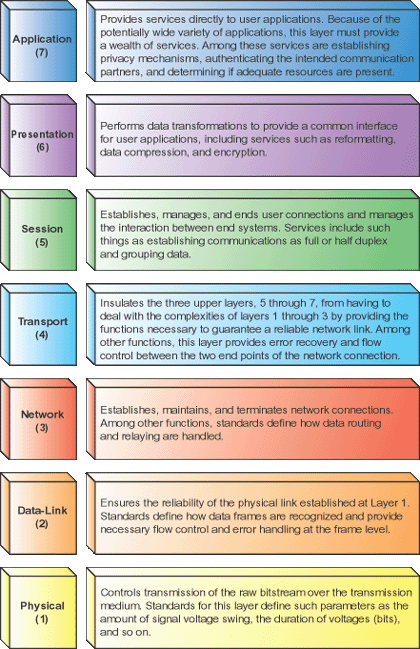
***Networking and Data Communications***

******

***The Application layer***

is the entrance point that programs use to access the OSI model and utilize network resources. Most application layer protocols provide services that programs use to access the network.  
  
***The Presentation layer***

There is only one function found at the presentation layer, and that is the translation of syntax between different systems.  
  
***Session layer***

functions are integrated into other protocols that include presentation and application layer functions.  
  
***The Transport and network layer***

protocols used to transmit data are often thought of as a matched pair. These include the TCP, which runs at the transport layer, plus the IP, which runs at the network layer.

***Network layer***

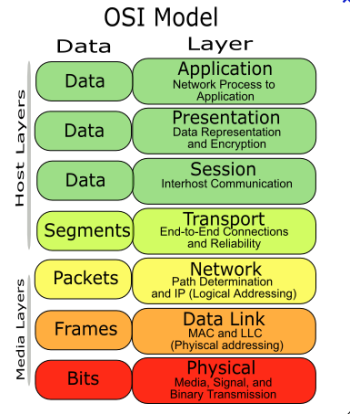
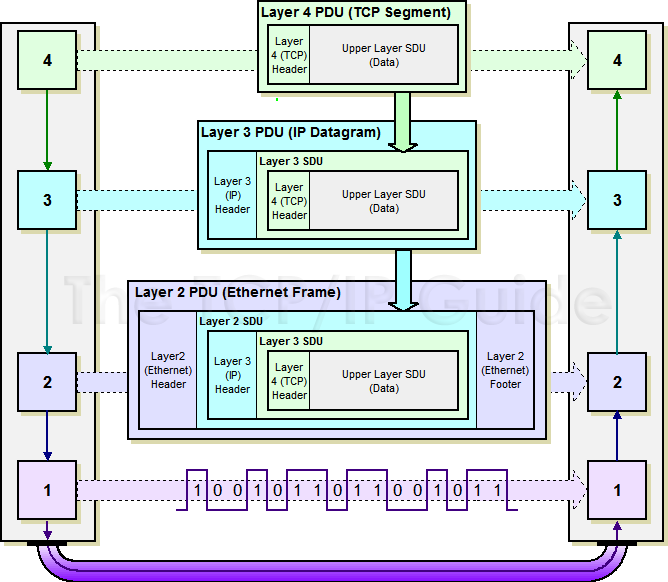
maintains and terminates network connections allowing for the passage of data.  
  
***The Data-link layer***

is the conduit between the computer's networking hardware and its networking software.

**The Physical layer**

defines the nature of the network's hardware elements, such as what medium the network uses, how the network is installed, and the nature of the signals used to transmit binary data over the network

**SIRM Model**

**Layers** *Please Dear, Not Tonight, Soft Penis Already* – Separate layers of abstraction

**Data** *DDD, Slow Posts, Face Book*

* International standardisation of protocols

Minimises data flow and number of layers, without forcing separate

**Revision Notes 2012**

**Physical – Layer 1**

* The data unit in layer 1 is the bit of data.
* Access and control of the physical medium.
* Protocol **defines physical standards** for transferring physical bits of information.

**Bandwidth Limit**

* Media has limited bandwidth( frequency range)
* Signal will be impaired or distorted dependent on the signal frequency & range of the medium.

**Media – Types of Connections**

* **Cable** – Ethernet and Coxial
* **Optical** Fiber
* **Radio Waves**
* **Terrestriial Microwaves**
* **Satellities**

**Cable**

* **Co-axial Cable** Central copper core with wire braiding for protection
* **Twisted Pair Cable** Two insulated wires twisted together in pair & *Low cost/easy to handle.*

**Optical Fibre**

* **Repeats** used to **retrieve** the **weakened** or **distorted** **signal** and **transmits a clean copy**.
* Electronic **switches /amplifiers cause bottle necks**
* Optical fibre is **ideal for** transporting **digital data.**
* Undersea cables use Forward error correction to assist in resolving error

**Terrestrial Microwave Links**

* Spaced 10 miles apart with **direct line of sight**
* **Point to point network** of repeater stations.
* *Nation-wide/continent-wide network*

