

# Equation Sheet (Y3S1) – Data Networking

---

## I. Application Layer

(none)

## II. Transport Layer

$$rate = \frac{cwnd}{RTT}$$

## III. Network Layer

(none)

## IV. Data Link Layer

$$d_d = d_{min} - 1$$

$$d_c = \frac{d_{min} - 1}{2}$$

$$r = \frac{n}{m}$$

$$\eta = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

## V. Physical Layer

(none)

## VI. Network Security

$$n = p \times q \text{ (choose p, q primes)}$$

$$z = (p - 1) \times (q - 1)$$

Find  $d$  such that:  $(e \times d) \bmod z = 1$  ( $e$  relatively prime to  $z$ )

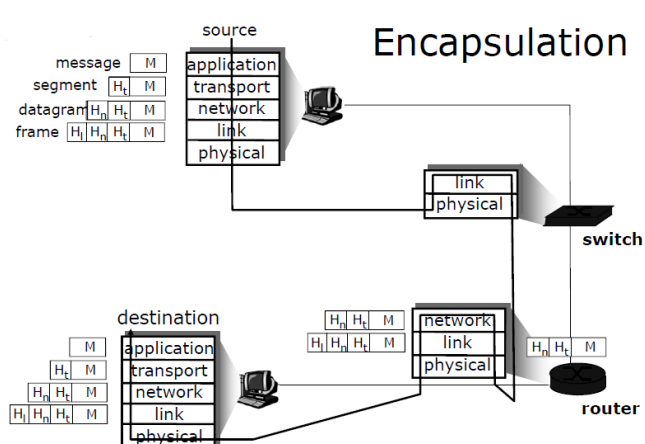
$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

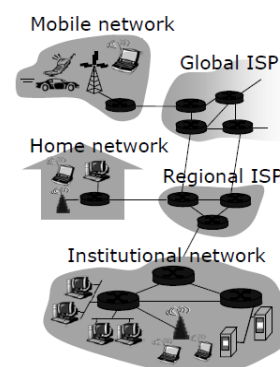
$$g^x \bmod n, g^y \bmod n \rightarrow \text{key is } g^{xy} \bmod n \text{ (Diffie Hellman)}$$

# Revision Sheet (Y3S1) – Data Networking

OSI Model			
Layer		Protocol data unit (PDU)	Function <sup>[3]</sup>
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access
	<del>6. Presentation</del>		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	<del>5. Session</del>		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment (TCP) / Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium



- ❖ millions of connected computing devices: *hosts = end systems* -running *network programs*
  - PC
  - server
  - wireless laptop
  - cellular handheld
- ❖ communication links
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- ❖ routers: forward packets (chunks of data)
  - access points
  - wired links
  - router



## I) Application Layer (7 - HTTP, FTP, DNS...)

Application architectures: client-server, P2P, hybrid

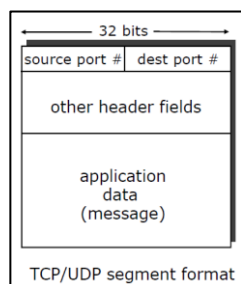
Application service requirements: reliability, bandwidth, delay

Specific protocols: HTTP, FTP, DNS

Typical request/reply exchange: **client requests** info or service, **server responds** with data & status code

Message formats: **headers** (fields giving info about data), **data** (info being communicated)

## II) Transport Layer (4 - TCP, UDP...)



Internet transport service model: connection-oriented reliable (TCP), unreliable datagrams (UDP)

TCP: **congestion** control, **flow** control, **connection** setup

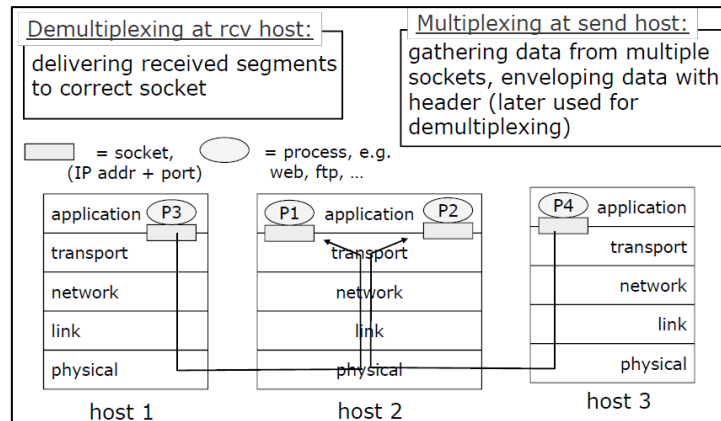
UDP: no-frills, **lightweight**, best-effort

Hosts receive IP datagrams (aside to be demultiplexed)

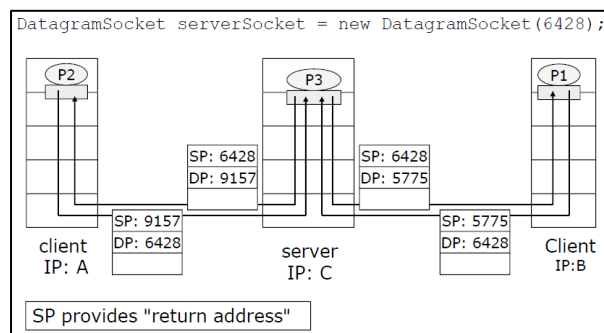
Socket: host-local, application-created, OS-controlled 'door' into which process can s/r

## A) Multiplexing/Demultiplexing (transport layer transmission)

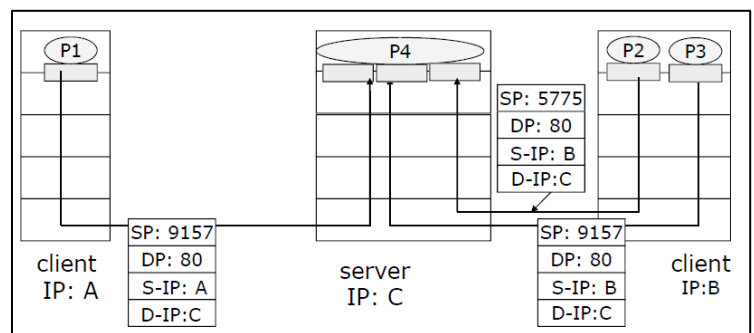
**Circuit switching** (reserved link), **packet switching** (statistical multiplexing)



