CITS3002

Daniel Paparo

1 Lecture 1

Computer network - an interconnected network of autonomous computers.

Interconnected - capable of exchanging information.

Autonomous - Not a permanent master/slave relationship between them.

This unit will focus on computer networking software and its support by operating systems and programing languages

1.1 Networks

1.1.1 Why do we need networks

- Users can access shared and distributed data
- Users can access shared and distant physical resources when their resources are limited
- Networks can provide "fault tolerance" for hardware failiers
- Network provide cost benefits (centralising with mainframes)
- Permit centralized facilities and remote collaboration

1.1.2 Research interests in networking

- Unreliable communications
- Temporal and spatical decoupling of work patterns and communities (ie. communication and observation at a distance)
- Privacy and security
- Parallel programming

1.2 Security

1.2.1 Threats to our networks

- interruption to service
- interception of data
- modification of data
- fabrication of data

1.2.2 Active threats

- self reproducing works
- a logic bomb or botnet
- a lurking trojan horse, trapdoor or botnet program

- a virus embedded in an application
- a rootkit embedded in an operating system and its utilities

1.2.3 Passive threats

- Keyboard logging
- EMF monitoring and bridging air-gap networks
- wired or wireless network packet sniffing
- simple social engineering

1.2.4 The attack profile

Attack tools are often prebuilt attacks - a wide veriety of attacks are freely available on the internet and come with very good documentation (cookbook attacks).

There can be many personal and professional motives:

- Attacks may be indiscriminate
- attacks out of curiosity
- attacks out of fear, greed, and malice
- political attacks
- industrial espionage
- electronic warfare

and technical motives:

- Intelligence gathering
- Denial of service
- Reading of protected information
- Modification of protected information
- Execution of arbitrary commands

1.2.5 Vulnerabilities

People, practices, programs and protocols are all vunerable to attacks. These can be caused by:

- Insufficient testing
- Dependence on flawed resources
- Trusting untrustworthy data
- inappropriate use of external resources
- Incompatible or incomplete design specifications
- The complexity of large applications
- The complexity of multiple interacting applications

1.2.6 Outcomes of attacks

- Financial gain
- incur a cost to victim
- To gain media exposure
- to serve a politcal agenda
- Servicing fear, greed or a desire for fame

1.2.7 Countermeasures

- Improved education
- Reduced system complexity
- increased software testing
- improved physical protection
- improved authentication
- improved authorization
- increased use of biometrics
- improved auditing
- a better understanding of encryption
- Introducting liabilities of software vendors

1.2.8 Encryption

Encryption is really only a way of slowing the bad guys down - The protection only needs to be good enough to slow them down enough until the threat can be dealt with.

1.2.9 Open source

Is open source really better? The ability to view the code allows for openness but it doesnt necisarrily mean any one may have - This may mean it is still as insecure but now people trust it. Proprietry software doesnt allow for viewing of the source code but that doesnt mean it is safe.

1.3 Network protocols

Protocol - A process/sequence of steps which allows us to exchange data/knowledge.

The difference between data and knowledge - Data is the actual 0s and 1s, whereas knowledge can be the lack of data (damaged or lost).

Some common protocols:

- HTTP - Hypertext transfer protocol
- NNTP - Network news transfer protocol
- SMTP - Simple mail tranfer protocol
- NTSF - Windows-NT File System

All protocol must:

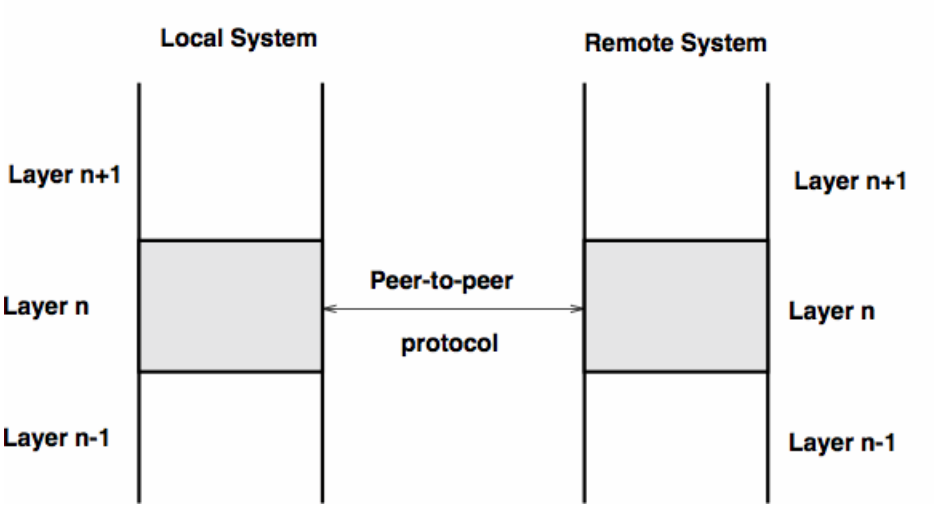
- Happen in an agreed to order
- Travel from the sender to the correct reciever
- Contain the correct, unambiguous, data

1.3.1 ISO/OSI reference model

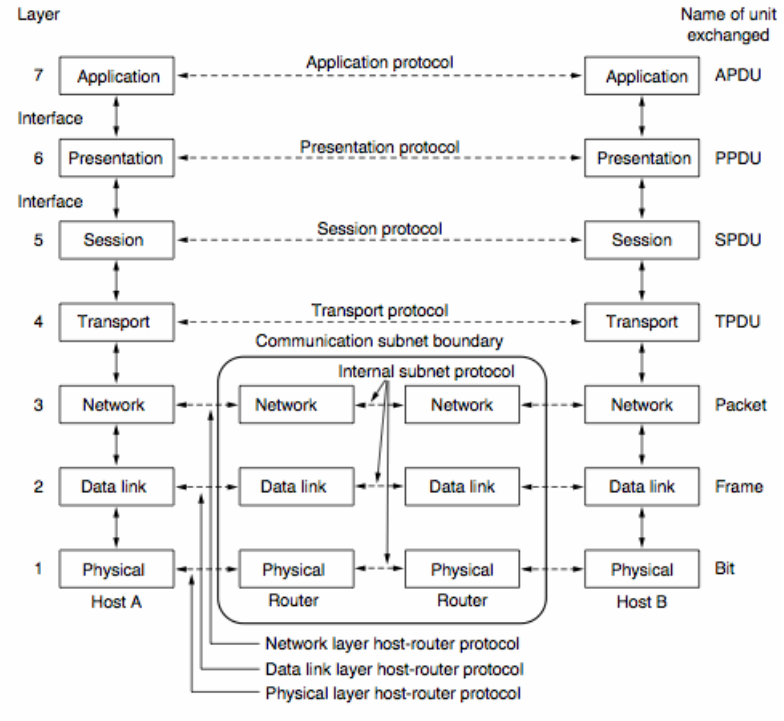
Barely any computer system these days will use this protocol today - but historically this was a very popular archetecture for network modelling - No implementation was provided. The OSI model has 7 layers of protocol which interact with the layer below it or the layer above it.

