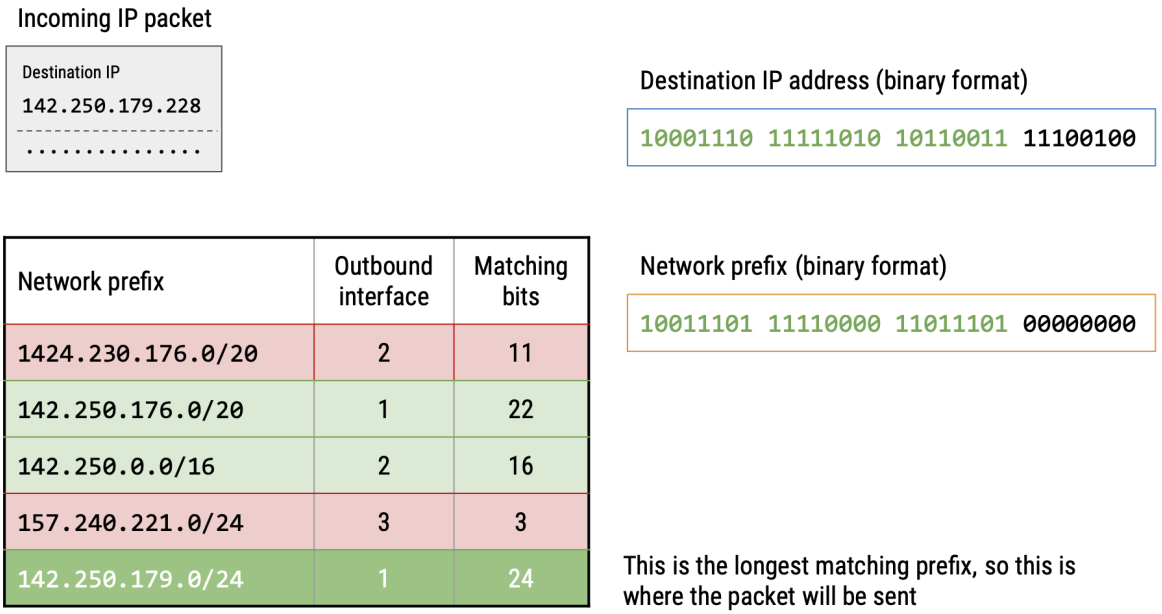

Figure 5 - Determining which interface to route a packet along using longest prefix matching

This process is repeated as the packet continues its journey to its destination, moving from router to router in "hops". Each router along the way determines the best path to send the packet along, meaning its path is not (cannot be) determined at the start of its journey. As the packet is passed through each router its TTL value is decremented and the MAC address within the encapsulating link-layer (Ethernet) frame is updated to direct the packet to its next router.

Eventually, the packet will make its way to the router of the destination network, which will determine that the packet is for a local device and send it on to that device within an Ethernet frame containing the destination device's MAC address.

An analogy for all of this is the process of sending a letter from one town to another. The destination address of the letter remains constant throughout transmission, however the identification of the "carriers" along the way (postal van, machines within a sorting office, HGV, more machines in another sorting office, another postal van at the destination town) will all differ and change at each stage. In this analogy, the postal address is equivalent to an IP address, whilst the names of the postal worker, the vehicle registration of the vans, etc., are akin to the MAC addresses and allow the letter to get to its next immediate "hop" in the chain.

Watch (extra): Computerphile video Routers, the Internet and YouTube offline (13 minutes) for additional revision

Read more: https://www.open.edu/openlearncreate/mod/oucontent/view.php?id=129631&section=12 and https://www.open.edu/openlearncreate/mod/oucontent/view.php?id=129631&section=13

Packet switching

Packet switching is the process that controls how **data packets are sent across the internet using different paths in an efficient and fault-tolerant way**, ensuring a reliable transmission of data from one host to another.

As the packets that represent some given data travel along their journey to their destination they may take different paths along the network of interconnected routers, arriving at different times and most likely in different orders at their destination. At each "hop" in their journey the router they arrive at will determine which path to send them along based on information such as network congestion and latency between routers. Algorithms such as Dijkstra's shortest path and Distance Vector Routing[10] are used to determine the best path to take.

This process of sending each packet along different paths is known as packet switching and results in very efficient use of the network as no paths between routers are tied up for long periods of time whilst large amounts of data get transferred. It is also means that packets can be routed around any unavailable/congested routers and avoid any points of failure within the packet switching network. Indeed, this was fundamental to the internet's original design, to help ensure a robust network that could withstand nodes within the network being destroyed as a result of a nuclear attack during the Cold War[11].

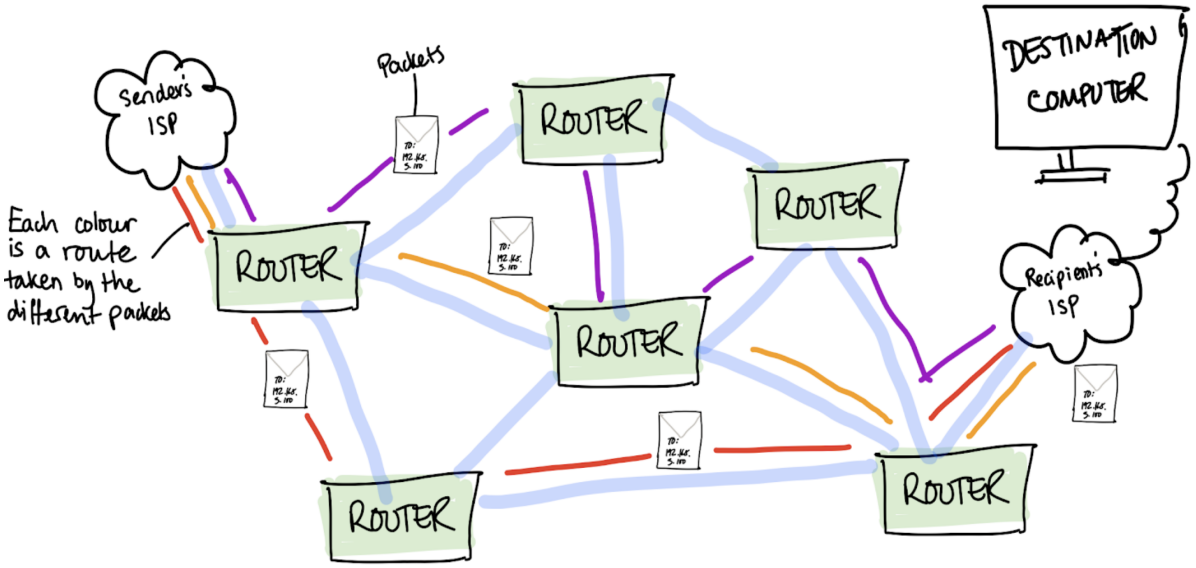

Figure 6 - An overview of the Packet Switching process

[10] https://www.geeksforgeeks.org/distance-vector-routing-dvr-protocol/
[11] https://www.rand.org/about/history/baran.html