**UDP connectionless demux**

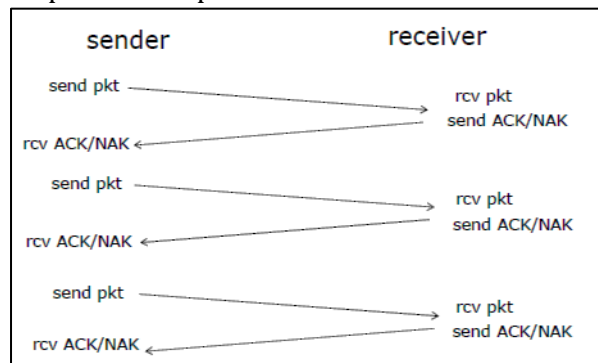


**TCP connection-oriented demux (multithreaded server)**

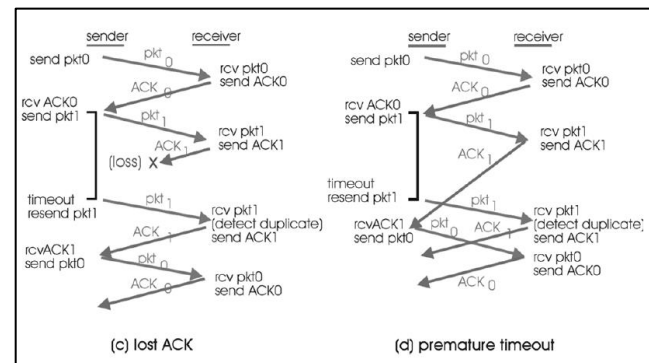


## B) Acknowledgment (reliable data transfer)

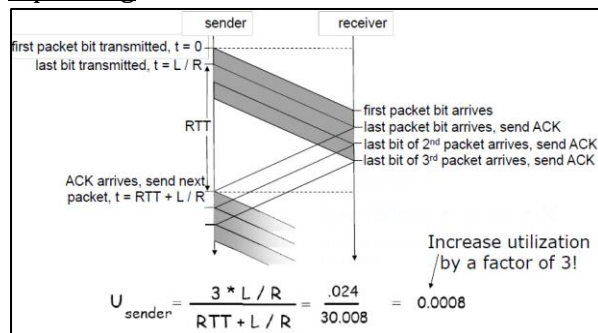
**Stop-and-Wait protocol**



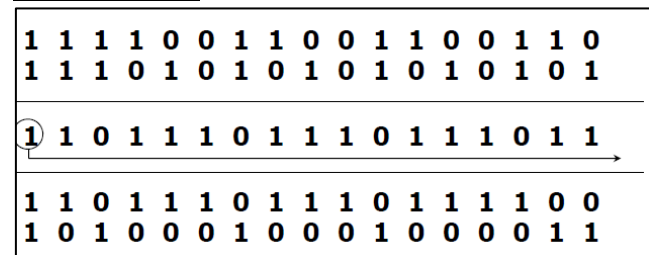
**S&W with timeout**



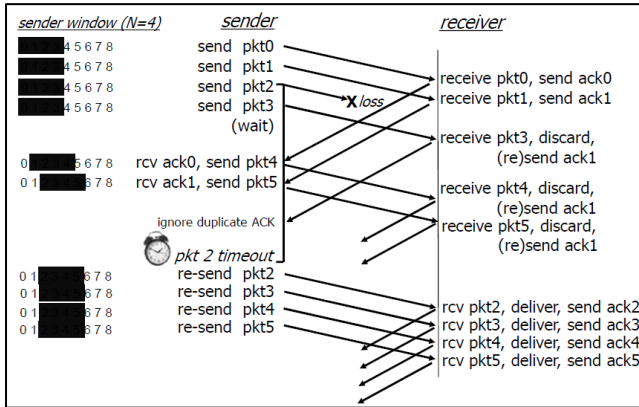
**Pipelining**



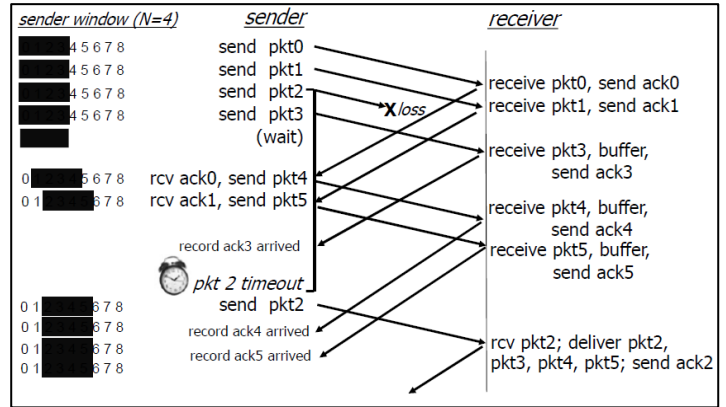
**UDP checksum**



## Go-Back-N protocol



## Selective Repeat protocol



## C) TCP in detail

Segment structure:

Reliable data transfer: see ACKs

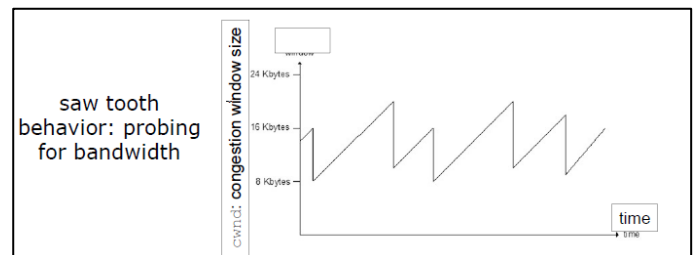
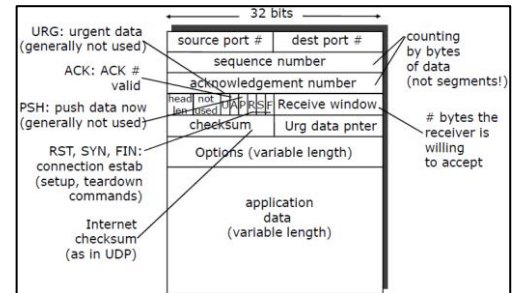
Flow control: adjust window size to match the rate of reception to the rate of sending

Connection management: 3-way handshake (SYNbit = 1, Seq = x; SYNbit = 1, Seq Y, ACKbit = 1, ACKnum = x+1; ACKbit=1, ACKnum=y+1) and closing connection (FIN, ACK, FIN; ACK)

Congestion control (too many sources - lost packets due to router buffer overflow, long delays due to queueing in router buffers): **two approaches** used - **end to end congestion** (no explicit feedback from network = TCP approach) & **network-assisted congestion control** (routers provide feedback via a single congestion bit = defines sending rate)

*Typical TCP congestion approach: additive increase (increase cwnd by 1 MSS every RTT until loss) or multiplicative (cut cwnd in half after loss)*

$$\text{Rate} = \frac{\text{cwnd}}{\text{RTT}} \quad (\text{bytes/s})$$



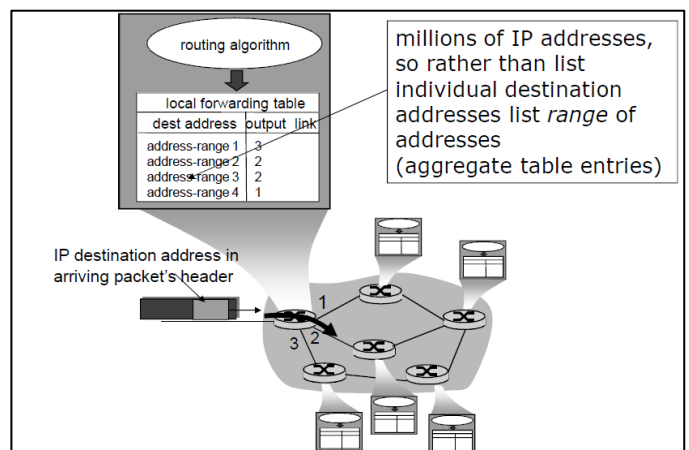
## III) Network Layer (3 - IP, ARP...)