**Satellites**

* **Leo/Meo** – appear to move relative to use and have been used for *mobile communications*
* **Heo** – *Highly elliptically orbit* – complicated orbits but provide a larger footprint/ closer to the poles
* **Geo** – geo stationary – Synchronized with the orbit of earth and appear stationary

**Network topology**

* **Star** – local exchange in telephone system
* **Ring** – different media used to connect points
* Fully connect – all nodes are connected
* Partially connected- some nodes must take routes

**Circuit Switch & Packet Switching**

* **CS** - Complete circuit from transmitter to receive

*e.g. analogue telephone for duration of call*

* **PS** – **Segmented** into packets taking **multiple** **routes** to destination. *Can use Store and Forward if a path until channel space is available.*

**Frequency Division Multiplexing**

* Divide large broadband **channel into a number of smaller sub-channels**.
* **Voice transmission** Each channel is filtered down to 3 kHz with a 0.5 kHz filter on each side then shifted up the frequency spectrum to its own unique frequency band. 12 of these is a group

**Time division Multiplexing**

* Only one transmitter can transmit at a time.
* Demultiplexing switch is synchronized with multiplexing switch at the transmitting end of the channel. Connects each receiver to the channel.
* **Round robbing** basis.

**PCM - Pulse Code Modulation**

**Analogue to digital conversion** of a signal

* **Sampling** 8khz samples with a 3khz cut off using codex at telephone exchange to next exchange and then decodes back to analogue to receiver
* Digital switch and signalling gives **faster set up** at the start of call and **better signal – noise ration**.

**Shannon’s result**

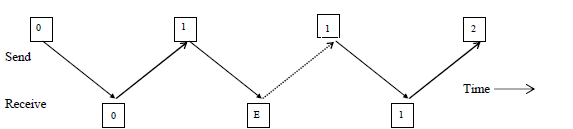
**Real channels are noisy & errors prone**, as noise acts to distort and corrupt the signal being sent.

* Maximum data rate for **error-free transmission** **depends on signal/noise ratio**

**Data Link – Layer 2 - Description**

Achieve the error free transfer of packets between adjacent nodes in a network.

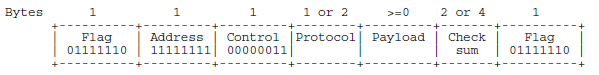
* Packages **bits into frames**. Bits-in/out.
* Process **acknowledgement frames** from receiver
* **Manages resources** and performs **flow control**.
* **Noise** **causes errors** on the channel
* Physical layer appear free of errors to layer 3.
* Uses Store and Forward mechanism

**Point to Point Protocol – Packet Layout**

PPP is a data link layer protocol developed for use in the internet. Connecting WANs together.

**PPP provides**

* **framing method** delineating frame start/end
* **Error Detection**
* A control protocol called Link **Control Protoc**ol for bringing *lines up, testing lines, negotiating options with nodes, and bringing lines down*
* **Interface and negotiating facilities** for working with different network layer protocols.

****

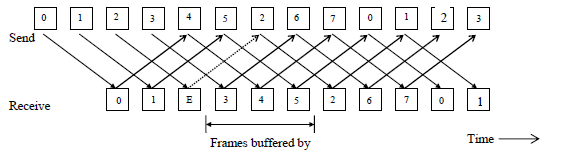
* **Flag** to “wake up” the receiver, mark packet start
* **Address :** Destinationreceives packet. broadcast
* **Control** : Default is un numbered frame. PPP usually does not provide a strictly reliable service because sequence numbers and acknowledgements are not used by default.
* **Protocol** : Indicates what type of packet is in the payload field e.g. IP
* **Payload**: carries the data. Default size but this can be negotiated during line set up
* **Checksum** is for error detection**.**

**Character Stuffing:** an extra byte code added into character steam to break up a sequence which would otherwise be interpenetrated as a delimiter sequence.

**Bitt Stuffing :** Adding extra zero onto end of file market to avoid terminating a sequence early.

**PAR: Positive acknowledgement & Retransmission**

* Simple protocol for a noisy channel
* **Returns dummy frame** with positive acknowledge
* **No acknowledgement** if errors. **Times out** instead
* Next frame is sent only when acknowledged

**Frame contains** : **Type of frame**(*Data or Ackt),* **Sequence Number. Information***( if* ***data*** *frame)*

**ARQ: Automatic Repeat Request**

Automatically requests the repeat transmission of a block of data if it detects an error using a frame.

**Types**: *Stop & Wait, Go-Back N, Selective Repeat*

**NAK: Negative acknowledgement**

* Detecting a **corrupt frame sends a negative acknowledgement**. *Similar to ARQ*

**Piggy-backing of acknowledgements**

* Piggy backs onto real data frames
* **Space** is **reserved** **for acknowledgemen**t
* No real data to be sent in the opposite direction, send empty frame with the indicator present
* Full duplex – **Simultaneous bi-directional flow**

**Flow Control**

**Control the flow of data packets** between the transmitter and the receiver by adjusting the output of packets according to the rate of return.