Principles of operation of a packet switching network

The process of sending data across the internet via packet switching can be summarised as:

1. The data to be sent is broken down into smaller packets by a transport layer protocol, either TCP or UDP.
2. If using TCP, each packet is given a sequence number and checksum based on the packet's contents.
3. The TCP/UDP packet is encapsulated within and IP packet and the source and destination IP addresses are added.
4. Each packet is sent along different paths towards its destination. This path is determined as the packets travel from router to router, rather than being determined in advance.
5. As packets arrive at their destination they are unencapsulated from their IP packet and, if a TCP packet, their checksum is used to determine that the packet has been sent correctly and an acknowledgement packet is sent to the sender if so.
6. The original data is then reassembled from the TCP received packets, using the sequence number to determine the correct order.
    a. If a UDP packet is received it is sent to the appropriate application based on the destination port as there is no mechanism within UDP for reassembling packets based on order or checking for errors during transmission.

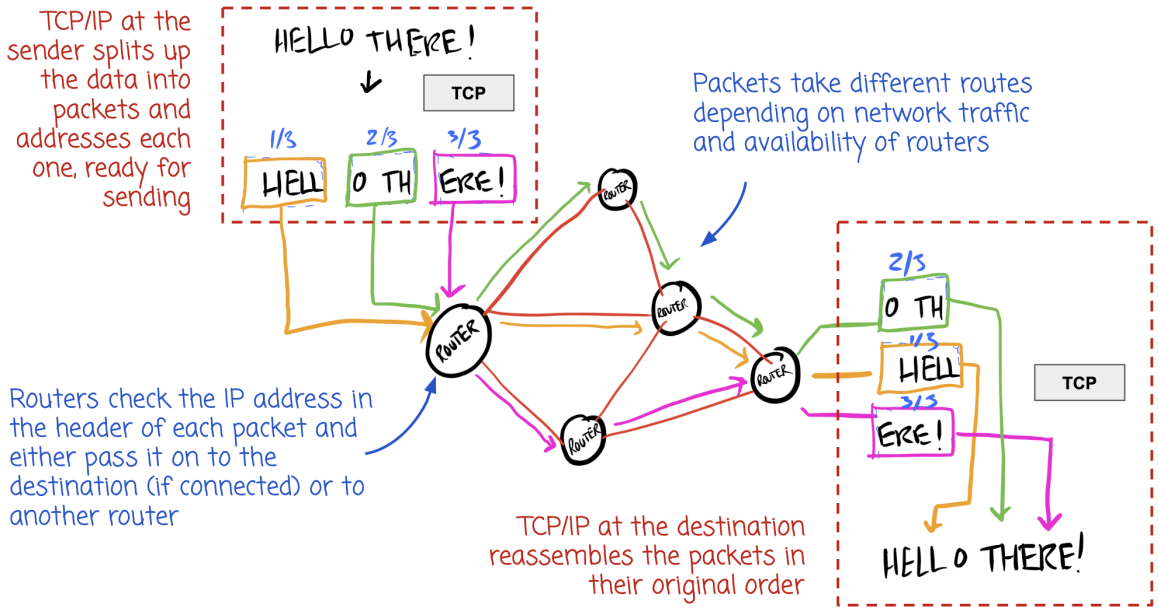

Figure 7 - The steps involved in sending a TCP packet via a packet switching network

The routing hierarchy

As data is passed between different countries, packets will be passed up to a national router and then transferred internationally along a Tier 1 or 2 network and then passed back down the routing hierarchy to its destination.