Two key network-layer functions: **forwarding** (moving packets from router input to appropriate router output) & **routing** (determine route taken by packets from source to destination)

Network layer connection services: **datagram network** (network layer connection-less service e.g. Internet) & **Virtual Circuit** (network layer connection service e.g. AsyncTransferMode, Frame Relay - guarantee >40ms delay, in-order delivery, bandwidth... Call setup and teardown required)

Datagram networks: **no call setup** at network layer, **no state** about end-to-end connections, packets forwarded using destination host address (packets with same source-destination **may take different paths**) - **reliability is layered on at application/TCP layers**

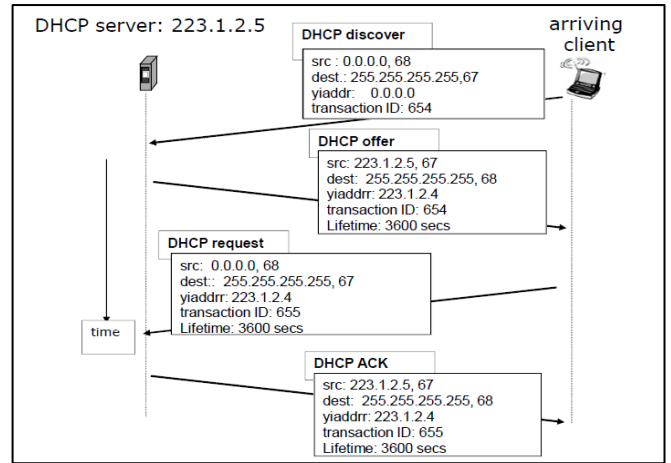
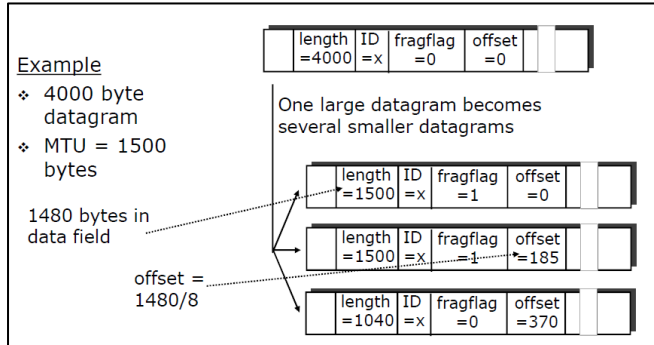
IP addresses: **32-bit identifier for host**, router interface (connection between host/router and physical link) Contains subnet part & host part.



**Subnet:** isolated sub-network defined by the **subnet mask** (e.g. /24 means first 24bits are reserved, rest is host)

**Acquiring an IP address:** **hard-coded** (/etc/rc.config) or Dynamic Host Configuration Protocol (DHCP server usually inside router).

**Note:** 225.225.225.225 reserved for **broadcasting** and 0000 for **network address**



**Fragmentation/Reassembly:** networks have a max transfer unit (MTU) size. IP datagrams larger than MTU are fragmented by router and reassembled at the final destination host. IP header bits used to identify and order fragments.

**ICANN:** Internet Corporation for Assigned Names and Number, **global authority** that **manages DNS root servers** (globally unique) - assigns domain names, resolves disputes, **allocates addresses continentally** (RIPE Europe, ARIN asia) → **allocated blocks of addresses regionally** (Nominet UK...)

**Address subnetting:** Individuals can subnet their own address block by **extending the subnet mask** into the host part of the address

**Router use of subnet mask:** to find out which subnet to route a packet to, the **subnet mask is ANDed with the address**

	0	Network(7)	Host(24)
A			
			Range = 1.0.0.0 to 127.255.255.255, Networks = 128, Hosts = 16,777,216
B	1	0	
		Network(14)	Host(16)
			Range = 128.0.0.0 to 191.255.255.255, Networks = 16,384, Hosts = 65,536
C	1	1	0
		Network(21)	Host(8)
			Range = 192.0.0.0 to 223.255.255.255, Networks = 2,097,152, Hosts = 256
D	1	1	1
			Multicast(28)
			Range = 224.0.0.0 to 239.255.255.255
E	1	1	1
			Reserved for Future Use(27)
			Range = 240.0.0.0 to 255.255.255.255

- ❖ Suppose
  - IP address 200.23.16.44
  - mask /28
- ❖ Gives
  - 11001000.00010111.00001000.00101100 AND 11111111.11111111.11111111.11110000 gives 11001000.00010111.00001000.00100000
  - 200.23.16.32
  - Thus within this network, subnet 2 is used
    - the highlighted bits give the subnet number

**Before:** Classful addressing - IP's used to be in 5 classes

**90s:** ran out IP → IPv6, private IPs (NAT)

**Now:** NATs used, 192.168 common

**NAT:** **network address translation** improves security (local net not explicitly addressable or visible from outside world hence 192.168.1.1 being so common as default gateway), allows ISP change without changing addresses of devices in local network, range of addresses not needed by ISP just one address for all devices. NATs must edit outgoing datagrams, remember every source IP/port NAT IP and port, edit incoming datagrams

**NAT traversal problem solutions:** **statically configure NAT to forward** incoming connection requests at given port to server, **automate static NAT port map config**, relaying (used in Skype) NATed client connects to a relay which bridges connection between external client and internal client.

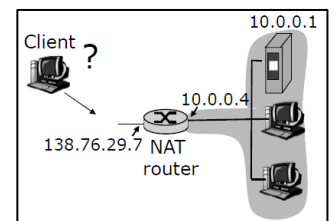
**ICMP:** **Internet Control Message Protocol** is used by hosts & routers to communicate network level info (error reporting, unreachable host, echo request (used by ping)...

**Traceroute:** source sends series of UDP segments to destination until it reaches destination and stops once it receives the ICMP "port unreachable" packet.

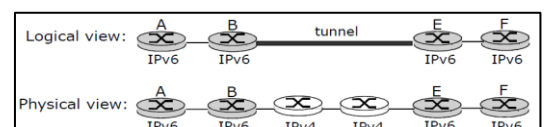
```

dataplicity@raspberrypi:/$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  12.414 ms  12.184 ms  12.056 ms
 2  * * *
 3  host-78-151-238-9.as13285.net (78.151.238.9)  15.677 ms  18.104 ms  18.008 ms
 4  host-78-151-238-56.as13285.net (78.151.238.56)  17.913 ms  * *
 5  * * *
 6  * host-78-144-4-35.as13285.net (78.144.4.35)  13.614 ms host-78-144-4-33.as13285.net (78.144.4.33)  17.368 ms
 7  * * *
 8  * * *
 9  * 72.14.237.177 (72.14.237.177)  11.708 ms 72.14.234.155 (72.14.234.155)  11.533 ms
10 google-public-dns-a.google.com (8.8.8.8)  11.426 ms  11.329 ms *

```



**IPv6:** improved performance with new header, checksum removed to improve processing time, not all routers can be upgraded, tunnelling IPv6 through IPv4 datagrams among IPv4 routers (uses ICMPv6)





Routing algorithm classification: global (all routers know full topology, link state algos) or decentralised (router only aware of neighbours, distance vector iterative algos); static (changes slowly over time) or dynamic (periodic fast response to link cost changes).

