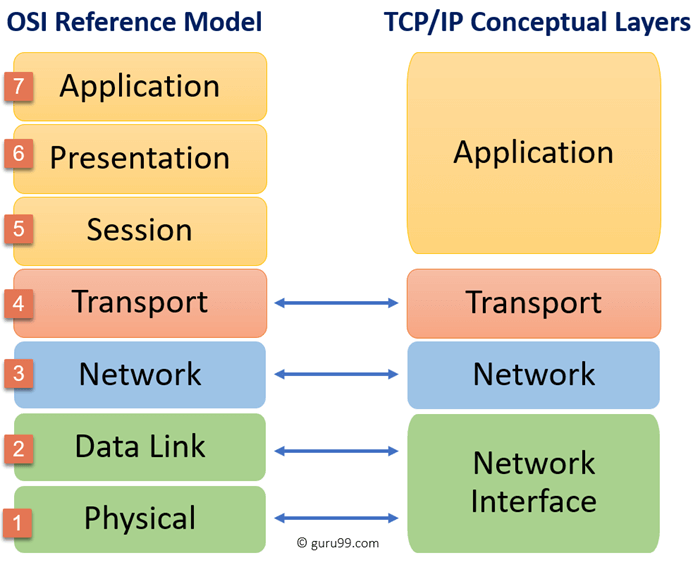
Networks

# Layers

Most networks are refered to by the seven layer OSI model of networking, however they are actally built up from the TCP/IP model. Luckly the two are very similar, however it is important to note that network engineers will rerference layers from the OSI, e.g. the application layer is layer 7:



Layer 1 - Physical (TCP/IP - Network Access Layer)

Layer 1 is the phyical layer of a network and is responsible for the transmission and reception of unstructured raw data between devices, built up of cables and electrical signals. Common concepts of the phyical layer are: bitrate control, transmission mode (simplex, duplex). Specifications include Ethernet, bluetooth, USB, etc.

Layer 2 - Datalink (TCP/IP - Network Access Layer)

Layer 2 is the datalink layer and provides the means for delivering data frames between nodes (devices) on the same level of network (local network). Messages are routed by the layer 2 address for each device which is defined by the devices Media Access Code (MAC) address.

Layer 2 devices can detect and sometimes correct layer 1 network issues. IEEE 802 divides the data link layer into two sublayers:

* Medium Access Control (MAC) - how devices in the netwrok gain access to a medium and transmit data
* Logical Link Control (LLC) - Identifying and encapsulating network layer protocols, controlling error checking and frame synchronization

Security, such as authenticated encryption can be applied at layer 2 using MACSec.

Layer 3 - Network (TCP/IP - Internet)

Layer 3 is the network layer and provides the means for transfering variable length network packets, via one or more networks. Layer 3 packets are routed by Internet Protocol (IP) address, which is an address given to each device on the network for the purpose of identificaion and location. In an isolated network each IP address will be unique to allow for accurate routing.

While a transport protocol at network level may provide reliable message delivery it is not required to be garaunteed. Common protocols are:

* IP - Internet protocol, responsible for delivery packets from source to destination host
* ICMP - Internet Control Message Protocol, reponsible for providing host with information about network problems
* ARP - Address Resolution Protocol, responsible for finding the hardware address of a host from a known IP address

Layer 4 - Transport (TCP/IP - Host to Host Layer)

The transport layer provides the function means of transferring variable-length data sequences from source to destination host, while maintaing quality of service (error free data delivery). To maintain quality of service the transport layer controls a link through flow control, segmentation, and error control.

Transport layer can keep track of segments and retransmit those that fail delivery. Common Layer 4 protocols are:

* Transmission Control Protocol (TCP) - Provices reliable end-to-end communication between systems. Sequences and segments data, and also controls data flow. While reliable is has large overhead due to its features.
* User Datagram Protocol (UDP) - Connectionless oriented protocol which provides low cost unreliable data transmission. Does not have the reliablity features of TCP. Good for realtime applications where any lost data does not require retransmitting., for example video streaming.

Layer 5 - Session

Opens communication between the local and host machine. Common protocols are:

* L2TP
* RTCP
* H.245
* SOCKS

Layer 6 - Presentation

Controls data format and encryption. For example, when using html, the presentation layer will convert application data into html files for transport.

Layer 7 - Application (TCP/IP - Application Layer)

Combines the top three layers of the OSI model (5 - Session, 6 - Presentation, 7 - Application), and is responsible for node-to-node communication and controls user-interface specifications. It manages communcation between applications, directs data to correct program, determines adaquate resources for communcation.

Common protocols are:

* HTTP(S) - Hypertext Transfer Protocol, used to transport communcations between web browsers and servers.
* SSH - Secure Shell, maintains secure connection through TCP/IP for terminal emulation
* NTP - Network Time Protocol, used to synchronise clocks on computers to one standard time source. Useful for time senstive processes.
* FTP - File Transfer Protocol, faciliates transfer of files between two machines, also includes a program to provide a user interface to the file system. Typicall port 20 and 21.

# Components

Hubs

Network hubs are dumb relays for network messages. When a message is received on a phyical port, it is relayed to all other connected ports on the hub. Since messages are not relayed directly to correct recipient, this causes many collisions, slowing down network speed and reducing privacy.

Switches

Switches are similar to hubs however they have Content Addressable Memory (CAM) in which they can store a table relating the devices on the network to which port they are connected to on the switch. This allows switches to direct messages receieved from a device to it specific reciepent on the network.