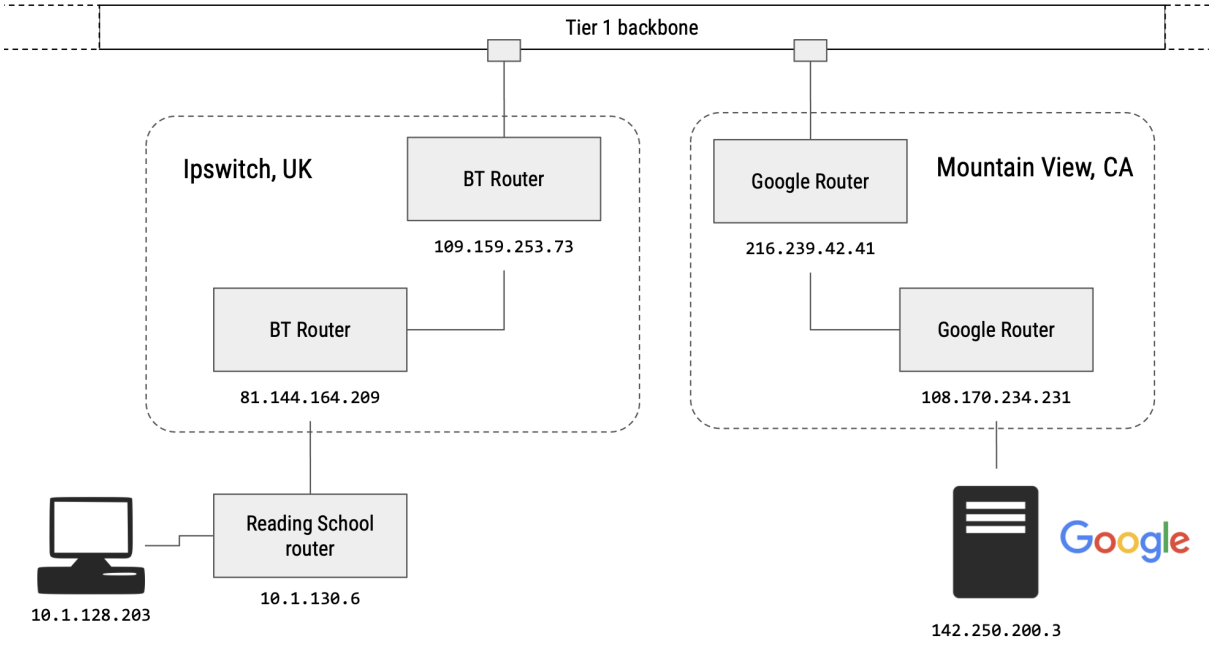

Figure 8 - The hierarchical organisation of routers

Below is the path taken between a Reading School desktop computer and one of Google's webservers, as shown using the *traceroute* program (`tracert` on Windows).

```
C:\Windows\system32>tracert www.google.co.uk

Tracing route to www.google.co.uk [142.250.200.3]
over a maximum of 30 hops:

  1     1 ms     1 ms     2 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2    <1 ms    <1 ms    <1 ms  81.144.164.209
  3     3 ms     3 ms     3 ms  86.189.33.32
  4     4 ms     4 ms     4 ms  core1-te0-11-0-12.colindale.ukcore.bt.net [109.159.254.2]
  5    28 ms     4 ms     5 ms  peer3-et3-0-1.slough.ukcore.bt.net [109.159.252.80]
  6     4 ms     4 ms     4 ms  109.159.253.73
  7     6 ms     6 ms     6 ms  216.239.42.41
  8     4 ms     4 ms     4 ms  108.170.234.231
  9    24 ms    13 ms    15 ms  lhr48s29-in-f3.1e100.net [142.250.200.3]
```

Figure 9 - Output of a traceroute to one of Google's web servers

Activity: Use traceroute ('`tracert`' on Windows) and IP geolocation to determine the path a packet takes from school to a remote destination.

Routers and gateways

Routers are devices that transfer IP packets from one network to another. Each router has multiple network interfaces and can pass packets onto routers connected to these interfaces.

Gateways repackage IP packets to use a different link-layer protocol are required where packets need to be sent between networks that use different link-layer topologies, for example moving from an Ethernet network to a PPP (Point to Point Protocol) link to an ISP over the telephone network.

Despite referring to the device that connect most homes to the internet as “routers” the reality is that most homes have a gateway that passes all outbound packets to their ISP, from which they will begin their journey through routers across the internet. Home “routers” do not have multiple incoming or outbound interfaces through which they can connect to multiple networks, but a single “WAN” port to forward any packets not destined for a local device.

Read: Gateway and router: what is the difference and similarity | FS Community

Summary: Sending data across the internet

- Routers are arranged in a *hierarchical organisation*, with packets destined for another network passed “up” the hierarchy until they are transferred to their destination network where they then descend the hierarchy to their destination host.
- When a router receives a packet, it looks at its destination IP address of the incoming packet (within the header).
- The router will have already determined the best route to reach of its known network prefixes using a shortest path algorithm.
- The router then compares the destination IP address against the prefixes (first *n* bits) of each network that it knows routes to. The router picks the route that has the longest matching prefix to the destination IP address of the packet.
- To do this, the router consults its forwarding table to determine which interface to send the packet out from once it has determined which router to pass it to.
- The process of transferring data as distinct packets through multiple connected routers and along different paths is known as *packet switching.* Packet switching ensures a reliable and efficient transmission of data but will likely result in packets arriving at their destination at different times and in a different order.
- Gateways are devices that can transfer packets across networks that use different link-layer protocols, for example connecting Ethernet networks to an ISP via a phone line.

How are resources located on the internet?

Specification points:

- *Describe the term 'uniform resource locator' (URL) in the context of internetworking.*
- *Explain the terms 'fully qualified domain name' (FQDN), 'domain name' and 'IP address'.*
- *Describe how domain names are organised.*
- *Understand the purpose and function of the domain service and its reliance on the Domain Name Server (DNS) system.*
- *Explain the service provided by Internet registries and why they are needed.*

As we've repeatedly seen, every host on the internet, including servers, has a unique (public) IP address. This address is essential for routing data (including requests and response) to its destination.

This system was fine in the early days of the Internet, however IP addresses are not particularly friendly and easy to remember and, as the number of hosts on the internet grew, the need arose for an easier way for humans to refer to specific hosts on the network. Such a system would allow a user to access a webserver with a friendly and easy-to-remember **fully-qualified domain name** such as www.google.com rather than the much less memorable IP address 142.250.200.36.

Domain names and Fully Qualified Domain Names (FQDNs)

Read: About fully qualified domain names (FQDNs) (iu.edu)

A domain name (such as google.com) is a human-readable identifier for a realm of administrative authority or control within the internet[12]. Individuals and organisation can register a domain name, meaning that they will then be able to manage it, doing things such as creating entries within the DNS system (DNS records) that point certain hosts owned by the organisation to a paticular IP address, such as *www.google.com* being pointed to 142.05.200.36 and *mail.google.com* being pointed to 142.250.200.5. In these cases, both 'www' and 'mail' are particular hosts (or services) within the 'google.com' domain and each points to a different IP address.

A **fully-qualified domain name (FQDN)** provides the **exact location of a particular host within the Domain Name System (DNS) hierarchy**. More on the DNS and its hierarchical organisation later.

A computer's FQDN is comprised of its **machine name** and **the names of all the domains to which it belongs**, including its **top-level domain**, separated by full-stops.

For example:

> www.google.com

Technically, the FQDN should also include a final "." for the root domain although as this is required in every FQDN, it is almost always ignored. Your computer adds it though and you can too. Try this link:

> https://en.wikipedia.org./wiki/Fully_qualified_domain_name
>
> (notice the extra '.' after 'org')

[12] https://en.wikipedia.org/wiki/Domain_name

Why do we use FDQNs?

- Fully Qualified Domain Names are **easier to remember** than IP addresses.
- FQDNs are **easier for humans to understand** and identify the service they relate to. For example, www.google.com is clearly a webserver on Google's network. It is less clear that 142.250.200.36 is the IP address of the same server.

Read more (optional): FQDN – Fully Qualified Domain Name Explained for Beginners (hostinger.co.uk)

Translating FQDNs into IP addresses

But how does a computer translate the URL http://www.google.com/ into http://142.250.200.36/ ?

This job is the responsibility of DNS, or the **Domain Name System**.