Link Cost: increases with physical distance, link delay (due to queue or actual delay), throughput

Static routing algorithms: (based on least cost paths) **Dijkstra's** shortest path (start at node A, update network, move to node B...) to make routing tables (aside)

A		B	
D	Next	D	Next
A	-	A	A
B	B	B	-
C	C	C	C
D	C	D	D
E	B	E	E
F	C	F	E
G	C	G	D
H	C	H	E

Dynamic routing algorithms: **Bellman-Ford** distance vector routing (set all to  $\infty$ , start at 0, SSABCDE, ASABCDE, BSABCD... up to n-1 iterations) - **good news travels fast, bad news travels slow** which leads to **Count to infinity problem of Bellman-Ford** (can be mitigated by agreeing on what value is taken to be infinity)

Link state routing (5 stages): **discover neighbours**, **measure cost** to neighbours, **construct a packet**, **send the packet**, **compute shortest path** to each router (done by flooding with sequence numbered and aging packets to prevent sequence numbers to restart at 0)

DV (distance vector) vs LS (link state): message complexity (DV exchange between neighbours only, variable time; LS  $O(nE)$  msgs sent), speed of convergence (DV variable may be routing loops + infinity issue; better LS  $O(n^2)$  with oscillations), robustness (DV node can advertise incorrect *path* cost, each node table used by other so errors propagate; better LS node advertise incorrect *link* cost, each node computes its own table)

Common protocols: RIP (routing information protocol, DV), OSPF (open shortest path first, LS), IGRP (Cisco)

#### IV) Data Link Layer (2 - PPP, Ethernet...)

##### A) Services, multiple access, MAC

Data-link layer is responsible for transferring datagram from one node to a physically adjacent node over a link.

**Nodes** = hosts & routers, **links** = communication path between nodes, layer-2 packet = frame (encapsulates datagram)

**Datagrams** may be transferred by different protocols over different links (Ethernet, ATM, 802.11...)

##### Services

**Link layer services - framing / link access** (encapsulate datagram into frame, add header & trailer, channel access if shared medium, Medium Access Control MAC address used in frame headers to identify source/destination), **reliable delivery between adjacent nodes** (as in transport layer, seldom used on wired low bit error links but often in wireless due to high error rate), **flow control** (pacing between adjacent sending and receiving nodes), **error detection** (errors caused by signal attenuation or noise, receiver detects errors and asks for retransmission), **error correction** (receiver identifies and corrects bit errors without retransmission), **half-duplex and full-duplex**.

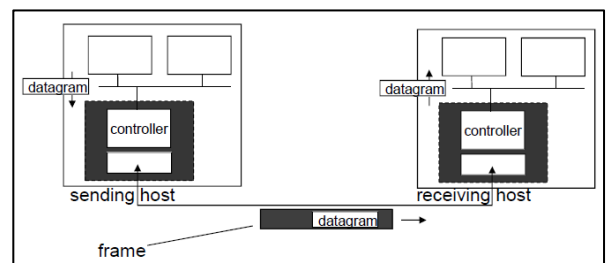
Implemented in hardware, firmware and software on **host adaptor** (network interface card, NIC) which attaches into host system bus.

Sending side **encapsulates datagram & adds error checking bits, flow control...** Receiver **looks for errors, performs flow control, extracts datagram** and **passes to upper layer** on receiving side.

**EDC** - error detection & correction (redundant bits) is not 100% reliable via **single bit parity** (single bit errors), **two-dimensional bit parity** (errors in blocks of data), **internet checksum** (multiple bit errors in packets), **cyclical redundancy check** (CRC - detects multiple bit errors in packets)

##### Multiple Access

Links can be **point-to-point** (PPP dialup) or **broadcast** (shared wire or medium such as Ethernet or 802.11). Ideally M nodes of rate R **transmit at R/M** in a **decentralized** simple **inexpensive** implementation.



## MAC

Three broad classes - **channel partitioning** (divide channel into time slots, frequency or code), **random access** (channel intact, allow collisions), **taking turns** (nodes with more to send can take longer turns)

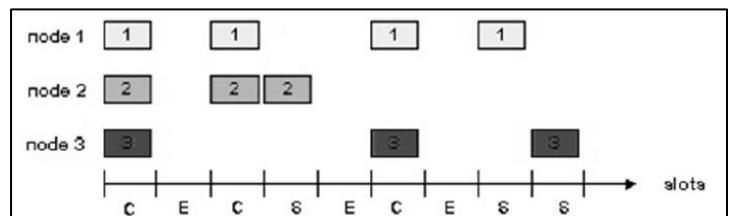
- Channel partitioning: **TDMA** (time division MA - gives channel access in rounds of fixed packet transmission time), **FDMA** (frequency division MA - channel spectrum divided into frequency bands) → share channel *efficiently* and *fairly* at high load (inefficient at low load)

- Random access: **MAC protocol specifies how to detect and recover from collisions** (e.g. slotted ALOHA, ALOHA, CSMA, CSMA/CD, CSMA/CA) - efficient at low load as single node can fully utilise channel (inefficient at high load due to large collision overhead)

- Slotted ALOHA - when node obtains fresh frame it transmits in next slot (assumes: all frames same size, time divided, nodes transmit only slot beginning, all nodes detect collisions). Best  $\eta = 37\%$

Pros: single active node can continuously

transmit at full rate of channel when others not transmitting, highly decentralised, simple. Cons: collisions, wasted slots, idle slots, nodes may be able to detect collision in less time to transmit packet, synchronisation required.



- ALOHA - when frame first arrives transmit immediately (collision probability increases but simpler, no sync needed). Best  $\eta = 18\%$
- CSMA - listen before transmitting, if channel idle transmit else exponential backoff (collisions can still occur due to delay, entire packet transmission wasted, distance and propagation delay determines collisions probability)
- CSMA/CD - like CSMA but collisions are detected within short time, colliding transmissions aborted reducing channel wastage

- Taking turns: **compromise between channel partitioning and random access**. Master node **polls** to invite slave **nodes to transmit in turn** (polling overhead, single point failure at master). **Token passed sequentially** to grant control (Bluetooth, FDDI, IBM Token Ring).

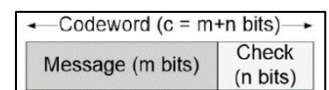
IEEE802 has two layers: **LLC** (logical link control - interface with network layer, manages flow control and error control - independent of network and works with many MAC protocols) & **MAC** (medium access control - assembles frames for transmission, disassembles, manages access to shared transmission medium)

## B) Error correction & detection

Types of error - **single bit error** & **burst error** (two or more bits, usually caused by impulse noise and fading)

**Hamming distance** - number of bits that differ between each pair of codewords

e.g. 00011 and 01101 have a Hamming distance of 3



**Coding scheme hamming distance** - minimum hamming distance  $d_{min}$  between all pairs of codewords

Error detection:  $d_d = d_{min} - 1$

Error correction:  $d_c = \frac{d_{min}-1}{2}$

So, if hamming distance is 3, 1-bit and 2-bit errors can be detected or 1-bit errors can be corrected.