But why a 7 layer model?

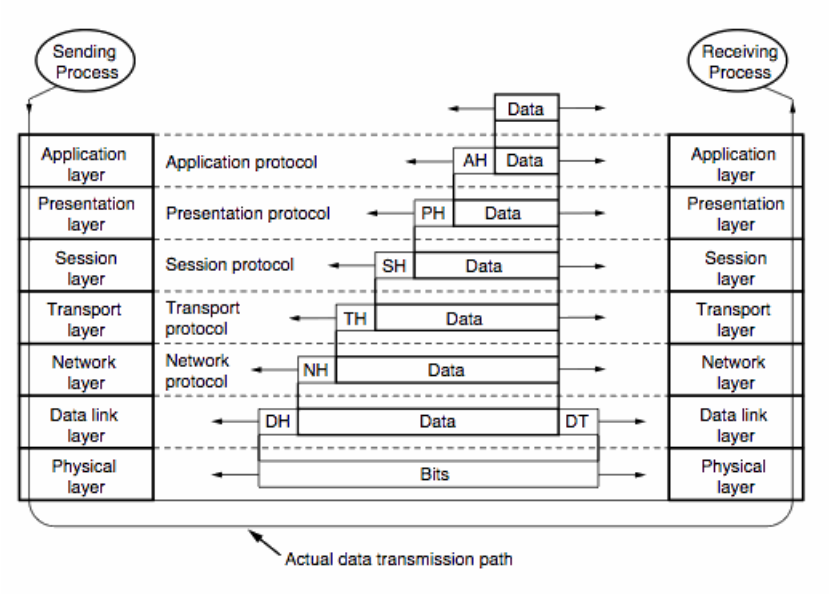
- A layer corresponding to each layer of abstraction



ISO/OSI



A layered approach



Physical layer

- Each layer provides a well defined, independednt function
- Within each layer unique protocol standards should be enforceable
- There should be a minimum of traffic between layers/across interfaces
- The number of layers should be sufficiently large that distinct functions are in different layers and that there are not too many layers for the whole model to become unmanageable

The physical layer Is responsible for transmitting a (raw) bit stream over the physical communication medium. As such it is concerned with the electrical and mechanical interface between the data and the physical medium.

The physical layer presents a bit stream to the layer above.

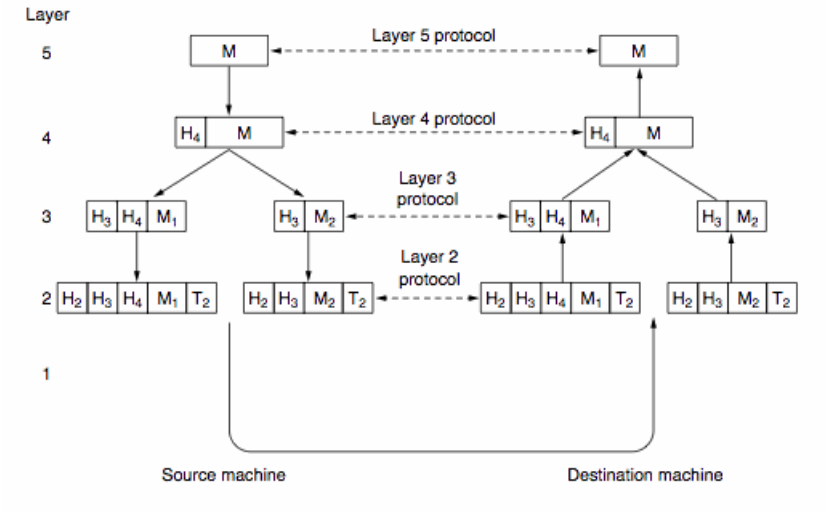
Data-Link Layer Takes the bit stream from the physical layer and constructs logical chunks of data termed frames. The purpose of framing is to ensure the reliable transmission of information by performing limited error detection and recovery.

Network layer Is responsible for providing the connection between "end systems" across a network. These connections might include multiple, intermediate links and are intended to be independent of the (sub)networks used to transmit the data.

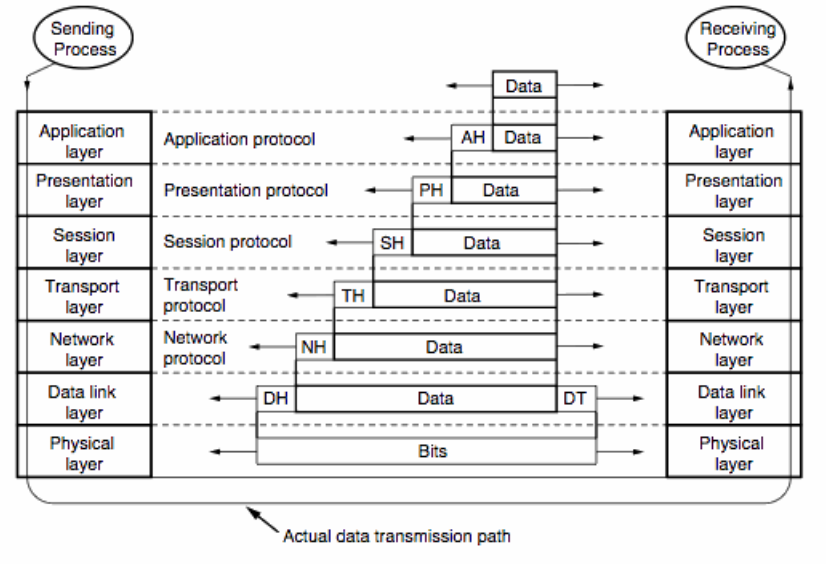
Network layer functions include:

- routing: deciding how to transmit frames between source and destination using addresses.
- relaying: enables data transfer (transparently) across intermediate (sub)networks.
- flow control: matches traffic flow with the physical capacity of a transmission path.
- sequencing: control ordering of frames across a network.