The purpose and function of the Domain Name System

Read: What is DNS? | How DNS works | Cloudflare

The Domain Name System (DNS) is a distributed database used to resolve fully qualified domain names into IP addresses.

DNS is often referred to as "the Internet's phonebook" and with good reason, for it performs a very similar function. Whenever you wish to call a friend, it is highly likely that you look their name up within your Contacts app, tap on their name and your phone calls them. Notice, however, that when your phone initiates the call it retrieves the associated phone number from its contacts database and uses this number to establish the connection with your friend's phone. Your computer does almost exactly the same thing whenever you ask it to access a host on the internet by name rather than IP address.

The DNS process

Watch: PowerCert explanation of how DNS works: https://youtu.be/mpQZVYPuDGU

When an attempt is made to access a host via its FQDN, this must be resolved to an IP address before the host can actually be accessed.

For example, when a web browser is told to access a URL such as https://www.bbc.co.uk/ it must first determine the IP address of www.bbc.co.uk (212.58.237.251) before it can then send the HTTP request to the webserver via TCP and IP.

To find this IP address, the following process if performed:

1. The computer checks its **local DNS cache** to see if a record for the required FQDN can be found. If so, the associated IP address is retrieved and used to send the request and the process finishes.
    - The local DNS cache a database of previous lookups and their results that is maintained by the OS and/or web browser and stored on the local computer.
2. If a local record does not exist, the OS will send a DNS lookup request to its assigned DNS *resolver* (specified within the OS's network settings).
3. If the assigned DNS resolver has a matching record in its cache it will respond with the required IP address and the process finishes.
4. If the computer's DNS resolver does not have a matching record, it will send a query to one of 13 **root** nameservers[13], which will then respond with the IP address of the Top-Level Domain (TLD)

[13] Actually, there are more than 13 root servers. The 13 IP addresses for the root servers point to edge-servers that can then distribute the request to any of a number of root servers, owned and managed by 12 organisations and physically spread across the world (see https://root-servers.org and https://www.cloudflare.com/en-gb/learning/dns/glossary/dns-root-server/)

nameserver that is responsible for the appropriate TLD specified within the FQDN being looked up, for example one of the nameservers for the .com TLD.
    - These 13 root severs have fixed IP addresses, which every DNS resolver has hard-coded into their software, so they will always be able to reach them to send a query.
5. The computer's DNS resolver will then use this information to send a DNS request to the appropriate TLD nameserver, which will respond with the IP address of the nameserver for the domain specified within the FQDN.
6. Finally, the DNS resolver will send the request to the nameserver registered to the domain specified within the FQDN. This nameserver is the **authoritative nameserver** for the domain and knows the IP addresses of each of its hosts. This will return the IP address of the host specified within the FQDN part of the original URL and the process finishes.

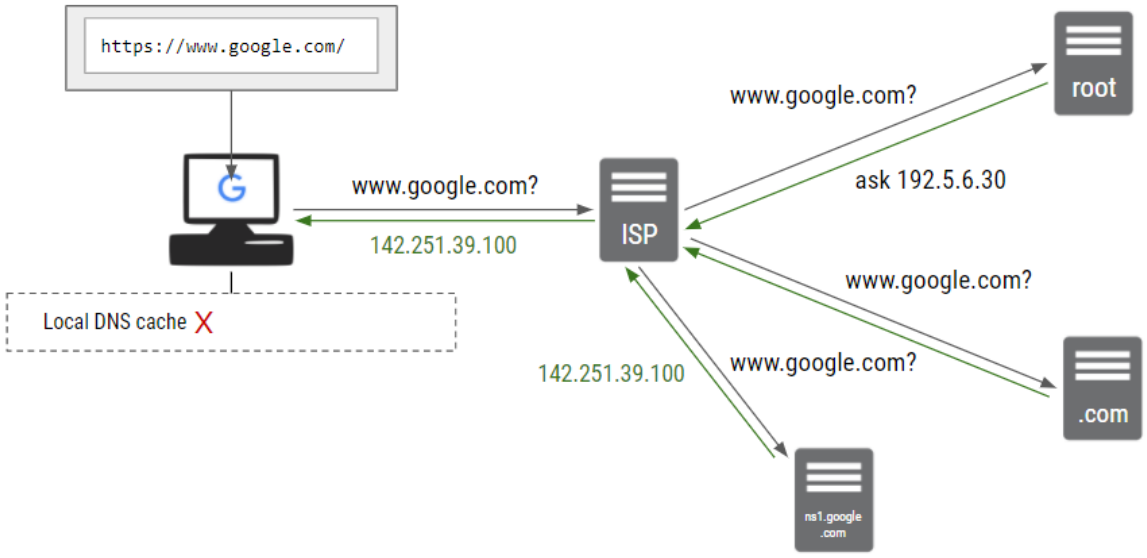

Figure 10 - An overview of the DNS lookup process

Note that DNS does not translate URLs into IP addresses, but just the FQDN part of a URL, in order to identify the IP of the required host. The protocol and path elements of the URL are still required to access the specific resource.

Going deeper: This video from Ben Eater includes a detailed walk-through of performing a DNS query: https://www.youtube.com/watch?v=-wMU8vmfaYo

Activity: Use 'dig' to query a root name server for the IP of an FQDN of your choice (e.g. www.google.co.uk), following the steps in the Ben Eater video linked above. Continue to query each name server that you are pointed to until you obtain an authoritative answer. To get started type the following:

```
dig @198.41.0.4 www.google.co.uk
```

When is DNS not required?

There are occasions when a host can be reached without requiring a prior DNS resolution. These include:

- When the local computer has a matching record in its local *hosts* file
- When the local computer already has a matching DNS record in its local cache.
- When the URL being looked up already contains an IP address.
- When the URL is for a local resource, i.e. a file on the computer's storage.
- When the URL refers to *localhost* – this will always point to 127.0.0.1 and, as such, no DNS query is required to resolve this.

How domain names are organised

Read: https://www.cloudflare.com/en-gb/learning/dns/glossary/dns-root-server/

Domains within the DNS system are organised into "zones" within a **hierarchical tree structure**, with the "**root zone**" at the top, immediately followed by **top-level domains**[14] (TLDs), such as a .com, .uk, .org.