For 000000, 000111, 111000, 111111,  $d_{min} = 3$ . (practice exercises)

Ratio of redundant bits to data bits is called redundancy:  $r = \frac{n}{m}$

Ratio of data bits to total bits is called code rate:  $c_{rate} = \frac{m}{c}$

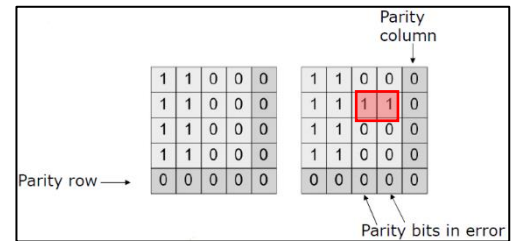
(practice the following 4 schemes)

### Parity checking (1D, 2D)

For 1010100, even parity gives 10101001 while odd parity gives 10101000. (total must be odd or even)

Parity bits produce a hamming distance of 2. (check when  $m = 2$  hamming distance is 2, all errors can be detected but not corrected)

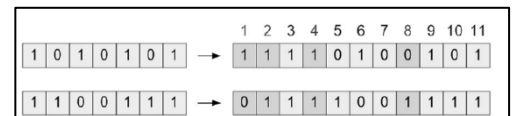
For larger data, 2D parity checking (more efficient) shown aside.



In 2D example,  $d_{min} = 3$  needed to correct single bit errors. Theoretical lowest limit reach using **Hamming Code**.

### Hamming Code

Number all bits 1 to  $c$ . Bits that are powers of two are check bits to record parity. Other bits hold the  $m$  data. (3 checked by 2 & 1, 6 checked by 4 & 2, 13 checked by 8, 4, 1...). Check 1 skip 1 → check 2 skip 2... (start with itself)



Can correct single errors (but not every single time, e.g. bit 3). Can correct burst errors using a matrix (2D parity)

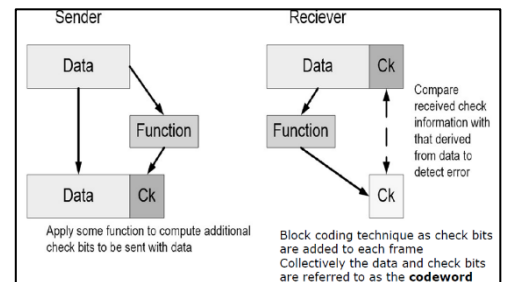
Appealing but carries large overhead. 1MB requires 24576 check bits for **correction** or 1 parity bit for **detection**.

### CRC

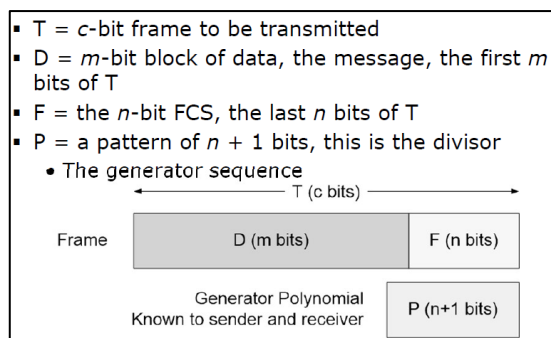
More common form of error checking: **Cyclic Redundancy Code (CRC)**

For  $m$ -bit message, transmitter generates  $n$ -bit sequence called **Frame Check Sequence (FCS)** resulting in  $c$ -bit frame which is exactly divisible by a predetermined number known to both sender and receiver. Receiver takes modulo and if remainder is 0 then no error in frame, else error in frame.

To check you can do binary addition with no carries (XOR) or binary addition with no borrows (XOR). This is shown aside.



0011	+	0011	-
0110		0110	
0101		0101	



CRCs can also be expressed in polynomial form where bits are coefficients of dummy variable  $x$ :

$$10100101 = x^7 + x^5 + x^2 + 1$$

Generate  $F$  by dividing  $2^n D$  by  $P$ . At receiver, divide  $T$  by  $P$ .  $T/P$  must have no remainder thus no error.  $P$  is 1-bit longer than FCS.

Apply both modulo only implementation of this.

Polynomial implementation is out of scope (CRC-16/32)

### Internet Checksum

Definition: 1s complement of 1's complement sum of all 16-bit words in transmission. Simpler.

E.g. 10101010, 00001111, 01011010, 10010010. Add all together & invert all bits of result (1s complement) - this is the checksum. **Carry out always added to LSB.**

To check, take data and checksum bytes and add all together. Result must be all 1s or all 0s.



### C) Link layer addressing & Ethernet

**IP address** - 32-bit network layer (used to get **datagram to destination IP subnet**). Can change, is dependant on IP subnet to which the node is attached (DHCP).

**MAC address** - 48-bit data link layer (used to get **frame from one interface to another physically-connected interface** within the same network - administered by IEEE bought in portions by manufacturers). Should not change and thus is portable across networks.

**ARP** - address resolution protocol. Each IP node has an in-memory ARP table mapping IP→MAC with a time to live (TTL) after which mapping is forgotten.

#### Ethernet

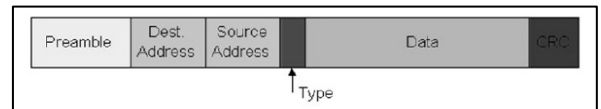
Dominant wired LAN technology (**cheap** ~£20 NIC), **simpler** than token LAN and ATM, 10Mbps-10Gbps. **Connectionless** (no handshaking between NICs), **unreliable** (no ACKs, gaps will be filled if app is using TCP else app will see gaps) and uses **unslotted CSMA/CD MAC protocol**. **Decentralised!**

$$\eta = \frac{1}{1+5t_{prop}/t_{trans}} \quad (t_{prop} = \text{max prop delay between 2 nodes in LAN}, t_{trans} = \text{time to transmit max-size frame})$$

Efficiency tends to 1 as  $t_{prop}$  goes to 0, as  $t_{trans}$  goes to infinity - much better performance than ALOHA.

**Topology** - bus topology till 1990s (all nodes in same collision domain), but now star topology prevails with a central switch, (each spoke running a separate Ethernet protocol allowing nodes not to collide with each other).