Network switches operate on layer 2, transmitting only frames defined by their MAC address.

The CAM table is populated by source addresses, since these are definitive.

# Utilities

Ping

Ping is a network utility used to test the reachability of a host on an IP network. Ping operates by sening Internet Controler Message Protocol (ICMP) echo request packets to the target host, then waiting for an ICMP echo reply. Pinging a host can provide various details about the network connection, such as:

* round trip time (ping time)
* packet loss
* errors
* standard deviation of ping time
* reponse times

Most opertating systems have a ping ultilty built into their command line. For example, the linux CLI for ping is:

ping -c <number-of-probes> <host-address>

# Routers

General

A router's, also called a gateway, job is to connect networks with different ip address groups.

ARP Requests

When newly connecting to a local network, the local machine may only know the ip address of a host it is trying to connect to, yet layer 2 switches can only route using MAC addresses. Therefore, the local machine must discover the MAC address for the desired host using an Address Resolution Protocol (**ARP**) request. The ARP message is simply a message sent to the switch with a MAC address of FFFF:FFFF:FFFF and the host IP address, which is then broadcasted to all other devices connected to the switch. If a connected device matches the IP address of the ARP message, it will reply to the swtich, returning the MAC address to the local machine, allowing for future frames to be sent to the correct address.

However, when the IP address is not within the range of the local network, the ARP request from the local machine will be sent to the network router (gateway) IP address. The router replies with its own MAC address, and the local machine will start sending its data frames to the router. The new data frames will be transmitted by the switch over layer 2 to the router, but they will also the layer 3 address details which the router can use to forward the message on to the external network. If the router does not already know the MAC address for the IP address for which the received packets is addressed to, it will also send out ARP requests to find it in its local network.

In modern internet networks there are many routers and therefore networks chained together. So, sending packets to IP addresses external to the local network, will often result in a chain of ARP requests being sent across the internet. While this first connection may be slow, once the MAC to IP address relation is processed, it is saved in the router cache, making the process of sending future packets very fast.

NAT

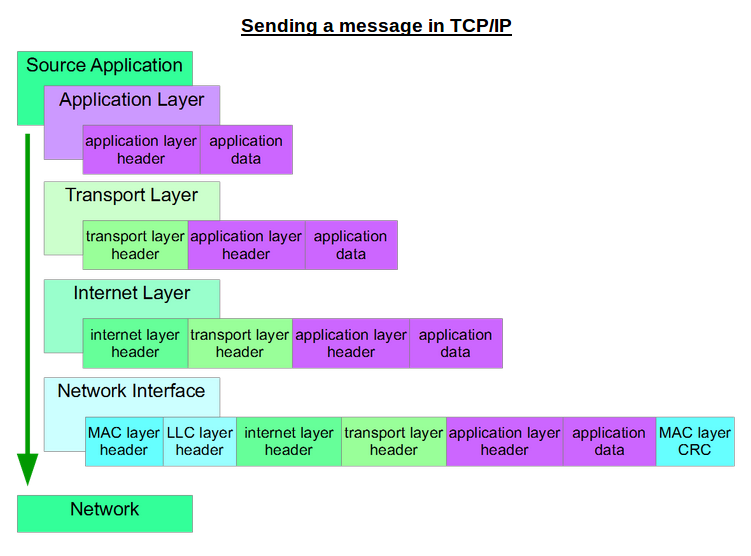
NAT - Network Address Translation is the process of remapping one IP address space to another by modifying the network address information in the IP header of the packet. In routing one public IP address is exposed for the network, and the router perform NAT on each packet to convert its local IP to public IP and vice versa. To manage multiple connections, the router opens up a new external port for every connection it is requested to route and stores this link to the local connection in the translation table. The server will then respond to the request on the new port and the router can link it to a local machine in the translation table, then perform NAT on the packet and forward it on.

# Process

TCP/IP

Creating the message:

1. Application data (layer 7) is created and encapsulated in the http data header
2. Transport layer data (layer 4), such as transport protocol (TCP/UDP) and port, is attached to the application data as a layer 4 header, creating a **segment**. Segments will normally be created from the larger application data by the transport protocol such as TCP, allowing for data flow control, error checking, and other reliablity measures.
3. Network layer data (layer 3), such as source and destination IP address, is attached to the segment in the layer 3 header, creating a **packet**.
4. Datalink information (layer 2), such as MAC addresses, is attached to the packet in the layer 2 header and trailer, encapsulating the message into a **frame**.



Sending the message:

1. Frame is sent using layer 1 (phyical layer) to the switch
2. Switch reads the layer 2 header (MAC address) and forwards frame to the corresponding device (router in this case).
3. Router receives frame and checks MAC address in layer 2 is correct
4. Router de-encapsulates the message and reads the layer 3 header, giving destination IP address.
5. Using its lookup table (or ARP if table does not contain value) the router determines the next step MAC address, be it the destination machine or next gateway router.
6. Router edits layer 3 with own source ip address and encapsulates in layer 3 header, along with new layer 2 headers contain its MAC address and the new destination MAC address
7. Router sends frame to next hop
8. Process repeated until source address is reached and the message is de-encapsulated and processed

TCP - Three Way Handshake

Start of communcation between two machines:

1. **SYN** - Local machine sends synchronisation message to host machine in order to start the communcation
2. **SYNC ACK** - Host machine sends synchronisation acknowledgement message back to local machine
3. **ACK** - Local machine sends acknowledgement message back to host machine to confirm communication