Each DNS zone is a distinct administrative area, managed and maintained by a specific organisation or administrator (user)[15]. Within a DNS zone will be authoritative nameservers for that zone, which provide the IP addresses required to resolve (or continue) a DNS lookup.

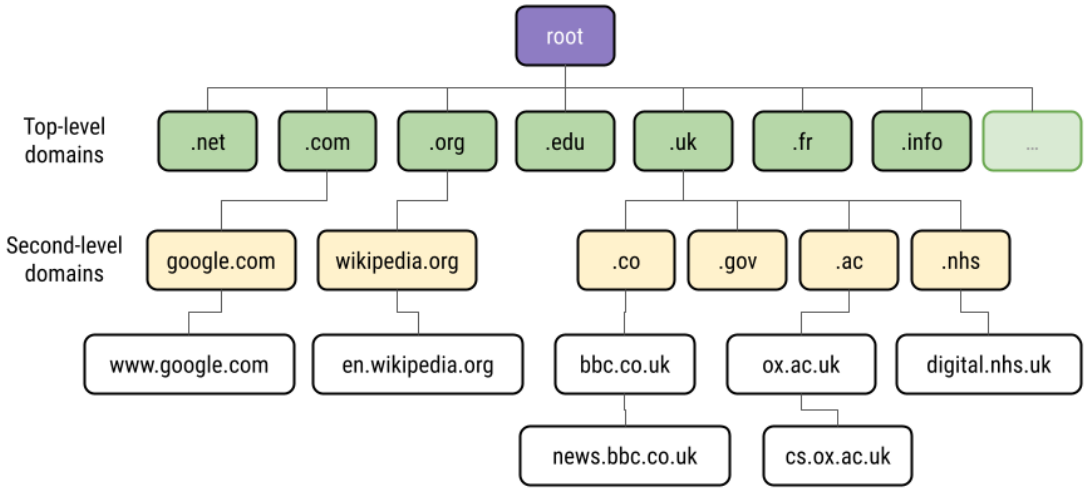

Figure 11 - The organisation of domains within the DNS namespace

As said, at the top of the DNS namespace tree is the root zone, where the DNS system's root severs operate. These severs can direct DNS resolvers to the appropriate TLD nameserver for a given DNS lookup request.

Each TLD is itself a zone and contains all the domains within it. For example, the nameservers for the .com TLD zone will be able to tell a resolver the IP address for the authoritative name servers for any domain ending in .com, such as facebook.com, google.com, microsoft.com, etc.

In addition to domains such as google.com, second-level domains also operate at the layer immediately beneath TLDs. Second-level domains (2LDs) within the .uk TLD zone include .co, .ac., etc, that then form addresses such as reading-school.co.uk and gov.org.uk. Within each second-level domain you will then find the ultimate domain names for each organisation as leaves within the tree, possibly within further

[14] https://developer.mozilla.org/en-US/docs/Glossary/TLD
[15] https://www.cloudflare.com/en-gb/learning/dns/glossary/dns-zone/

subdomains (such as cs.ox.ac.uk, where ‘cs’ is a subdomain of ‘ox’ which is a subdomain within the ‘.ac.uk’ domain).

Uniform Resource Locators (URLs)

Read: [What is a URL? - Learn web development | MDN (mozilla.org)]

With the Fully Qualified Domain Name we are able to access a particular host on a particular network (domain). However, when we access a page from a website, or a particular file on a server, we accessing a specific **resource** from that host. We therefore need a means of referencing a particular resource/file from a particular machine on the Internet and this is done with a **URL**, or **Uniform Resource Locator**.

Definition: A URL is a reference to a unique resource/file on a specific host on the Internet.

A URL is comprised of three main parts:

- The protocol required to access the resource
- The FQDN of the machine hosting the resource or its IP address
- The path to the specific resource on that host

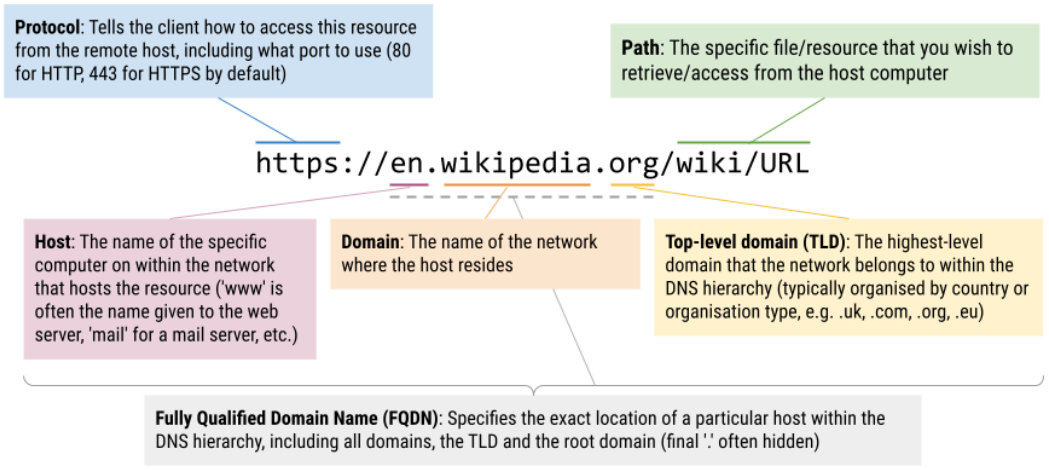

Figure 12 - The parts of a URL, including the Fully Qualified Domain Name (FQDN) of the host

Whilst most URLs follow the pattern shown in the example above, it is possible to provide additional information that affect how the request is directed or handled upon its receipt. For example:

- It is possible to encode additional values (parameters) on the end of the URL which can be used to provide additional information to the server hosting the resource that might affect how a request is handled (e.g. a search query, authentication key, etc.):

  https://www.google.com/?query=url+parameters

- It is possible for the host to be specified using an IP address rather than FQDN:

  http://205.43.114.89/index.html

- You can also specify specific ports immediately after the host (as either IP address or FQDN), for example to access a web-app running on port 8080 rather than the standard 80:

  http://nat.awdimmick.net:8080/

Summary: locating resources on the internet

- Data (including webpages and files) and resources (mail servers, FTP servers, webservers, etc) are stored on or provided by host computers across the internet.
- Each host has a unique public IP address; however these are difficult for humans to remember and interpret.
- To avoid the need to use IP addresses, the Domain Name System provides a means of addressing specific hosts using a human-readable name.
- The exact location of a host within the DNS namespace is identified by its Fully Qualified Domain Name (FQDN), which comprises the name of the host and all of its domains, including its top-level domain (TLD) and the root domain ('.')
- When a host is to be contacted by its name, a DNS lookup query is sent from a local computer to its DNS resolver which, in turn, contacts other DNS name servers until it finds the matching IP address for the requested host. This is passed back to the originating computing to use to establish a connection.
- Domain names are organised within a hierarchical structure, with the root zone at the top, following by top-level domains (e.g. .com, .edu, .org, .uk, etc.) and, within each of those, second-level domains (e.g. goole.com, wikipedia.org, co.uk).
- In order to access a specific resource form a specific host, URLs (Uniform Resource Locators) are used. Each URL specifies: the protocol being used to access the resource, the FQDN or IP address of the host, the path to the resource being requested.