**Frame structure** - composed of preamble, destination address, source address, type, data and CRC.

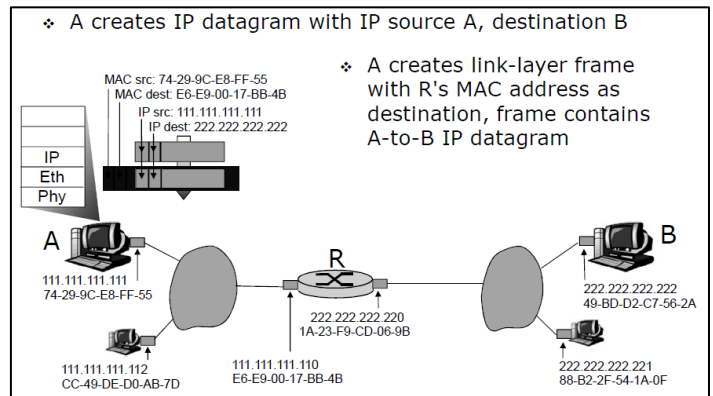


**Preamble:** 7x 10101010 followed by 10101011 (to sync clocks)

**Address:** if frame destination address matches data is passed to network layer protocol.

**Type:** indicates higher level protocol (mostly IP)

**CRC:** checked at receiver if error is detected frame is dropped and retransmission might be requested.



## V) Physical Layer (Wired Embedded Systems Networks, Coding / Error Detection & Wireless)

Embedded computing system: a system that includes a programmable computer but is not itself a general-purpose computer (sophisticated functionality, real time operation, low cost, low power, compact design, contains general and special purpose registers)

Embedded communications: **on-chip** (network on a chip NoC - packet or virtual circuit switching on a single IC via switches and interconnection links), **between devices on a PCB** (synchronous/asynchronous), **between physically separated devices** (802.11ac Wi-Fi, 802.15 Bluetooth).

### A) Embedded Systems Networks (wired)

**$\mu$ C needs to communicate** with **IO devices** such as DACs, EEPROMs, LEDs...

Memory-mapped IO: devices mapped to specific memory locations (like RAM), uses load/store instructions (simpler programming, devices have relatively high pin count thus package size will increase and so will cost)

Ported IO: special bus line and instructions, one or more addressable registers for control and data (lower pin count than MM IO, need special commands for IO, )

IO bus architecture: classified as **serial/parallel**, electrical characteristics, **protocol**...

Bus arbitration: **scheme 1** (every device connects to the bus request line and first come first served), **scheme 2** (daisy chain devices and pass the request to CPU device priority decreasing down chain), **scheme 3** (one bus request line per bus and arbitrator applies policy to decide who next), other MAC protocols...

Design considerations: data rate & error rate (impact max bus length), fault tolerance, bit sequence, device selection, synchronisation

**Parallel buses** include ATA, PCI (speed is faster than serial for given clock rate but **clock skew reduces speed** to that of lagging link, only requires a latch to receive data, usually for on-chip applications).

**Serial include** PCIe, SATA, Lightning, Thunderbolt (requires conversion to parallel form on chip at high speed, less interference for longer comms links i.e. between chips). Only serial buses will be investigated here as they **dominate embedded systems**.

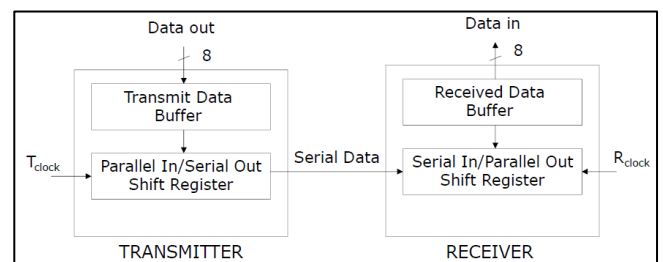
Serial design issues: **single line** (switching between high and low), **differential line** (both lines change voltage in opposite directions, allows for greater cable lengths than single line but requires more wire), **bus termination** (end of bus terminated with matching impedances for differential signalling OR unterminated/terminated at one end only)

UART: device required on  $\mu$ C because data is transferred serially outside the chip and parallel inside. Mostly used for asynchronous comms in **simplex** (one direction), **half** (one direction at once) and **full duplex** (two directions).

Framing: handled by UART to **start, stop and sometimes checksum** the data for **asynchronous comms**

Bit rate vs. Baud rate: **bit rate = data rate**, **baud rate = symbol bits** (including extra bits for start, stop & parity)

Bit banging: most rudimentary comms algorithm for limited devices, code can become complex, more memory is used up, more processing power is needed, other tasks and interrupts may lead to timing errors and glitches



## Asynchronous communication (RS232, RS422, RS485, USB/FireWire, CAN)

**No system clock** is used, which benefits systems where **CPU and IO run at different speeds**. **Simple & cheap**. **Maximum overhead** of 2/3 bits per character (~20%) good for non-continuous **bursty data**.

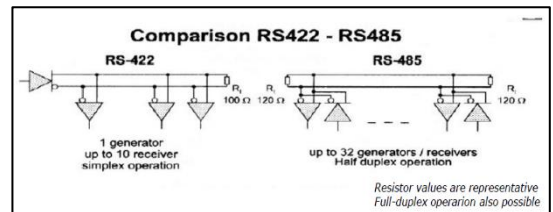
- **RS232** (serial): used pre-internet for computer comms with a MoDem (modulator/demodulator) mainframe computer over analogue telephone lines. **Very slow** with **TX, RX and SG pins** minimum (but **could support dataflow** to prevent overflowing of receiver buffer using RTS/CTS Xon and Xoff ASCII chars).

Architecturally bi-directional point to point. Two channels for **full duplex** comms. Logic levels 1 (-25V to -3V) and logic 0 (+25V to +3V).

Large amplitude, bipolar voltage swing. Non-differential signalling limits noise immunity. Max cable length 15m.

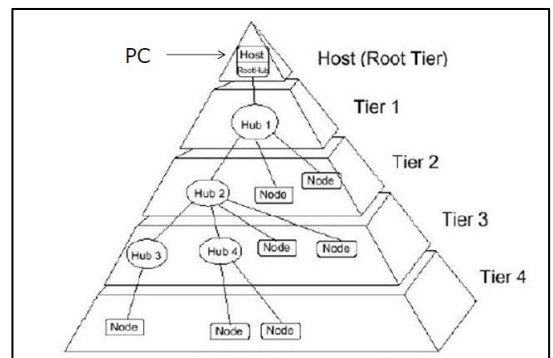
- **RS422** (serial): **differential twisted pair**, high **noise immunity** thus **high data speeds** (10Mbps) and **long distance** comms (up 1.2km), -6V to +6V nominal voltage swing, up to **10 receivers**

- **RS485** (serial): **differential** signalling, **multiple masters** (so more like a bus), speed and distance as in RS422, **common for DAQ & control** rather than RS232, **no specification for physical layer** protocol so can build protocol on top of this physical spec. Tri-state (1, 0, Z) capability for multi slave/master systems, **half or full duplex**, 2 or 4 wire version, slave devices have unique addresses.



- **USB/FireWire** (serial): 4-wire **differential** signalling, many versions, up to 480Mbps, plug & play, 127 external devices max, bus provides power and ground.

Tiered star structure (see aside), point to point, bus expanded by hubs



PC host server as master, devices must wait to be asked by host, each device has unique address, data sent in packet. Control transfers, bulk data transfers, interrupt transfers, isochronous data transfers, complicated protocol.