Transport layer Provides a reliable end-to-end service independent of the network topology. This is achieved by splitting messages into network sized packets and joining them back together



Data-link layer



Network layer

again at the other end.The transport layer often supports multiplexing to optimize network cost (several transport connections mapped into a single network connection) or splitting to enhance services (single transport to multiple connections).

Session layer Is the first upper layer crucial to internetworking and manages the dialogue between end systems. Typically the session layer provides:

- establishment and closing of connections.
- synchronization to allow checking and recovery of data.
- negotiation of full and half duplex communication.

Presentation layer Provides a standard format for transferred information by overcoming compatability problems between systems using dissimilar data encoding rules and (possibly) different display (input and output) technologies.

Application layer Provides the interface between the application processes. In particular, functions such as file transfer, remote job execution (remote procedure calls) and application independent virtual terminal support are provided. In overview, the application layer provides transparency to the users, load balancing between machines, data bases (banks and airline reservations), and the prospect of distributed operating systems.

2 Lecture 2

2.1 Physical layer

The physical layer is the physical wire or bandwidth, but also its interactions with the drivers. Properties of the physical layer will constrain the data link layer. When things go wrong at this layer it is the resposibility of the data link layer to fix it

2.1.1 Metrics of network measurment

Latency: also known as propogation delay, is the time which it takes for a bit of information to get from origin to destination

Bandwidth: How many bits come out at any one time

Throughput: The actual achieved number of bits travelling through the medium

$$T_{Latency} = T_{Propagation} + T_{Transit} + T_{queue}$$

$$T_{Propagation} = \frac{distance}{mediumspeed}$$

$$T_{Transit} = \frac{size}{bandwidth}$$

$$T_{Queue} = timeinlocalandremotesystem$$

Transmission errors can be caused by:

- Thermal background noise
- Impulse noise

- Distorted frequencies
- Cross talk
- Reduction of dynamic range

2.1.2 Data frames

Packets: a finite group of bits. At the datalink layer we more talk about frames
The data link layer nee to distinguish between the start and the end of a frame:

- Could potentially start with a count
  - This works until there is an error
- We can put a frame around it (like a special character) to mark the start and end of a frame

Bit Stuffing: Putting some nullifying character like: `"He said, \"Hi\" "`, where the `\` is a nullifying character.
Data layer and physical layer must work very closely

2.1.3 Error detection and correction

Detection is far easier to perform than correction - we must decide if its worth correcting errors, or just detecting there was an error and asking the sender to try sending again.
Forward error correction: Can send but cannot recieve a reply (such as satelites and stealth submarines)
We need redundancy in the data:
Parity: Essentially saying: if there is an odd or even number of 1's or 0's in the previous 7 bits - the parity bit is placed in the 8th bit and is either a `0` or a `1` depending if it is odd or even parity.
Modulo 2: highlight if 2 bits are the same or different:

$$1 \mod 1 = 0 \quad \mod 0 = 0$$

$$0 \mod 1 = 1 \quad \mod 0 = 1$$

for example:

1001101

1011110

Has a modulo-2 of:

0010001

To detect δ errors a distance of $\delta + 1$ errors is required. To correct δ errors a distance of $2\delta + 1 - 2\delta$ is required as we need to carry redundant words.
Hamming distance: The number of errors when comparing 2 sets of equal size
for example:

0000000000

1111100000

has a distance of 5. In this example 4 is the maximum number of errors allowed in these words - its hamming distance.

In a word of 7 bits, 11 bits can be used to allow for an error of just 1. We do this so that the sender can decide where the check bits are. We want these checkbits to represent some of the data bits. This is smarter that parity bits which represent all of the bits.

These methods are all about trying to avoid asking the sender again for another copy of what has already been sent - This is really good for one way communication - and also as correction is a very expensive process.

2.1.4 Cyclic redundency checks

We need to add things up, but also fold them into positions that allow better redundency.

Polynomial modulo-2 arithmatic

Generator polynomial: Generates the redundant bits

Certain polynomials are good for detecting errors.

3 Lecture 3

3.1 The data link layer

The same ides of the OSI datalink layer are still used today, but have been implemented in different ways.

The datalink layer recieves a raw bitstream from the physical layer. In combination to this:

- The physical and datalink layer should be able to figure out anything that is wrong with the data.

Only data without missing pieces/no duplicates will be sent up to the layer above it. Any data passed up will be error free and need to have no duplicates.

3.1.1 Data link layer complexity

The complexity of the datalink layer will greatly depend on the requirements.

Simplex connectionless: Sends information in one direction only

Half-Duplex connectionless: Sends data in one direction at a time

Full-Duplex connection oriented: Both parties can send and recieve at the same time

Having two sets of code (one for sending and one for recieving) of the data level code.

There is a requirement for flow control in both sending and recieving.

3.1.2 Detecting frame corruption

There will be a requirement of metadata to help us do this.

The sender: Will calculate a checksum of what is about to be sent. The reciever: Will check what has been sent matches up with the checksum

Checksum:a digit representing the sum of the correct digits in a piece of stored or transmitted digital data, against which later comparisons can be made to detect errors in the data.

A checksum will:

- Be set to a known value (0) then will be calculated
  - This is to checksum the checksum.
- Reciever will send back an acknoledgment that everything went well

3.1.3 Detecting frame loss

This is different to corruption: This is when a frame does not arrive at all. If we dont know it was sent, how do we know it didnt arrive. To combat this the sender can recieve a confirmation reciept after a frame has been recieved - The lack of a reciept shows something has been lost. If the acknoledgment does not come back:

- Reciever did not recieve the original message
- The acknoledgment was lost

This must now extend to the concept of time, and as such we will need to write in an "event driven style".

Event driven style: Will react in some way based on events which occur, and will react to those eventsr

- Allows useful work to continue until it is interrupted by the event

By starting and stopping, the code is attempting to save memory by not allowing other layers to send newe data to be processed.

3.2 Network simulation

Simulation provides a low cost way to develop and test networking protocols. More aspects of a simulated network can be varied, such as error rates, throughput etc. There are some bad points, though: simulations can be written badly or used badly.

A simple one to use is the CNET simulator.