The role of Internet registries

By now you will hopefully agree that the internet is a complex beast, with many systems, protocols and processes working together to enable the global connection of devices to one another.

There are number of organisations that are responsible for the administration of various aspects of the internet. Without them, it would be impossible to coordinate things like IP address allocation and the DNS top-level domains and, in turn, no internet as we know and enjoy it today. Such organisations are known as **internet registries**.

The allocation of public IP addresses to organisations is the responsibly an organisation called **IANA** (the internet Assigned Numbers Authority). But IANA hasn't got time to administer all of the IPv4 and v6 addresses to each end-user on the Internet. Instead we have – you guessed it – another administrative hierarchy!

Beneath IANA are five regional internet registries[16] (RIRs), each responsible for the administration of **Internet numbers** (IP addresses and Autonomous System numbers[17]) for a particular area within the world. RIRs allocate blocks of addresses to local internet registries (LIRs) who can then assign them to customers and organisations. Most LIRs are ISPs, or organisations such as Janet (the Joint Academic Network), which is the responsible for allocating IP addresses to UK higher-education institutions[18].

[16] https://en.wikipedia.org/wiki/Regional_Internet_registry
[17] https://en.wikipedia.org/wiki/Autonomous_system_(Internet)
[18] https://community.jisc.ac.uk/library/janet-services-documentation/internet-registries

ICANN (the Internet Corporation for Assigned Names and Numbers) is the organisation responsible for managing the DNS root zone and defining TLDs. ICANN distributes responsibility for maintaining the authoritative name servers (and therefore the domains) within each TLD to different organisations, for example .uk is managed by **Nominet**, the official registry of all .uk domains[19], whilst .com is managed by Verisign[20].

For individuals and organisations that wish to "buy" a domain name, there are many **domain registrars** to provide this service. Domain registrars are organisations that have permission to register domain names on behalf of the organisation that operates a given TLD. Examples of domain registrars include godaddy.com, cloudflare.com, 123-reg.co.uk to name just a few. In reality it is not possible to outrightly own (purchase) a domain, but instead registrants of a domain need to renew their registration on a regular basis (typically yearly or every two years). If a registration lapses, the domain is made available once again to be registered and managed by another organisation.

[19] https://www.nominet.uk
[20] https://www.verisign.com/en_US/domain-names/index.xhtml?loc=en_US

Further reading

The structure and function of the Internet is a huge topic and you certainly won't be expected to understand its full complexity at A Level. However, if you are interested in learning more about any of the concepts covered, here are a selection of links and sources that I used to put together this document.

General

- How Does the Internet Work? (Main source for this document) http://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm
- How Internet Infrastructure Works | HowStuffWorks (an accessible overview that covers all of the main points) https://computer.howstuffworks.com/internet/basics/internet-infrastructure.htm/printable
- ICANN - Wikipedia: https://en.wikipedia.org/wiki/ICANN
- Internet Assigned Numbers Authority (iana.org): https://www.iana.org/
- Retiring the [pre-Internet] NSFNET Backbone Service – Chronicling the end of an era (PDF): https://www.merit.edu/wp-content/uploads/2019/06/Retiring-the-NSFNET-Backbone-Service_-Chronicling-the-End-of-an-Era-1.pdf
- Ben Eater's whole video course on how the networks and the Internet work: https://eater.net/inet

IP addresses and routing

- IPv4 and IPv6 address formats - IBM Documentation: https://www.ibm.com/docs/en/ts4500-tape-library?topic=functionality-ipv4-ipv6-address-formats
- DNS, Routing, BGP (and why Facebook went down) – Ben Eater: https://youtu.be/-wMU8vmfaYo
- Border Gateway Protocol - Wikipedia: https://en.wikipedia.org/wiki/Border_Gateway_Protocol
- Watch: Hop-by-hop routing (Ben Eater): https://www.youtube.com/watch?v=VWJ8GmYnjTs (13m49)

DNS

- Watch: Computerphile on DNS: https://youtu.be/uOfonONtIuk
- DNS in Detail: https://youtu.be/jpTY1S5vs9k
- Some history on the "A" root server: https://a.root-servers.org/
- How the DNS root servers use a single IP address to point to many physical servers around the world with IP Anycast: https://en.wikipedia.org/wiki/Anycast#Domain_Name_System
- Top-level domains - Wikipedia: https://en.wikipedia.org/wiki/Top-level_domain

Networking in general

- These courses from Computing At School (CAS), which is designed to teach teacher how to teach the wider networking topic (includes some really good walkthroughs within its explanations):
    - Data networks and IP addresses Introduction to Data networks and IP addresses: https://www.open.edu/openlearncreate/course/view.php?id=2771
    - Identify network hardware and protocols Identify network hardware and protocols: https://www.open.edu/openlearncreate/course/view.php?id=2772
    - Purpose of network hardware and protocols The purpose of network hardware and protocols: https://www.open.edu/openlearncreate/course/view.php?id=2774
    - The operation of LAN and WAN hardware and protocols The operation of LAN and WAN hardware and protocols: https://www.open.edu/openlearncreate/course/view.php?id=2784

Appendix: Traceroute examples (for routing)

Here are some examples of the output produced when using traceroute to determine the path between my local computer and a remote host. Some of these hosts are within the UK and some in the US.

From any of these, we would be able to construct a diagram showing the route the packets took, looking at the regional and local networks along the way. Geolocation helps identify the owner of each IP, allowing us to know which networks the packets are travelling through at each stage.