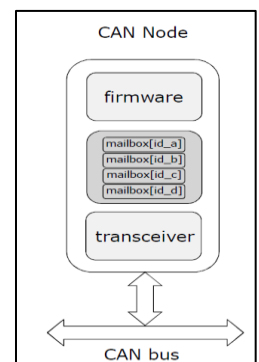
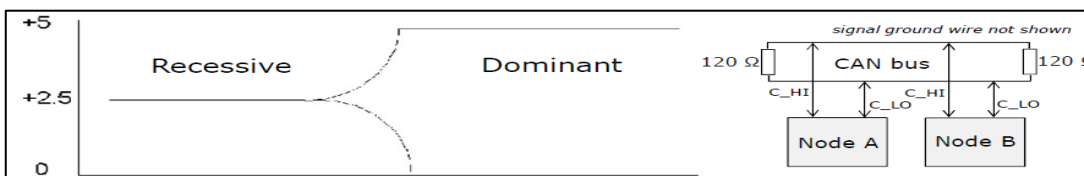
Firewire is **Apple competition to USB**, supports up to **63 devices**, **differential** signalling, **peer to peer** (device to device connections allowed).

- **Controller Area Network** (serial, asynchronous but required nodes to be in sync)

Two wire **twisted pair** bidirectional **differential signalling**. Bus used to **reduce wiring complexity** in ECU/MCUs (star topology → bus topology). **Low cost** (minimal wiring and hardware), **low noise** (shielded twisted pair reduces EMI as both signals move together), **scalable**. CAN does not specify physical interface only datalink layer.

CAN do not have addresses, they have **mailboxes** (buffers) which have pre-assigned identifiers. Each message carries identifier of intended target (multiple nodes can have a mailbox with the same identifier). All data is **broadcast** on the bus, read or **ignored**.

**Typical physical implementation** - *Dominant and Recessive* bits.



CAN arbitration is done as dominant bits override recessive bits (**see examples**).

	CAN	LIN	MOST
Architecture	Multi-Master	Master-Slave	Master-Slave
Time Behaviour	Non-Deterministic	Deterministic	Non-Deterministic
Maximum Bandwidth	1 Mbit/sec	20 Kbit/sec	150 Mbit/sec
Bus Access	Random, priority control	Schedule controlled	CSMA
Addressing	Broadcast	Broadcast & P2P	P2P, Group, Broadcast
Data Bytes per Frame	8	8	384
Physical Medium	Twin wire (twisted pair)	Single wire	Fibre-optic
Network Topology	Line, Star	Line	Ring
Application Areas	Powertrain, Chassis	Sensors, Actuators	Infotainment, Media

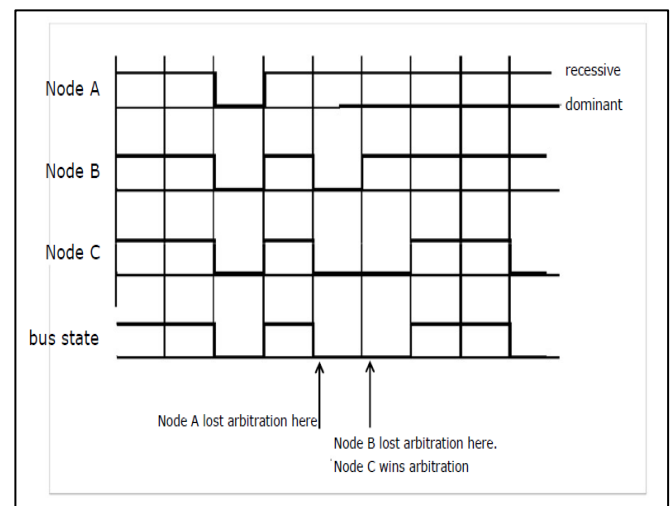
	CAN-FD	FlexRay	IP / Ethernet
Architecture	Multi-Master	Multi-Master	Multi-Master
Time Behaviour	Non-Deterministic	Deterministic	Non-Deterministic
Maximum Bandwidth	8 Mbits/sec (data phase)	20 Mbits/sec	100 Mbits/s
Bus Access	Random, priority control	Schedule controlled	Fully duplex
Addressing	Broadcast	Broadcast	Broadcast, Uni/Groupcast
Data Bytes per Frame	64	254	64 to 1522
Physical Medium	Twin wire (twisted pair)	Twin wire, Fibre-optic	Twin wire (twisted pair)
Network Topology	Line, Star	Line, Star	Point-to-Point
Application Areas	Powertrain, Chassis	Safety Critical, Backbone	Diagnostics, Cameras

Nodes transmit when bus is idle. When multiple nodes transmit simultaneously highest priority wins.

5 kinds of error checking performed by all nodes:  
**message level** (verify **checksums**, verify a node received the message via **ACKs**), **bit level** (verify **transmitted and received bits are the same** via a **node listening** as it transmits, verify **bit stuffing rule** is respected).

Encoding done via **NRZ**. **No common clock**, **nodes synch on data transitions**.

Too slow for car **infotainment** (ethernet or MOST used in this case for audio/video streaming).

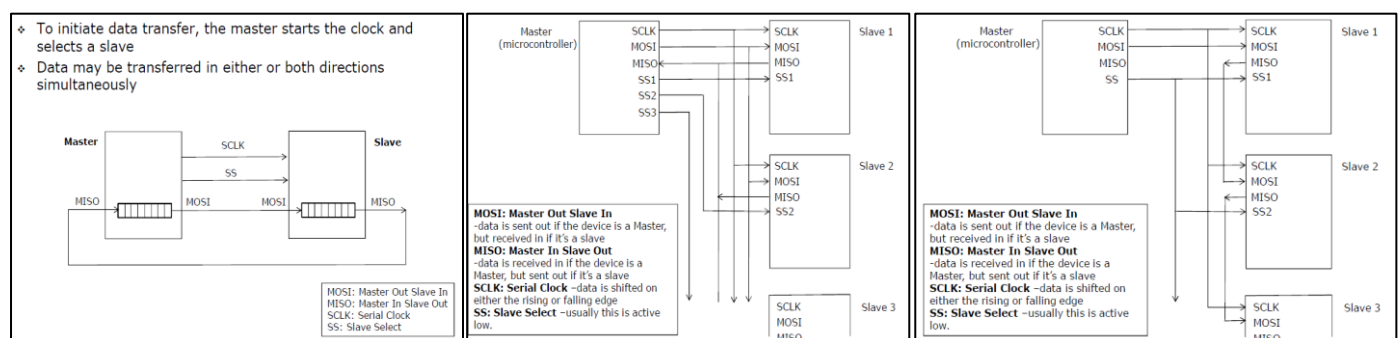


### Synchronous communication (SPI, I<sup>2</sup>C, LIN)

Transfers occur on **successive edges of system clock**, **clock must be transferred as a separate or embedded signal** with the data. Usually used for **short distances** (max data rate is higher than asynchronous).