LAN: Local area network

WLAN: Wireless Local Area Network

WAN: Wide area network

4 Lecture 4

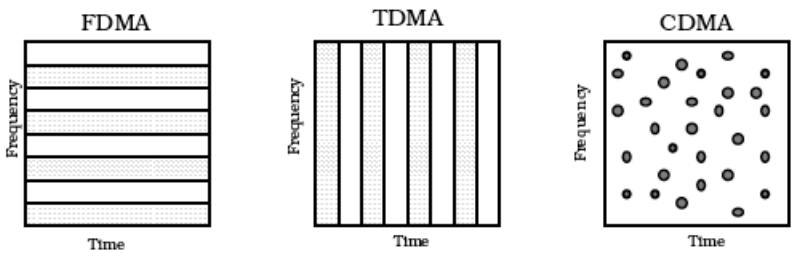
4.1 Sliding window

Allows for multiple outstanding frames, for instances such as high latency applications. This process keeps the medium as full as possible but also creates a need for a greater deal of housekeeping tools.

- If a frame is lost along the way, the reciever will tell the sender that there was a n issue, but will keep the next frames buffered for when the fix comes through.
- Reciever will buffer the frames until there is a contiguous lot of frames that can be sent up to the data link layer
- sender must retain the information until the confirmation is recieved

4.2 Frame pipeline

Pipeline is almost like a water pipe - it can only have so much liquid traveling down it at any one time.



Types of multiplexing

TCP: (transport control protocol) Sends through just one frame and sees the acknoledgments when one frame is succesful it doubles the number of frames at one time to detect the throughput maximum. A form of *Adaptive protocol*.

Go-back-N: Only buffer room at reciever for one frame - all frames not the expected next frame are lost.

Selective-repeat: Buffers all things after lost frame to avoid needign to resend all data.

Note: With buffering, the sequence numbers must be remembered well, not just using integers.

4.3 Simplified sattelite broadcast

- Major latency issues
- Only deploying costs, once its there there are no additional costs for extra data
- No conjestion

4.3.1 Conventional chanel allocation

Polling: the process where the computer or controlling device waits for an external device to check for its readiness or state.

Frequency and time division multiplexing The technique by which the total bandwidth available in a communication medium is divided into a series of non-overlapping frequency bands.

FDMA (Frequency Division Multiple Access): Division of the bandiwidth into seperate bands, of which each can be used by a specific sender and reciever (like in CV radios)

TDMA (Time Division Multiple Access): Allows many devices to use the same frequency chanels, but each has a different period of time to use it.

CDMA (Code Division Multiple Access): Used in cell phone communications - uses some set changing frequency and time calculation so the sender and reciever switch frequency and clock in unison to allow for multiple and potentially more difficult to track access.

Mobile phones use frequency hopping, which allows optimization of traffic and error handling.

4.4 Network protocols

4.4.1 Pure ALOHA protocol

Frames are required to have a maximum size and this can help with the period of vulnrability: maximum of two frames at a time.

Preamble 62 bits	SFD 2 bits	Destination 6 bytes	Source 6 bytes	Length 2 bytes	Data 46 - 1500 bytes	CRC 4 bytes

1010...1011

MAC addressing scheme

4.4.2 Slotted aloha protocol

Only transmit at the beginning of time slaces, the beginning of the time slice - this halves the probability of collsision.

4.5 Local Area Networks

4.5.1 Persistant CSMA Protocols

Persistant: How egar nodes are to try again after failing a transmission

Persistance can be thought of as a probability, how long a device should sleep until it can transmit again, Or should it just go ahead and transmit straight after a failiur. There are different levels of persistance.

4.6 Ethernet

Different physical properties are used to hold the ethernet protocol ### Contention algorithms Introduction of randomly allocated slot times help reduce the amount of clashes. Two nodes colliding have a chance that they will still collide, but the ethernet protocol will limit at two collisions and essentially step in to try to fix this issue by making the random number range greater and greater for each collision. This mechanism is adaptive to the network conditions, it essentially detects how busy the wire is.

5 Lecture 5

5.1 Ethernet packets

MAC Address: A 48 bit address, each part with its own importance

Preamble: Up to 62 bits wide, alternating 0s and 1s with a double one as the end delimiter. The length can vary as we are expecting devices to not neccessarily be constantly connected to the data stream.

The packet has a maximum size as:

- To avoid clogging the bandwidth
- To avoid corruption causing larger issues

The packet has a minimum as:

- Whenever someone is transmitting they will saturate the medium.
- Ensures a transmitter can hear a collision that it is involved with

5.2 Packet transport mechanisms

- Carrier detection
- Packet error detection
- Interferance detection
  - Nodes not transmissitting must also be listening
- Truncated packet filtering
- Collision consensus enforcement

5.3 Hubs switches and collision domains

Hubs: Essentially act as all sockets in one line (can easily be listened to by any of the attatched devices).

Switch: Data not sent to every device - is smart enough to see who is who and redirect traffic accordingly.

Collision domains: With a switch there is substantially ess collisions than with a hub.

5.4 Interconnecting

We are mainly interested with dealing with connections between different technologies.

Bridge: Sees the data - converts it - resends it

5.5 Wireless transmission

Notion of the edge of the network. Two problems come up with standardisation:

- A wants to talk to B, but A cant hear that B is talking to C
- B is talking to A, and C cant hear that.

5.5.1 Collision avoidance

This is termed as the hidden node problem. This measures the likelyhood of there being a collision. Regular collision detection wont work in this situation therefore we must move to collision avoidance. Nodes must now announce when the medium is theirs, and they must announce this. A frame will send a **request to send** (**RTC**).

- The request is made to the network

Network allocation vectors essentially say: "This period of time is mine" on some network.

Ad-hoc networking: Direct connection between two devices.

5.5.2 Access point association

Access point: Almost like a switch, but only in one medium.

Probe frame: Asks if there is an access point available.

Similar to frame formatting to enthernet networing, with the variation of variable addressing.

5.5.3 Attacks against wireless transmission

The main issue with wireless transmission is that it can be attacked by anyone within the wireless transmissions proximity, as the attacker does not need to have physical access to the network (like an ethernet port).

Physical layer attacks

- An attacker can saturate the bandwidth by creating some device to create collisions and slow down or completely stop legitimate transmissions

Data link layer attacks

- Antenna diversity can be exploited to make the antenna not want to transmit to the original device
- Spoofing of MAC address to make an illigitimate connection to the network, or to confuse access points

5.6 Wireless encryption

5.6.1 Wireless equivilance protocol (WEP)

A key is generated and sent via the wireless network, but this means that the key is being transmitted and can be picked up by anyone listening on the network.