* Aims to send data **as fast as possible**
* Needs **Error detection/correction** and **reordering**
* Acknowledgements limit packets sent so receiver/network **isn’t overloaded/ congested**.

**Sliding Window Protocols**

Effective means of **performing flow** control within a packet switching network**, including error control** if a error protection method (checksum) is available.

**Stop and Wait**

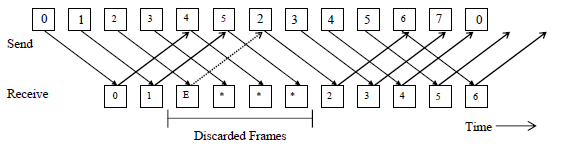
* Sends a single frame then waits
* Highly inefficient in communication channel

**Pipelining**

* **Send frames continuously**, and maintain a window long enough to allow **sufficient time to acknowledge** the sender.
* Window essential in **full use of channel capacity.**

**Go Back N**

* **Pipelines all frames** until an error is detected
* Corrupt frame cause frame and *all subsequent frames to be retransmitted*
* Stores 1 frame at a time– *Windows size of 1*
* **Saves memory but wastes Bandwidth**

**Selective Repeat**

* No unnecessary retransmission but requires more memory. Also sequences in memory
* **Saves bandwidth but costs memory**

**Efficiency of a channel Utilisation**

**Channel Capacity =** bits / (seconds/2)

e.g. 1800 / 50000 \* 0.5

**Frame size =** l bits

**Network – Layer 3**

* Data into packets and routes them within network
* Network interfacing must be done here as host does not see the network topology.
* End to End communication only

**Virtual Circuit**

* **Error free** perfect channel. All packets arrive in **correct order** and **no lost packets**
* Peer Processes only communicate directly in L.1
* Communicate only occurs 1 layer up or down.
* Layers protected from problems above.

**Datagram – Accepted from Transport Layer**

* **Not error free** & May be **delivered** **in any order**
* **Greater efficiency than Virtual circuit**

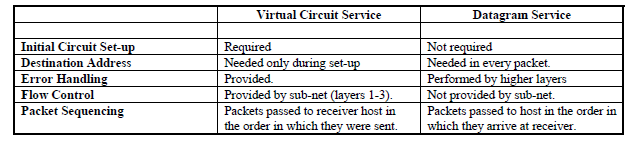
**Connection Oriented Service**

* A l**ogical connection** rather than a physical one
* Holds records of existence of connection and packets are sent between them
* Aware of a data transaction with each other.

**Connection-Less Service**

* **Unreliable service** and **no acknowledgement**
* Each message **carries a full destination address** and **routed independently** relying on routers to forward it to the correct destination.

**Connection Oriented Vs Connection Less**



* If the sub-net (*layers 1-3)* provides a datagram service, **error control** and **packet re-transmissio**n must be done by ***layer 4****.*
* **TCP** reliable connection oriented service.
* Corrects errors which might happen under IP.

**Routing**

* Determining the **optimal route** to get a packet to its destination **quickly** and **cost –effectively.**
* For VC’s route is **established** once at the **start of session***. E.g. Session Routing*

***Non-Adaptive Routing Algorithms***

* Fixed policy, routed to pre-fixed methodology.
* Suited towards non changing networks and non dynamic traffic conditions

**Flooding – Non Adaptive**

* Rebroadcast packet to all lines except incoming
* **Very robust** /*damaged network*/ **Low Efficiency**
* **Selective flooding** *– Aprox sends in right direction*

***Adaptive Routing Algorithms***

* changes route based on how heavy the traffic is , how busy a particular node is
* Optimise efficiency as traffic conditions change and network structures change

***Hot Potato algorithm – Adaptive Routing***

* Packet is placed in the shortest output route as soon as it is received. No matter where it goes.
* Achieves equal traffic on all links. *circle forever*
* Routing table biases packets toward correct route

**Optimality Principle**

* Set of optimal routes from node to all other nodes will form a tree with node x as the root node.
* Tree is known as sink tree for node x.
* Find and use sink trees for all routers in a network

**Shortest Path First – Dijkstra’s algorithm**

* Each node has its cost calculated such as bandwidth, traffic, delay, etc to all other notes
* Each node has a table to find the best route.

**Link State Routing – Uses Flooding**

*Preferred method for the internet for several years*

1. Find neighbouring routes and network address
2. Measure transmission delay to each neighbour
3. Construct a packet containing information gained
4. Send the packet to every router in the network
5. Use this information to calculate the shortest path

* **Finding Neighbours** Sends out Hello Packet. P-to-P
* **Line cost** Sends echo, gets reply from neighbours immediately and measure round trip time.

**Link State Packets**

* Packets has routing table for connecting nodes
* Sequence number is updated each time there is an update. Old packets are checked and killed.
* Age counter is decremented for each router so it does not live forever.

**Hierarchical Routing**

Packets contains information about topology of the network for that region.

Packets redirected to another region.

*E.g. International routing, country to country*

***\*Dijkstra’s Algorithm used here***

**ICMP – Internet Control Protocol**

Report starange events on the internet and test network.Echo and timestamp used to measure delays.