C10-E4-TEACH (10.1.128.203) -> www.apple.com (2.19.168.202)

```
C:\Windows\system32>tracert www.apple.com

Tracing route to e6858.dscx.akamaiedge.net [2.19.168.202]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2     2 ms     2 ms     1 ms  81.144.164.209
  3     3 ms     2 ms     3 ms  86.189.33.32
  4     3 ms     3 ms     3 ms  core2-te0-13-0-3.southbank.ukcore.bt.net [109.159.254.30]
  5     3 ms     4 ms     3 ms  peer3-et0-1-7.redbus.ukcore.bt.net [194.72.16.106]
  6  1301 ms   440 ms   214 ms  109.159.253.245
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     3 ms     8 ms    15 ms  a2-19-168-202.deploy.static.akamaitechnologies.com [2.19.168.202]
```

C10-E4-TEACH (10.1.128.203) -> www.google.co.uk (142.250.200.3)

```
C:\Windows\system32>tracert www.google.co.uk

Tracing route to www.google.co.uk [142.250.200.3]
over a maximum of 30 hops:

  1     1 ms     1 ms     2 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2    <1 ms    <1 ms    <1 ms  81.144.164.209
  3     3 ms     3 ms     3 ms  86.189.33.32
  4     4 ms     4 ms     4 ms  core1-te0-11-0-12.colindale.ukcore.bt.net [109.159.254.2]
  5    28 ms     4 ms     5 ms  peer3-et3-0-1.slough.ukcore.bt.net [109.159.252.80]
  6     4 ms     4 ms     4 ms  109.159.253.73
  7     6 ms     6 ms     6 ms  216.239.42.41
  8     4 ms     4 ms     4 ms  108.170.234.231
  9    24 ms    13 ms    15 ms  lhr48s29-in-f3.1e100.net [142.250.200.3]
```

Home (Virgin Media ISP) -> www.google.co.uk

```
pibb@bbpi:~ $ traceroute www.google.co.uk
traceroute to www.google.co.uk (142.250.179.227), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  9.468 ms  9.242 ms  9.139 ms
 2  10.53.39.149 (10.53.39.149)  23.640 ms  23.688 ms  23.287 ms
 3  winn-core-2b-xe-031-0.network.virginmedia.net (62.253.120.141)  23.605 ms  23.499 ms  23.635 ms
 4  * * *
 5  eislou2-ic-3-ae0-0.network.virginmedia.net (94.174.238.226)  36.895 ms  36.804 ms  37.063 ms
 6  72.14.221.42 (72.14.221.42)  36.453 ms  24.093 ms  24.199 ms
 7  108.170.246.161 (108.170.246.161)  21.815 ms 108.170.246.129 (108.170.246.129)  25.615 ms 108.170.24
6.161 (108.170.246.161)  26.945 ms
 8  142.251.54.27 (142.251.54.27)  30.606 ms 142.251.54.25 (142.251.54.25)  26.498 ms  25.830 ms
 9  lhr25s31-in-f3.1e100.net (142.250.179.227)  25.841 ms  23.785 ms  27.664 ms
pibb@bbpi:~ $
```

C10-E4-TEACH (10.1.128.203) -> 8.8.8.8 (Google public DNS server)

```
C:\Windows\system32>tracert 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:

  1    <1 ms     7 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2    <1 ms    <1 ms    <1 ms  81.144.164.209
  3     3 ms     2 ms     3 ms  86.189.33.32
  4     4 ms     4 ms     6 ms  core1-te0-11-0-14.colindale.ukcore.bt.net [109.159.254.10]
  5     6 ms     4 ms    17 ms  peer8-et-0-0-1.telehouse.ukcore.bt.net [62.172.103.170]
  6     *        6 ms     6 ms  109.159.253.191
  7     6 ms     6 ms     6 ms  216.239.54.127
  8     5 ms     5 ms     5 ms  142.250.215.127
  9    15 ms     4 ms     5 ms  dns.google [8.8.8.8]

Trace complete.
```

Home (Virgin Media ISP)-> 8.8.8.8

```
pibb@bbpi:~ $ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  4.275 ms  4.222 ms  4.876 ms
 2  10.53.39.149 (10.53.39.149)  23.999 ms  22.826 ms  23.709 ms
 3  winn-core-2a-xe-2112-0.network.virginmedia.net (62.253.120.245)  23.844 ms  23.778 ms  23.709 ms
 4  * * *
 5  eislou2-ic-4-ae0-0.network.virginmedia.net (62.254.59.130)  28.383 ms  27.838 ms  28.008 ms
 6  142.250.160.116 (142.250.160.116)  35.976 ms  24.615 ms  22.318 ms
 7  * * *
 8  dns.google (8.8.8.8)  20.991 ms  21.282 ms 172.253.65.210 (172.253.65.210)  24.315 ms
pibb@bbpi:~ $
```

C10-E4-TEACH (10.1.128.203) -> www.reading.ac.uk (134.255.0.151)

```
C:\Windows\system32>tracert www.reading.ac.uk

Tracing route to wap-slb-vip.rdg.ac.uk [134.225.0.151]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2    <1 ms    <1 ms    <1 ms  81.144.164.209
  3     2 ms     3 ms     2 ms  86.189.33.32
  4     4 ms     4 ms     6 ms  core1-te0-11-0-14.colindale.ukcore.bt.net [109.159.254.10]
  5     5 ms     4 ms     4 ms  peer2-et4-0-1.slough.ukcore.bt.net [194.72.16.206]
  6     4 ms     4 ms     9 ms  linx-gw1.ja.net [195.66.224.15]
  7     6 ms     4 ms     4 ms  ae23.londtt-sbr1.ja.net [146.97.35.169]
  8     4 ms     4 ms     4 ms  ae28.londtw-sbr2.ja.net [146.97.33.62]
  9     5 ms     5 ms     5 ms  ae30.londpg-sbr2.ja.net [146.97.33.5]
 10     6 ms     5 ms     5 ms  ae19.readdy-rbr1.ja.net [146.97.37.194]
 11    18 ms    21 ms    20 ms  reading-university-1.ja.net [193.63.109.26]
 12     6 ms     6 ms     7 ms  xe-0-0-7.fw-ext.net.rdg.ac.uk [134.225.255.38]
 13     8 ms     7 ms     7 ms  alumni.reading.ac.uk [134.225.0.151]
```