1. The client sends an authentication request to the Access Point.
2. The Access Point replies with a clear-text challenge.
3. The client encrypts the challenge-text using the configured WEP key and sends it back in another authentication request.
4. The Access Point decrypts the response. If this matches the challenge text, the Access Point sends back a positive reply.

The sender essentially generates a key, and the data to be sent - will XOR the plain text and key together and this will be sent. The reciever will use the recieved data with the already known key (sent previously) and do another XOR to discover the plain text.

This has been superseded by WPA.

5.6.2 WiFi Protected access (WPA2)

Unique encryption keys are generated for each device, and are constantly being changed to ensure no device can be compromised. An integrity check; TKIP (for Temporal Key Integrity Protocol), is used to maintain the integrity of the network.

6 Lecture 6

6.1 Network layer

The network layer is the lowest layer which needs to deal with end to end transmission. The network layer is the third level of the Open Systems Interconnection Model (OSI Model) and the

layer that provides data routing paths for network communication. Data is transferred in the form of packets via logical network paths in an ordered format controlled by the network layer.

Logical connection setup, data forwarding, routing and delivery error reporting are the network layer's primary responsibilities.

We are mainly talking about two types of connection:

- Local area networks
- Wider area networks

Very few reponsibilities on the network layer in the event of a LAN, as there is very little routing.

6.1.1 Responsibilities of the network layer

As with all of the other OSI layers, the network layer only talks to the layer immediately above and below. The network layer for most hosts are only at the edge of the internet. In the event the host does need to make a routing decision:

- The hosts may run at different speeds and will need to adjust accordingly
- Hosts may be routed through other switches

6.1.2 Fragmentation

Fragmentation can occur by:

- More data to be sent than can actually be sent along the physical media

Fragments can arrive at different orders and as such there is a requirement to ake sure they are reconstructed apon arrival.

6.1.3 Header management

Source and destination addresses are the physical destinations of the destination and source - this is the address, and are stored in the header

At this point the packets wont use the whole OSI stack - if the network laying is just passing it along it will not pass it up, it just goes along.

6.2 Modern networking layer scheme

Virtual circuits: Virtual routing between ports - nothing to do with physical cords - more to do with allocating a location for the lifetime of th transmission.

The router will place the appropriate packet in the appropriate porty. **Packet switching** will not preallocate the buffer and then make a decision at the time of sending.

Routing: The process of making a decision of the best route and packet size for the network.

6.2.1 Routing algorithms

Non-Adaptive: termed as static routing, decisions of routing is made before the nodes are brought to life. May use shortest path finding algorithms.

Adaptive: Same decisisons are made as in non-adaptive, however they are made more regularly on some schedule or when it is so required by the network.

Flooding: Sending packets to all nodes - if it is sent to all nodes then it must be sent to the right node - but all irrelevant nodes will be reached.

Distance vector routing: Run periodicially so we know the ongoing optimal route

Count to infinity problem: When dead nodes are believed to still be accessible from connected notes of some source node - it will keep sending around infinitely.

6.2.2 Flow control

In some situations, the metrics are changing; sometimes minimizing the number of hops is most efficient, other times its minimizing the time in the queue.

End-to-end flow control A similar method to virtual ports - requesting the sppace to be sent to the next node.

Load shedding: Essentially discarding packets in an attempt to curb immediate conjestion. This no means the network layer has **Unreliability**.

Leaky bucket protocol: Packets will be put into a buffer but will only come out at a constant rate, regardless of input speed - if there is an overflow in the buffer it will simply dump the buffer.

7 Lecture 9

TCP/IP: Transport control protocol/internet protocol - TCP and IP are two seperate protocols, not one, but generally get grouped together

7.1 History of the internet

Growth in the internet is slowing down. A great number of developments in the internet have been made out of necessity and mostly by the scientific community.

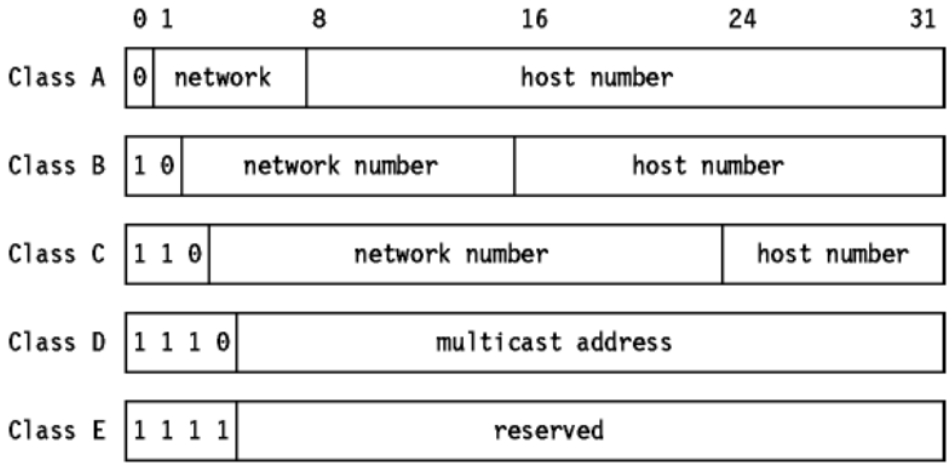
RFC: request for comment, essentially an attempt to publish standardisation.

7.2 TCP layer model

The physical element of the lowest TCP layer is normally left out. The model is as follows:

1. Application
2. Transport
3. Internet
4. Network access

- A network can contain many hosts
- A host can run many processess
- A process can have many concurrent network connections



IP addressing scheme

The TCP model has no concept of the "user", and as such hthis concept gets put into the application layer. The application layer constantly grows in responsibility and purpose.

The most important part of networking is the ways in which the application layer moves the data

7.3 IP version 4

Rollout of IPv6 is taking years, so IPv4 is still prolific. Version 4 uses 32 bit integers, while version 6 uses 128 bit integers.

Addresses are broken up into feilds:

Where:

- Class A: 1.0.0.1 to 126.255.255.254 and supports 16 million hosts on each of 127 networks.
- Class B: 128.1.0.1 to 191.255.255.254 and supports 65,000 hosts on each of 16,000 networks.
- Class C: 192.0.1.1 to 223.255.254.254 and supports 254 hosts on each of 2 million networks.

We can tell the class of an IP as the beginning identifier is delimited by a zero, showing the start of the IP.

Non-Routable addresses: Router addresses within a home/business network. These cannot leave the internal network, and must be mapped by the router to the routable address.