* Time exceeded : packet killed as counter hit zero
* Echo Request : Asks machine to send echo
* Echo Reply: responds to echo request
* Timestamp request : echo with timestamp
* Timestamp reply : echo reply with timestamp

**ARP – Address Resolution Protocol**

Broadcasts message to find the mac address with an ip address. Workstation replies and transmits on lan

**RARP – Reverse Address Resolution Protocol**

Broadcasts to find IP address using its Mac address

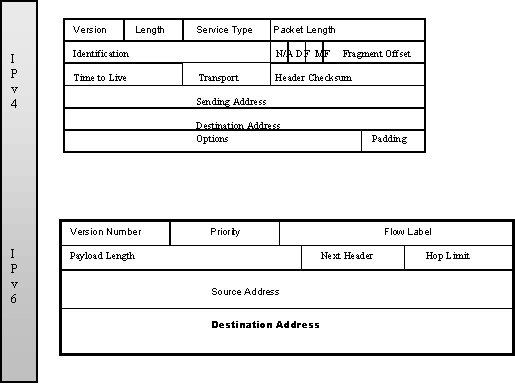
**OSPF –Open shortest path first**

* Interior Gateway Routing - Routing within individual organisation Autonomous systems.
* Exterior gateway protocol routes between separate ASes.

**IPV4 Datagram Header**

* **Version** current version of IP
* **IP Header Length** number of 32 -bit words. 5- 6
* **Total Length** Size of Datagram (in bytes, this is the combined length of the header and the data)
* **Time To Live** Number of hops /links which the packet may be routed over, decremented by most [routers](http://www.erg.abdn.ac.uk/%7Egorry/eg3561/inet-pages/router.html) - used to prevent accidental routing loops)
* **Protocol** indicates the type of transport packet being carried (6 = [TCP](http://www.erg.abdn.ac.uk/%7Egorry/eg3561/inet-pages/tcp.html); 17= [UDP](http://www.erg.abdn.ac.uk/%7Egorry/eg3561/inet-pages/udp.html)).
* **Source** original sender
* **Destination** final destination

**IPV4 & IPV6**

**IPV6 Datagram Header**

* **Version**- Internet Protocol version number = 6
* **Priority** - Priority value. Indentifies the packet as either a congestion controlled or no congestion controlled. No congestion controlled has priority over congestion controlled packets.
* **Flow Label**- Provide packet flow control. Further discussed in the Quality of Service (QoS) section
* **Play load length** - Length of payload, i.e. the rest of the packet following the IPv6 header, in octets.
* **Next Header** - Indentifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field. It also describes any IPv6 Extension headers that follow.
* **Hop Limit**- Suicide pull
* **Source**- The address of the sender of the packet
* **Destination**- recipient of the packet

**Advantages**

* Huge address space and better security than ipv4
* Reduces size of routing tables and allows mobility whilst keeping the same IP address

**Provider Based Addressing – IPV6**

Five bits after the prefix identify the registry in which to look up the provider.

**Address Format**

* Written as eight groups of four hexadecimal digits with colons between groups.
* Groups of zeros can be omitted and 16 zeros can be replaced with ::

**Extension Headers**

* Optional extensions for routing, authentication, fragmentation, destination so routers less to sort.
* **Jumbo grams** : used for transferring large files across the internet e.g. GB’s. Typically 64kb or greater

**Transport – Layer 4**

Two computers are communicating directly at this layer using the subnet layers 1-3 to deliver packets.

* Creates a **multi route network** if high throughput is needed **by the session layer**.
* **Establish** and **delete** **connections** across network
* Use connections to transfer messages and files
* TCP can operate over several networks.

**Comparing Layers 2 and 4**

* **Transport layer is global** – intern-connected networks and cross network traffic.
* Explicit addressing for each node is needed. Talks to all nodes in subnet
* **Data link is node to node**, node at end of cable.

**Flow Control & Buffering - Bandwidth**

* *Low* – **buffer at sender** – *easier retransmission*
* *High*– **buffer at receiver** – *full use of channel*

**Dynamic buffer allocation**

* **Variable size window**. Allocates memory available
* Sender **stops sending** when it fills, receiver **piggy backs buffer allocation** and **acknowledgements** back to sender. Sender **adjusts packet sizes**.

**Packet Delay – Sliding Windows Protocol**

Sub net can store packet or roam around network. If sender times out new packet is sent. Sequence algorithms needed to sort out packet sequence.

**Synchronisation:** Time stamp or hop counter used

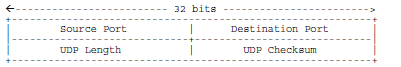
* **Time stamp** requires a global time source. Radio or satellite transmitted time signal
* **Hop counter**: life prolonged if stored in node

**UDP - User Datagram Protocol**

* Connectionless transport protocol gives application layer access to IP datagram service

**Advantages**

* Client server application in which the client gives one request and expects a single reply uses UDP.
* Useful when speed of pack delivery is essential but quality and reliability of service are not.