C10-E4-TEACH (10.1.128.203) -> www.bris.ac.uk (137.222.0.37)

```
C:\Windows\system32>tracert www.bris.ac.uk

Tracing route to www.bris.ac.uk [137.222.0.37]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2     1 ms     7 ms    <1 ms  81.144.164.209
  3     3 ms     2 ms     2 ms  86.189.33.32
  4     3 ms     3 ms     3 ms  core2-te0-13-0-1.southbank.ukcore.bt.net [109.159.254.22]
  5    10 ms     3 ms     8 ms  peer7-et-4-0-5.telehouse.ukcore.bt.net [194.72.16.130]
  6     3 ms     3 ms     3 ms  109.159.253.93
  7    10 ms     3 ms     3 ms  ae24.londtt-sbr1.ja.net [146.97.35.193]
  8     4 ms     4 ms     3 ms  ae28.londtw-sbr2.ja.net [146.97.33.62]
  9     4 ms     4 ms     5 ms  ae30.londpg-sbr2.ja.net [146.97.33.5]
 10     6 ms     6 ms     7 ms  ae0.briswe-rbr1.ja.net [146.97.37.202]
 11    14 ms     9 ms     9 ms  ae4-1000.bradss-ban1.ja.net [146.97.67.106]
 12    10 ms     9 ms    13 ms  university-of-bristol.ja.net [146.97.144.2]
 13    10 ms     9 ms     9 ms  www.bris.ac.uk [137.222.0.37]
```

Comparing Reading and Bristol universities is particularly interesting as they use the same ISP (janet – Joint Academic Network), to which all UK universities are connected. JANET is its own Tier 2 network, so we can see that we hit similar (and in some cases the same) routers along the way for parts of the journey to each webserver.

Home (Virgin Media ISP) -> www.reading.ac.uk (134.225.0.151)

```
pibb@bbpi:~ $ traceroute www.reading.ac.uk
traceroute to www.reading.ac.uk (134.225.0.151), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  3.028 ms  3.948 ms  4.425 ms
 2  10.53.39.149 (10.53.39.149)  15.172 ms  16.994 ms  16.485 ms
 3  winn-core-2b-xe-031-0.network.virginmedia.net (62.253.120.141)  17.443 ms  17.656 ms  24.945 ms
 4  * * *
 5  * * *
 6  host-62-254-42.174.not-set-yet.virginmedia.net.42.254.62.in-addr.arpa (62.254.42.174)  39.537 ms  19
.476 ms  19.943 ms
 7  213.46.175.50 (213.46.175.50)  20.842 ms  20.026 ms  26.104 ms
 8  ae24.londhx-sbr1.ja.net (146.97.35.197)  26.754 ms  27.421 ms  27.134 ms
 9  ae29.londpg-sbr2.ja.net (146.97.33.2)  27.823 ms  28.422 ms  27.843 ms
10  ae19.readdy-rbr1.ja.net (146.97.37.194)  19.596 ms  24.278 ms  23.941 ms
11  reading-university-1.ja.net (193.63.109.26)  34.753 ms  35.030 ms  34.098 ms
12  xe-0-0-7.fw-ext.net.rdg.ac.uk (134.225.255.38)  22.487 ms  26.755 ms  26.241 ms
13  wap-slb-vip.rdg.ac.uk (134.225.0.151)  26.816 ms  23.404 ms  21.762 ms
pibb@bbpi:~ $
```

Compare similarities between tracing a route from school (BT) and home (Virgin media). Notice the second hop is to a 10.x.x.x address, which is private, therefore on the Virgin Media internal network.

C10-E4-TEACH (10.1.128.203) -> win.awdimmick.net (86.20.104.26)

```
C:\Windows\system32>tracert win.awdimmick.net

Tracing route to win.awdimmick.net [86.20.104.26]
over a maximum of 30 hops:

  1    <1 ms     4 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2     4 ms    <1 ms    <1 ms  81.144.164.209
  3     3 ms    11 ms     3 ms  86.189.33.32
  4     4 ms     4 ms     4 ms  core1-te0-11-0-12.colindale.ukcore.bt.net [109.159.254.2]
  5     3 ms     3 ms     3 ms  peer3-et3-0-1.redbus.ukcore.bt.net [194.72.16.184]
  6     3 ms     4 ms     3 ms  109.159.253.63
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9    14 ms    14 ms    14 ms  winn-core-2b-ae5-0.network.virginmedia.net [62.254.85.154]
 10    15 ms    15 ms    15 ms  rdng-cmts-26-pc2-650.network.virginmedia.net [62.253.120.142]
 11    25 ms    24 ms    23 ms  cpc96358-rdng26-2-0-cust25.15-3.cable.virginm.net [86.20.104.26]

Trace complete.
```

This is the route to one of my home computers, where I use Virgin Media as my ISP. We can see the packets taking their same journey up through BT's network, until they move onto the VirginMedia network at 62.254.85.154

C10-E4-TEACH (10.1.128.203) -> www.facebook.com (157.240.214.35)

```
C:\Windows\system32>tracert www.facebook.com

Tracing route to star-mini.c10r.facebook.com [157.240.214.35]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  smoothwall.local.readingschool.reading.sch.uk [10.1.130.6]
  2    <1 ms    <1 ms     1 ms  81.144.164.209
  3     2 ms     2 ms     2 ms  86.189.33.32
  4     3 ms     4 ms     3 ms  core1-te0-13-0-1.southbank.ukcore.bt.net [109.159.254.18]
  5     5 ms     4 ms     5 ms  peer3-et3-0-2.redbus.ukcore.bt.net [194.72.16.192]
  6    36 ms     4 ms     3 ms  109.159.253.201
  7     6 ms     4 ms     4 ms  po161.asw01.lhr3.tfbnw.net [129.134.44.204]
  8     4 ms     4 ms     *     po232.psw03.lhr8.tfbnw.net [129.134.50.79]
  9     4 ms     4 ms     4 ms  157.240.39.129
 10     8 ms     4 ms     5 ms  edge-star-mini-shv-02-lhr8.facebook.com [157.240.214.35]

Trace complete.
```

This is a good example of a traceroute, which goes through BT's regional routers and along to Facebook's, eventually hitting their webserver in the US. This IP can be typed directly into browsers as an example of requesting webpages without DNS.