Local host: 127.0.0.1 is used to route directly, back to the same computer.

Classless interdomain-routing (CIDR): An IP is specified with a mask, which tells us where the network and host portion start and begin. This prolonged the inevitable running out of unique IP addresses.

7.4 Mapping IP addresses to physical addresses

Address resolution protocol (ARP): Mapping a logical address (IP) to a physical address (MAC)

1. Source (A) sends out a broadcast asking for where the destination (B) is physicaly
2. All devices on the network hear this broadcast

3. B responds with its physical address back to A
4. All devices on the network hear this response, and will save B's and A's address in some map of the network
5. A will send its message to B

Only MAC addresses can be used for communication, these addresses will be cached by devices on the network once they have been mapped.

ARP snooping: The process of remembering MAC addresses of devices that have already been used on the network

ARP packets are embedded within an ethernet payload

7.5 Configuration of networks

Static configurations are becomining problematic as networks are no longer static - As such it is now necessary to send the network information accross the network itself. This information includes:

- Unique IP
- Client hostname
- Default router
- Subnet Mask
- Domain Name Server
- Time / Timezones

This concept of dynamic netorks brings up reverse-ARPing:

RARP: can be summarised in the - "Here is my physical address, what is my IP address"

7.5.1 BOOTP

- Can be used for network booting
- Can be used to send the kernal at boot
  - Extremely slow, especially when there is additional strain on the network

7.5.2 Dynamic host configuration protocol (DHCP)

Leases an IP to a device wanting to use the network - if the device is still using it, when the lease times out it will be renewed, if not it will lose the lease - when that device reconnects it will need a new lease, and potentially a new IP address if the previous one has already been taken.

7.6 Sending data

TCP: Realiable streaming

UDP: Unreliable datagrams, uses a *best effort approach* - The sender will not be notified of any success or failiure, and as such a failed transmission will not be resent.

Most systems will use their own protocols (as apposed to HTTP, FTP, etc) and will not have an RFC - this will include systems like movie streaming, messaging services and the like.

7.6.1 Internet control message protocol

Unreliable using pings accross the network

Traceroute: Working out the routing rout by sending out a ICMP with a limit of only one hops - when this hop is completed the router that terminates it will send its address back with a notification of failiur. This will then repeat with two hops, three etc. until the destination is found, and will accurately map out the rout required to get from two points

8 Lecture 10

8.1 Ports

TCP/IP allows for ports (16-bit integers) which allows for network connections by processess. This allows for end to end connections.

Data from applications require information to be sent - The TCP/IP system allows for this. The TCP/IP heaers are of fixed size.

8.1.1 Port Numbers

- Ports below 1024 - reserved ports.
- Ports above 1024 - Not reserved but may already be assigned.

We do not want just any process to be able to access the reserved ports as it may allow for malicious entities to gain access to the ports which may be meant for other applications. A port number is used by an operating system to route network connections to the right process

- The ports do not have a specified protocol - It is just an expectation that it will be the correct one when a program connects to this port.

Sequence Numbers: In TCP/IP there are offsets within the TCP segment which are registered by the sequence number.

Window Parameter: Window sizes increase as the pipelining is going well, this is recorded by the window parameter.

The windows start off as sliding window protocols, and adapt as the network conditions change and become known - This is a form of adaptive process, and will increase in window size as the network becomes more free, but will back off when the network shows signs of not holding up.

8.2 TCP/IP

TCP/IP provides:

- Connection Orientation
- Reliable connection startup
- Point-To-Point communication
- Full duplex communication
- Stream interface - Will read and write of any size
- Graceful connection shutdown

	protocol	*local-addr, local-process*	*remote-addr, remote-process*
connection-orient server	`socket()`	`bind()`	`listen(), accept()`
connection-orient client	`socket()`	`connect()`	
connectionless server	`socket()`	`bind()`	`recvfrom()`
connectionless client	`socket()`	`bind()`	`sendto()`

Server Client paradigm

8.2.1 3-Way handshake

A minimum of three messages are required for a connection to begin, this is known as the TCP three way handshake:

1. *A* sends *B* a TCP **SYN**chronize packet
2. *B* recieves **SYN**
3. *B* sends *A* a **SYN**chronize-**ACK**nowledment packet
4. *A* recieves **SYN-ACK**
5. *A* sends *B* an **ACK**nowledgment packet
6. *B* recieves **ACK**

The connection has now been established.

8.2.2 Retransmission

The adaptive protocol and grow and shrink in packet size based on transmission headers.

8.3 Network APIs

A set of services which collectively form an API. The network API is vastly based on the standard file system - Ideas of reading and writing to the network are carried accross from reading and writing to/from disks.

In a network we musut constantly check the permissions related to the end point - This is as apposed to a file system which can normally assumed to be connected once the connection has begun. This is because we cannot just assume a network connection will stay active once we begin a connection: potentially a client will just disconnect from our server.

Socket Descriptor: Shares same namespace as pipes and file systems

Sockets have names - on unix these use pathnames (as with the file storage system).

8.4 Client/Server paradigm

Server: Must exist first - Tells the host OS that "If anything arrives on this port, give it to me".

A new `accept()` will create a new socket connection.

9 Lecture 11

Opening a computer up to the network opens it up to a whole load of potential security flaws. Most security is pushed to the application layer. According to the OSI/ISO model, there are a few main security requirements:

- Data confidentiality: Protect data as it traverses the network
- Data integrity: Ensure the data is the data we are expecting from our sender.
- Data origin authenticator: Verifies the sender is who they say they are.
- Data receiver authenticator: Verifies the reciever is who they say they are.
- Peer entity authentication: Validates the whole layers
- Non-Repudation: Neither sender, nor reciever can deny what they've sent or recieved

Note: The TCP/IP suite does does not implement or require any of these OSI security requirements - These are left up to third party libraries to implement.

9.1 Encryption

It should be assumed that a bad guy can intercept our communications passively and without us knowing - i.e. Eavesdropping. This brings about the idea of encryption. Most encryption methods will use some kind of key.

Symmetric key cryptography: Will use the same key to encrypt and decrypt some data.

Known plain text analysis: Sending known plain text through some encryption to help an attacker understand the encryption system, and potential crack they key.

Security by obscurity: Using the closed source algorithms and techniques to slow down potential hackers solving encryption and security methods - as apposed to open source.

There is no such notion of something that is encrypted, as apposed to something that isn't (both are equal in the eyes of the computer), it is up to the discretion of the interpretation by the application program using the data to make this difference.