**TCP – Transmission Control Protocol**

* Deals with **loss of packets** and **delayed duplicates**
* Copes **un-reliable datagram network**
* **Connection Oriented –** Established connection
* **Times out** and **retransmits** if necessary
* **Reorders** datagram’s at destination

**TCP Service**

* **Connections** identified by *two socket numbers*
* **Socket Number** = address of host **(IP)** and (**Port)**
* Port is the **Service Access Point** provides *application layer with access to the session*
* **Port numbers** below 256 are reserved for standard services. well known services.

*Port 21 is FTP, 23 is Telnet, 80 is HTTP*

* A TCP connection is **full duplex** and **point to point**, byte oriented. Data is a stream of bytes.

**TCP Protocol - Segments**

Transport Address (uses **Sockets)** each process a unique address and can listen to incoming requests.

* TCP packet is a **segment** of data *(including header)*
* Must fit IP’s *64kbyte payload*.
* **Maximum Transfer Unit** few thousand bytes
* **Segments fragment** when too large.
* **Unnecessary packets** when too small

TCP uses ***windows sliding protocol***

1. Sender times and waits for acknowledgement
2. Receiver ack & asks for next expected segment
3. Not received in time, sender sends another copy.

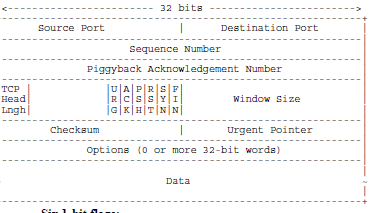
**TCP Header**

* **Source/ Destination**– Identifies 2 user processes
* **Sequence and piggyback number** – 32 bit
* **TCP Header Length** – size of header

**Six 1 bit flats**

* **URG –** Urgent
* **ACK –** acknowledgment
* **RST –** reset connection
* **SYN –** establish connection
* **FIN –** end connection
* **PSH –** indicates pushed data

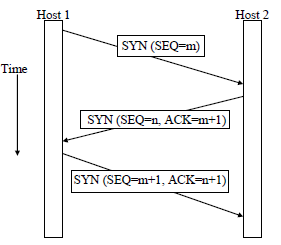
(*PSH – delivery ASAP and not wait until buffer is full)*

**

* ***Window size*** *– How long sliding window can be*
* ***Check sum*** *– for error detection*
* ***Urgent pointers*** *–offset from current sequence no*

**TCP Connection Management – 3 way handshake**

Prevents problems caused by **delayed duplicate** connection requests and **acknowledgements appearing out of the sub net** when not wanted.



* Client in the sender sends **connection request**
* Server in the receiver responds with an **acknowledgement to accept** the connection
* Sender responds with **another acknowledgement**.

**Connection Release**

Connection is release in two stages. Both connections

* Either host can send a segment with **fin bit set**.
* Other host should **acknowledge** close connection.
* Sender will close connection within **2 segment life cycles** if no acknowledgement is received.

**TCP Congestion Control**

* **Dynamically** **changes the size of its window** in order to manage flow of packets into the network
* Sender maintains two windows. One specified by **receiver in its acknowledgements**, and the other as the **congestion window**.
* Maximum number of bytes specified **by smallest**.
* **Threshold** - Sender modifies **congestion window** based on **time outs**. Continuingly increases window until **time out occurs and then reduces it**.

**TCP Time Management**

* A **re-transmission timer** is **started** when a **segment** is **sent**. If **times out**, then **retransmitted**.
* **Round Trip Time** (*RTT*) is **current best estimate**.
* RTT is no updated in most TCP implementations **retransmitted** **after** a **time** **out** because 2 acknowledges for the segment may return.
* **Time out** is **doubled** when **naturally** times out.

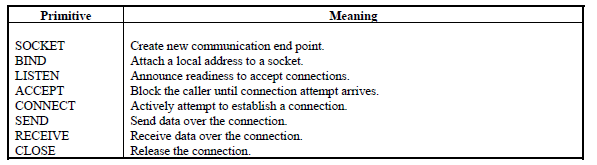
**Other Timers in TCP Implementations**

* **Persistence time** – **prevents** **deadlock**. If server sends window =0, then packet is lost with updated window size, it will **probe to get communication back**.
* **Keep-Alive Timer** – if idle for a long time**, checks connection**. If no response then terminated.
* **Time Wait** – Connection closed to make sure all **packets** and **acknowledgements** are **dead** before **connection closed.**

**Sockets**

**End points** in the Transport layer **for** the **connection** which will be **set** by a **connection oriented protocol**.

**Establish** a **new** **record** **recording** the **existence** **of** a **new** **communication** **end point** and gives it table space in memory.



**Session – Layer 5**

* User interface to the network
* **Binding** – Setting up a session between two users
* May **retrieve communication**, or **repair damage** after a failure in a lower layer.
* **Order out of sync messages** from transport layer and groups them before the user receives them.

**Presentation – Layer 6**

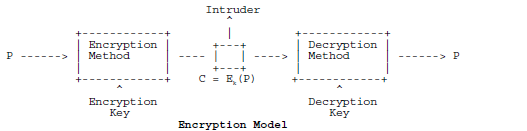
* **Text compression** for efficiency of storage and communication of data, encryption for security
* **File Format:** convert file type that use different storage, headers, terminators etc.
* **Terminals**: Conversion for different standards.
* **F.Point conversion**: Represent data differently

**Security and Privacy**

There are four main needs for data security

* **Secrecy**- keeping sensitive data secret
* **Authentication** – checking that you are communicate with the right person
* **Non-repudiation** – providing that someone was the source of an order or message
* **Integrity control** – preventing malicious tampering with orders or messages.

**Basic Encryption Model**



* **P = Plain Text**, the secret message or Data
* **C = CipherText**, the ecrypted data which the intruder can observe( also called cryptogram)
* **K = encryption key**, which need not be the same as, or a simple inversionof, the decryption key.
* **Intruder** may be passive or active – listening to data (**passive**) or may alter it( **active**).

**Substitution and Transposition**

**Substitution** – replace every character with another

**Transposition** – move characters around within data

**D.E.S – Data Encryption Standard (S&T)**

* **Difficult to decipher** even if a large amount of cipher text is available.
* Uses **large amount of transposition** and **substitution**, repeated several times over.
* DES is a **giant mono alphabetic substitution cipher** on a 64 bit character.
* **Vulnerable to attacks**, especially if key is shorter

**Cipher Block Chaining (CBC) (S&T)**

DES can be improved using CBC as the previous block is **exclusive ORed** before **putting** it thought **DES** **encryption**. *Same block, different encryption*

***Triple Encryption***

Triple encryption uses two DES keys and three stages

* Stage 1 : DES encryption is done with key 1
* Stage 2 : decryption is done using key 2
* Stage 3 : DES encryption is done with key 1

Doubles length to 112 bits while retaining compatibility with the conventional single encryption systems (key 1 = key 2), then triple is same as single.

*A.E.S has been developed to replace D.E.S*

***Public Key Cryptography***

Public Key - To establish communication between two people *without first prior meeting to swap keys.*

*Decryption key is different from public key*

***E = Encryption , D = Decryption, P = Plain Text***

* The receiver of messages will calculate encryption and decryption keys which turns

***Plain text -> cipher text - >plain text***

* **Receiver** then **sends encryption key** to sender.

*Intruder may know the encryption key E.*

* The Encryption is so robust that the intruder cannot break the cipher even with possession of the key and cipher text.

*The decryption key is needed to decrypt C.*

1. The Plaintext can be decrypted from cipher text, E(P), only if one knows D : **D(E(P)) = P**
2. **D cannot be derived from E**
3. Cipher cannot be broken even though the intruder knows how the plaintext converts into cipher text

**Secrecy**: Both Protect data when it is to be stored or transmitted over a communications network

**Authentication**

Authenticating is a process of verifying that a communication partner is who they claim to be.

* One host **sends an encrypted random number** to a host, host 2 **decrypts it** and **sends it back to host** 1. This is **then repeated** vice versa.

**Digital Signatures**

Large businesses need to verfity digital signatures

1. *Big Brother knows all* ***secret codes and keys****.*
2. *Alice wants to* ***send a “signed” message*** *to Bob****,***
3. *she sends an* ***encrypted string to Big Brother*** *containing Bob’s name, a random number, a time stamp, and the message itself.*
4. *Big Brother* ***decrypts the string*** *and* ***re-encrypts*** *Alice’s name, her time-stamp and her message with his own encryption key, KBB,* ***to produce the digital signature.***
5. *This signature is then* ***packaged into another string*** *with Alice’s name, her random number, her time-stamp and her message, and is* ***encrypted with a key KB,*** *which Bob knows.*
6. *It is then sent to Bob. Bob can therefore* ***decrypt all the information*** *that he needs from Alice, but he* ***cannot decrypt the signature*** *which Big Brother encrypted with KBB.*

**Pretty Good Privacy (PGP)**

PGP is an email security package which provices privacy, authentication, digital signatures, and text compression. It makes use of RSA public key algorithm, block encryption codes, and other techniques to offer security for email.

**Secure Socket Layer (SSL)**

* Can intercede between Transport layer (TCP) and Application Layer (HTTP) to provide security protection for internet transactions.
* *Includes negotiation between client and server,* **mutual authentication** of both, **secret communication** and **data integrity** **protection**
* **Encryption heavily involved** in **SSL** & **hash coding.**

**Data Compression and Text Compression**

* Reduce the communication load by reducing the number of bits in data transmission.
* Compress data into a smaller number of bits

**Relative Frequency of Occurrence**

Assign the shortest codes to the most common symbols e.g. as in Huffman Coding

**Context Dependent Encoding**

When the probability of symbol occurrence is dependent on the preceding symbols

* e.g. charachters occurrence in English words.
* *Probability of U after Q is 1 but x after z low*

**Predictive modelling**

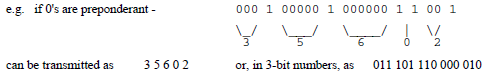
* List of n grams can be maintained with a probability for each n gram.
* Used to predict with good accuracy, the characters which are most likely to come next in the text stream
* Identical models can be used at the transmitter and receiver to predict the next character.
* The transmitter only needs to transmit information to confirm or correct the prediction which the model makes.
* There is a net reduction in the information which has to be transmitted, and hence data compression.