9.1.1 DES algorithm

The **Data Encryption Standard** was used by the US government, and was released to the public after some modifications (to make it less secure) was done to the standard. There is also triple DES which allows for a second key holder to engage in the transaction, which effectively creates a multi-party encryption (potentially for multiple parties to have a stake in the encryption). DES is no longer used as it has been compromised.

9.1.2 Diffie-Merkle-Hellman key exchange

In symmetric key cryptography it used to be the case that a key would be sent unencrypted accross the network, leaving it suseptible to some third party who may be snooping on the network - This created the need for some more secure way of sending encrypted symmetric keys.

The Diffie-merkle-hellman key exchange system was created in an attempt to secure the traffic of keys accross a network:

1. A wants to send a key to B
2. A puts the key in a secure box and locks it with A's padlock
3. B does not have A's padlock, so instead: B will add its own padlock and returns it to A

4. A removes A's padlock and returns the box to B
5. B can remove B's padlock now and both parties securly have access to the box's contents (which is normally a key)

9.1.3 Public key cryptography

In public key cryptography there are two keys: A **public key** and a **private key**. The public key is given freely (potentially on the websites homepage or sent directly to a client), and the private key is stored only by the server, and will be considered compromised if this gets leaked and no longer usable. These two keys are related to eachother mathematically - The exchange between two parties; A and B is as follows:

1. A and B openly publish their public keys: E_A, E_B.
2. A sends $E_B(Plain\_Text)$ to B
3. B calculates $D_B(E_B(Plain\_Text)) = Plain\_Text$
4. B can then reply with $E_A(Plain\_Text)$

MIT/RSA Algorithm is one of the more commonly used algorithms for public key cryptography.

Asymmetric key cryptography requires longer keys and greater computational power for the same protection, as apposed to symmetric key cryptography - As such they are used to encrypt symmetric key, which are then used for the rest of the data transfer.

9.2 Digital signatures

There is a need for a way to verify who we are talking to. Someone (an organization) will need to vouch for the validity of the signature - the financial or reputational burden of failing to make a good vouch for a sources character can be used to get a verification organization to ensure the validity of an identity.

Checksums can provide a decent way to figure out if data has been tampered with in transit - a cryptographic checksum gives a number which effectively sums up the entire message for verification - they are not a compression mechanism.

- These checksums are not unique - Its just very difficult to find a duplicate phrase for the sum.

Digital signature: Verifies the validity of the certificate (comparible to a transport ministers signature on a drivers license).

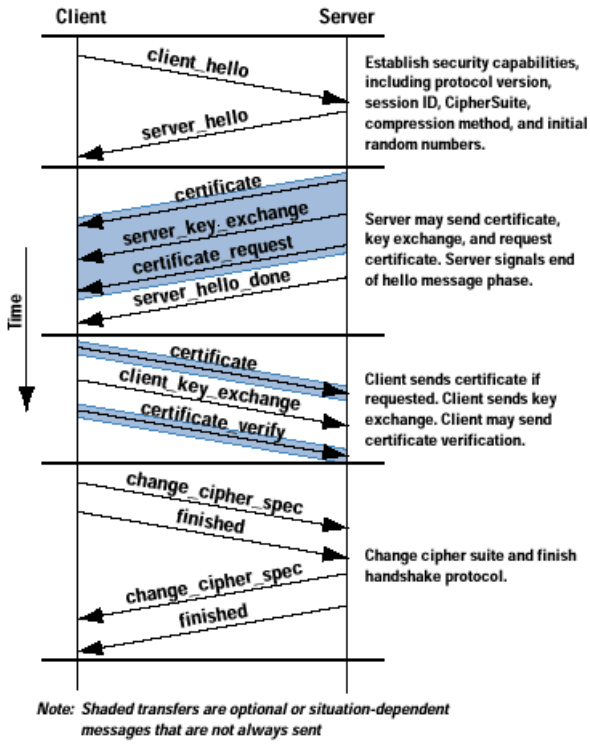
Digital certificate: Binds a public key to some entity or person. (comparible to a drivers license to a person)

Root authority: The end of the chain of trust - the organisation at the very end of a trush chain will have an innate trust - the reputation flows down from this authority.

Why do we trust these root authorities?

- Reputation from good trust over a long time, and investment.
- The knowledge that this trust will be lost if this authority messes up - There is financial losses from losing trust of their clients

Revocation lists: A list of now untrustworthy certificates (these will be valid certificates, but they are now unpaid for, leaked, or expired).



Hello and negotiation parameters

10 Lecture 12

Just because a source is encrypted doesnt mean it can be trusted.

SSL (Secure Socket Language) sits in the application layer, but only when the application layer requires it (it is a protocol that can be used). Some may say that the SSL sits in its own TCP/IP layer, just below application and above the transport layer.

HTTPS: Not its own protocol, but rather it is just the **HTTP** protocol wrapped in a secured SSL socket.

We will normally append or prepend some additional information to verify the source - potentially this can be some kind of signature.

Cypher-Suite: A combination of different algorithms which we can use together.

10.1 Hello and negotiation parameters

10.2 The SSL Layer

The application may ask the TCP layer: "How big would you like the message to be", and this will determine how big the application will send the message, to ensure that the SSL socket can correctly wrap the information without it being broken up in an incomprehensible way. SSL is paranoid about the timing: The data may have been compromised and intercepted if there is sufficient time between the send and response - this does not merge well with TCP as it is eagar to send things sequentially and with backing off. Now the application layer requires the TCP layer to individual messages to be sent once, in tact, and in a timely fashion.

Note: We encrypt the information after it has been compressed. This is as compression requires some knowledge of the data, which is lost after encryption. This process of encryption effectively removes much of the compression capability (for example the encryption process will make many of the same character come out seemingly random - This does not work well for encryption algorithms such as h264).

10.2.1 Man in the middle attacks

The intercepting party acts as a reciever from the server, will decrypt the information, and then act as a server to the original client - neither the server nore the client can tell of this attack (other than potentially from some kind of lag). This relates back to the point that: Just because we have a certificate doesnt mean it is authoritative.

10.2.2 Using SSL

We can use protocol tunnelling, such as stunnelling to avoid changing the code of legacy apps to include SSL.

Most certificates regularly used are already trusted by the operating system, and/or the browser - There is often no need to check the chain of trust to find the root certificate in this instance (as this may potentially be offline at the time, or will be many layers up, wasting large amounts of time and resources to track it down).