***Markov Model***

* *Using either markov model or n gram tables to predict the next character.*
* *Transmitter then sends a single bit to indicate whether the model has produced the correct character. If the character is wrong, the transmitter may invite the receiver to make another prediction, or send the correct character.*
* *Models may be adaptive to learn from transmitter text, therefore optimising their performance.*

***Run Length Coding***

*If long strings of 0/1’s in the data, re-code the data in terms of the length of the strings of each symbol*

**

**Picture and Video Compression (JPEG/MPEG)**

* Eye more sensitive to brignness then color so sample colour at a lower spatial frequency and use fewer bits to represent the color information
* Eye more sensitive to low spatial frequency. Use more bits for low frequency components

**JPEG – Joint photographic Experts Group**

* **Lossy compression** for still photos
* A picture is divided into blocks of 8 by 8 pixel blocks and transformed into an array of frequency components ( using **Discrete Cosine Transformation)(DCT)**
* **Quantisation** – higher and lower frequencies can be allocated more or less bits as needed
* **Compressed** using Huffman coding, which is a process of allocation shortest codes, to most common symbols, and longer code to others.

**MPEG Moving Pictures Expert Group**

* For compression of full motion pictures on digital store media
* Frequency transform coding and predictive coding are combined
* Predictive coding – a block of pixels in the current frame is predicted from the previous frame. The prediction is subtracted from the real frame, producing a prediction “error” this error signal is then coded for transmission.
* **Inter frame compression** & **intra frame** **compression**

**Data Compression**

* LZ Coding is Lossless compression
* Adaptive dictionary encoding method in which repeated strings are referenced to an earlier occurrence of that string in the data.
* Only beneficial if 3 or more characters referenced

**Dictionary for Encoding and Decoding**

* No need for a dictionary to exist before the encoding and decoding take place
* The dictionary is build from the data as it passes through the system
* The transmitter and receiver can store a record of the uncompressed data and search that data when performing encoding and decoding.
* *String searching this way can cause slow down encoding – with certain applications requiring fast encoding and for these a dictionary is better*

Lossless Compression for Zipping Files

LZ coding forms the basis for many lossless compression systems used in computing

**Compression ratios**

Programs 1.4 : 1

Word Processor 2:1

Bit maps 3-1

**Application – Layer 7**

* **Hides the network** from the user.
* **Problem/ resource partitioning**: distributing a task around a network, Distributed databases

*Contains the application which users actually uses :*

* **DNS** ( *Domain Name System* ) Resolving ASCII to IP
* **SMTP**(*Simple Mail Transfer Protocol*) Email
* **FTP**(*File Transfer Protocol*) transferring files
* **HTTP**(*Hypertext Transfer Protocol*) WWW
* **Telnet** ( *Remote Terminal*) access apps remotely

**DNS – Domain Name System**

* Internet needs binary **address for any destination** before it attempts to route and deliver a message.
* Readable **ASCII** is **converted** to a binary **IP** address
* DNS is a **distributed database** which performs translation, mainly between *host names, email addresses, and their IP Addresses*

**DNS - How DNS Works**

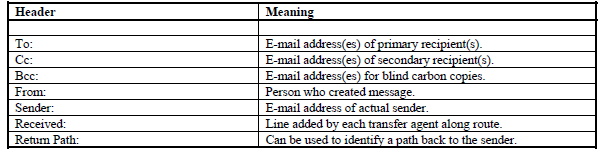
* Application sends **ASCII** in a **resolver procedure** which sends a **UDP** **packet** to local **DNS Server**
* DNS Server **looks up string** and **sends** back equivalent **IP address** to the resolver, which passes it to the App
* Application can then **set up TCP connection**/ **use** **UDP**
* If local DNS server **cannot resolve** the mapping it sends the query onto the DNS server at **the top of the domain tree**.
* If **no domain name** is given then it can only search the **local DNS**

**RFC – Request for comment**

* **Most** of the **internet standards**, **protocols**, etc are **defined** in **RFC documents**.
* Invites other **knowledgeable people** to **comment** on its **appropriateness** **for use**.
* **Draft Standard** when in working implementations at **two independent sites** for **4 months**
* **Internet Standard** when its good working solution

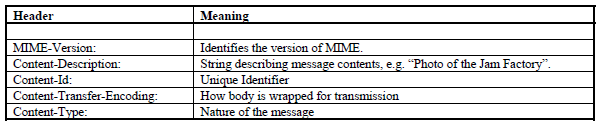
**Electronic Mail**

The **user Agent** in an email system is a program which **enables the user to compose, send, receive** and **control** their **message box**.

* Send electronic messages to anyone on network
* Key early protocols were **RFC** **821** and **822**

**MIME – Multi- Purpose Mail Extensions**

* Mine extensions allow **other data types** to be included, **graphics**, **video**, etc
* Added **5 new message header** to fit in with 822

**SMTP – Simple mail transfer Protocol**

* **Establishes TCP connection** to **port** **25** on **destination** **host**. Daemon **listens** **on host device**
* The **daemon receives email**, and **delivers** them **to** **appropriate mailboxes**.
* **Mail** **not** **deliverable**, **daemon** **send**s **error** **report**.

1. Sender expects **receiver** to **send** text **message** **identifying** **itself** and saying it is **prepared** **to** **receive mail**
2. If so, the **sender** **say** **who** the **source** **of** the **message** **is** and **who** the **destination** **person** **is**
3. If the **receiver** **has** a **destination** it **tells** the **sender** it can **send** the **message**
4. Client sends the **message** **over** **TCP** **connection**
5. The **server** **acknowledges** the **messages**
6. **all** **messages** are **sent** TCP **connection** **broken**.

**Email Gateways**

* Used to try to **convert email messages** **between** **different mail protocols**.
* **Difficult** due to **large differences** **between** way email is handed via **different protocols**
* **Problem**: Two addresses bounce messages

**WWW – World Wide Web**

* WWW is a **distributed collection** of hyper text documents stored on computers of the world.
* The **main protocol** used by the WWW is **HTTP**
* Every **website** has a **server** **waiting** **for requests** on a **TCP Port 80**. E.g. **client server basis**

**HTTP – Hyper Text Transfer Protocol**

**HTTP connects** to remote systems and reads their hypertext documents.

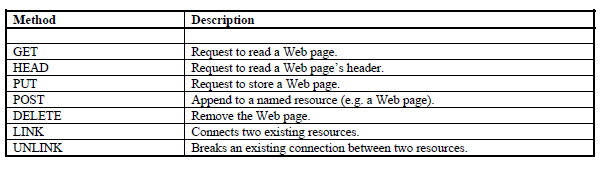
**HTTP - Fetching a web page**

1. User **selects URL** for site
2. Browser takes url and sends the second part of it to the **dns server for resolution** into Ip address

The first part of hte URL indicates that HTTP should be used

1. **DNS** **replies** **with** appropriate **IP Address**
2. Browser **establishes a tcp connection** to port 80
3. **Browser sends a command** **asking** the server **for** the **file** which is identified by the remaining url
4. The **server** **looks** into its user directory **and finds** the **html file. Sending it back** to the browser
5. Tcp **connection closed**
6. Browser displays the contents of the file
7. **Browser starts** **another tcp connection to fetch** another file. E*.g. graphics in the document.*

Each file is fetched as a seperate tcp connection and get command.

****

*These request methods are the basic commands which make up the protocol, and enable browsers to fetch web pages, send them to remote websites, and delete pages and link pages together*

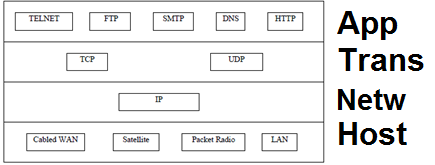
**Data Communications**

* **Data communication Protocols** set of rules by which computer communicate
* **Packets** segmented into a multiple packets.
* **Router** Nodes in a Wan
* **IMP:** Interface Message Processor.

Communication device interfaces a host to a network. Superseded by routers.

* **Host** Program wanting to communicate

**Typical Protocols – Internet Model**

* **UDP** – User Datagram Protocol
* **TCP** – Transmission Control Protocol
* **Telnet** – virtual terminal, open on a remote host
* **SMTP** – simple mail transfer protocol, email basis

**NAT - Network Address Translation**

* Assigns a public IP to a computer or group inside a private network, usually a firewall.
* This limits the number of public addresses on the internet.
* At network device public address is assigned and a source port.

**Sub Net – Layers 1-3**

IP is connectionless

**Carrier Sensing – Busy or Idle channel**

Can be sensed for the presence of a transmitted signal

Propagation delay is small and short relative to packet length. Channel state can be sensed before and during the transmission of a packet.

**CSMA – Carrier Sensing Multiple Access**

* **Carrier Sensing** – presence of transmitted signal
* **Multiple Access** - multiple transmits over medium

**Persistent** : Station waits until the channel is idle, then transmits a packet of data. If a collision happens, waits random amount of time and tries again when the station is idle.

**Non Persistent** : when channel is idle waits a random amount of time, then senses the channel again.

**CSMA/CD – with Collision Detection**

* When stations detect collisions, abandon transmission immediately. Ethernet is a persistent protocol
* Ethernet uses CSMA/CD
* **Contention** happens when two stations attempt to transmit but no one is sending.
* **Adaptive randomisation strategy**: minimises delays under light loads and gives stable performance under heavy loads. L doubles for each failure.
* Acknowledgement: first contention slot after successful transmission is for packet. Every station except receive must remain quiet.
* Channel Efficiency: P/ (P +5.4T)
* P = mean packet duration, T = propagation time