10.3 Client/Server paradigm

10.3.1 Client

Normally the piece of software that interfaces with the human - It makes the connection and has no shared namespace with the server (these will be different computers, typically in different areas, premises, or companies). This seperation is important for the integrity of the system.

10.3.2 Two-Tiered architecture

In a two-tiered system: The client (normally a web-browser) will be in direct contact with the server which is serving the information they require.

- This is scaled by simply adding more clients
- Apparent concurrency - Potentially causes problems as the clients increase
- Clients must wait for other clients to finish using the server until they can get in
- Does not scale well for many services

10.3.3 Three-Tiered architecture

- Server the clients connect to simply routes the client to a backend server.
- The font end server essentially deals with load balancing
- Front end server might need to edal with query verification
- This form of system will scale well especially when there are many clients, or many backend services being offered

Intranet: Protocols inside a company

10.3.4 Partitioning client/server responsibilities

With networking we are able to share the responsibilities of a monolithic program to different hardware and software systems - The following considerations must be taken into account:

- Is there a functional partition
- Is the data partitioned
- Is there any extensive use of global variables
- Are there any hidden communication mechanisms

From these points it becomes obvious that distributed applications are far more complex than they first appear

10.4 Concurrency in servers

10.4.1 Iterative Servers

Each client is handled until its completion by the server. This method can cause problems when there are too many clients wanting concurrent connections. This is good when servicing requests that are too quick, or the services are long enough to send a placeholder (i.e. some kind of loading screen). This type of server simply blocks until the processor is free to be used for serving the request.

10.4.2 Concurrent servers

Each client is handled by a seperate fork (child process) by the server, effectively allowing for multiple clients to be handled concurrently. This will avoid time outs, and can potentially get more clients through at one time.

11 Lecture 13

11.1 Vulnerabilities in the TCP/IP Suite

Its easiest to determine vulnerabilities by breaking the stack into small discrete sections. As the application is the only place we can add new stuff, this is where most new requirements (such as user authentication through usernames and passwords) is added.

11.1.1 Packet sniffing

Allowing a network access card to hear all traffic on the network using a **promiscuous mode** is called packet sniffing. Packet sniffing does not work with switched networks, but physical media, such as wifi, can have this occur. Packet sniffing and logging can have a legitimate reason; in situations where the network traffic must be logged.

11.1.2 Port scanning

An attacker can check what ports are currently open by scanning, and as such can coninect to these ports and do some malicious activity.

Attackers can hide their attacks by:

- Asking for ports slowly
- Asking for ports in a random order
- Doing these requests from hundreds of IP addresses simulatenously

An attacker can use stealth scanning in which the attacker searches for connections without trying to connect to it. This is done by sending a set of different TCP flags or invalid flags, and seeing how the router responds - This can tell the attacker what router and what version of router firmware they are dealing with - this can be used to launch a custom tailored attack based on known vulnerabilities.

11.1.3 IP Spoofing

We can attack by spoofing an IP address for services that use IP addresses as a crude session identifier - all we would need to do is spoof our IP and our device is now connected to the service under that users session ID, without any authentication.

11.1.4 UDP packet spoofing

Attacker will try to send off unwanted UDP packets to attempt to gain access to a file system.

11.1.5 TCP sequence number attacks

If an attacker can guess the next sequence number in some transmission (note that sequence numbers will normally not be sequential, but rather apparently random) then the attacker can sneak their packet into the stream - This is called session hijacking:

- The attacker can inject the wrong data to the reciever
- The attacker can recieve data from the host (by asking for a resend of the sequence number packet)

11.1.6 Denial of service attacks (DOS)

A DOS attack is defined by an attackers attempt to stop services accross a network, or atleast slow them down.

Smurf DOS attack To essentually amplify the attackers messages through hosts reponding to this until the network is overwhelmed - these messages may be some kind of ARP request which will never be satisfied, with every node in the network essentially saying "I dont know what youre asking for" to eachother until the network is saturated.

SYN Flood attack

1. The attacker sends a SYN request to the server
2. Server will reply with a SYN-ACK and waits for the attacker to reply
3. Attacker replies with a SYN instead of an ACK, in the hundreds, overwhelming the server

This brings about the question: How can a server know what a valid or invalid packet/request looks like? All we can use is history, and known malicious patterns.

11.1.7 DDOS Attacks

Using a whole bunch of other computers to conduct an attach in sync. It is available to a "mainstream" attacker simply by paying for hacked PCs to all conduct a synchronised directed denial of service attack - Using this method it is more often than not impossible to determine the root of the attack.

11.2 Security

At the network boundaries it is important to snoop on all the traffic to determine if we wish to allow the traffic through. The following are normally considerations that a **firewall** must take into account:

- IP Address
- IP Ports
- Not exposing LAN topology to the wider internet
- Constraining our network traffic based on content
- Only permitting internal access to remote users based on their verified identities or locations
- Logging all internet connections

11.2.1 Packet filtering

A firewall mwill make the decisions for if a packet will be allowed, or if it should be blocked from entering an internal network.

IPTables: Used to specify the sets rules that we wish for the firewall to block

Filter Rules: Specific rules which we want the firewall to block

If a rule is matched, then we need to figure out what to do with it. Dropping a packet can be done to simply forget about a packet, without notifying a sender that the packet has been discared, but there are also rules which alert the sender of this. We can also load a module for when a certain rule is triggered.

11.2.2 IP Masquerading

Network Address Translation (NAT) is used by routers and firewalls to map a set of IP address to that of the gateway on the wider network. Essentially the firewall just allows the device trying to make a connection to the wider network to use the IP address of the gateway - avoids using the internal, unroutable address. This wider network IP address is the only address which the ISPs will allow to make connections on the internet.

Stateful Connection: Looking at a series of packets and determining if they are in the correct order.

11.2.3 Intraorganizational firewalls

An organization may hve multiple firewalls at different security levels and areas of a network/virtual network.

DMZ (demilitarized zone): Contains the systems and services which will be exposed to the wider internet, from inside an office. Devices in the DMZ will often have read-only hardware, and limited access to more internal elements of the business.

11.2.4 Virtual private networks (VPN)

A VPN will ensure that communications between users who may be seperated by unsafe public networks will remain secure - This may be two seperate business offices accross the world, wanting to connect to the same network infrastructure.

Each VPN gateway employs tunnelling:

- Authentication
- Confidentiality
- Integrity
- Protection from insertion and replay.