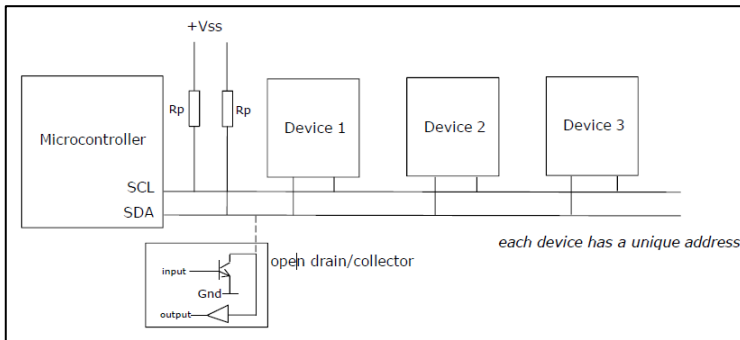
- **SPI** (serial) Serial Peripheral Interface works when **master starts the clock (SCLK)** and **selects a slave (SS)**. Data is **transferred in a shift-register** either or both directions simultaneously (**MISO/MOSI**) **full duplex**. **Multi-master** mode is possible. **Multi-slave** via daisy chain or extra SS bits. **Simple, protocol-free** (although can be added on top).

**Advantages** - **full duplex, fast & simple point-to-point streamed data, widely supported** (no addressing needed)  
**Disadvantages** - **multi-master config is complex, no ACKs, no arbitration, no flow control, short distance (PCB)**

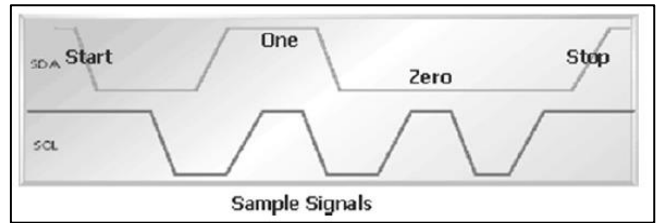




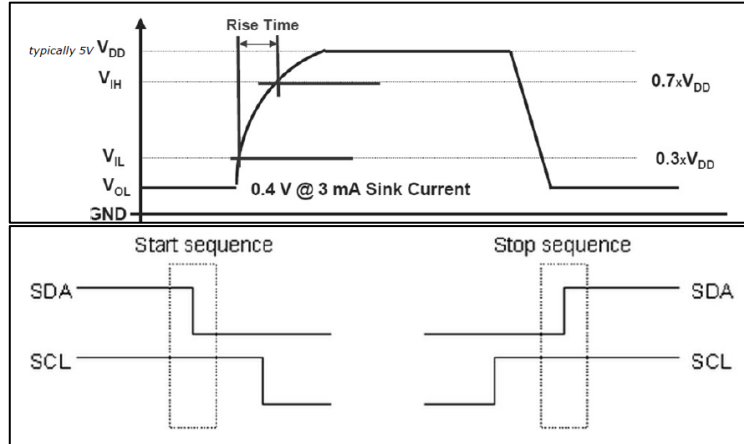
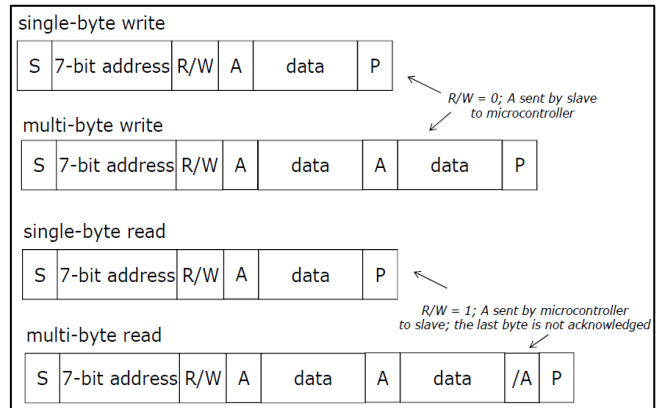
- **I<sup>2</sup>C** (serial) Inter-Integrated Circuit bus 3 speeds: **slow** (<100Kbps), **fast** (400Kbps), **high speed** (3.4Mbps). **Half duplex, multi-master**. No chip-select or arbitration logic. Lines float *high* by default via resistors, driven *down* by open-drain drivers. Master initiates data transfer through SDA (data), drives SCL (clock).



Data transmission scheme as follows:



Data transfer starts with **master transmitting the address of the slave**.



**Bus arbitration** is done by monitoring SDA (if 2 masters generate a start at the same time there will be a conflict)

**Clock stretching** occurs when master issues a read, slave places data on bus but master controls clock so if slave is not ready master must wait for the slave (think EEPROM coursework).

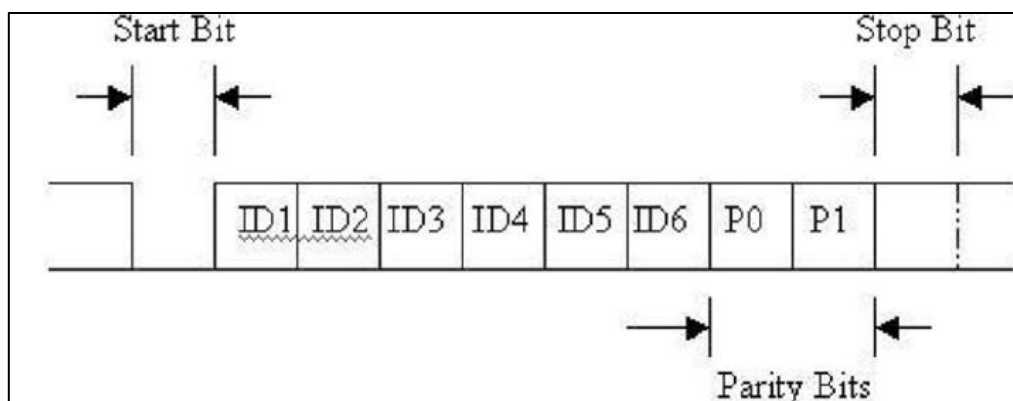
**Advantages** - simple hardware & protocol, suitable for **on & off-board bursty data**, easy addition of devices via addressing, cost and complexity scalable, widely supported

**Disadvantages** - limited address space, bus length requires I<sup>2</sup>C repeater to extend, dynamic removal of devices is hard as no inventory function, overhead of flow control inefficient.

- **Local Interconnect Network** (serial) automotive bus **slower** (20Kbps) **low-cost alternative to CAN**.

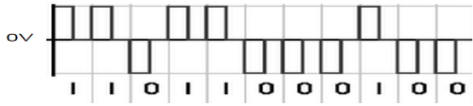
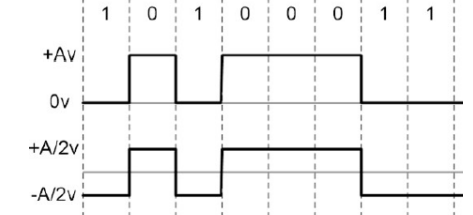
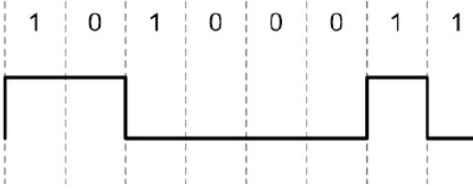
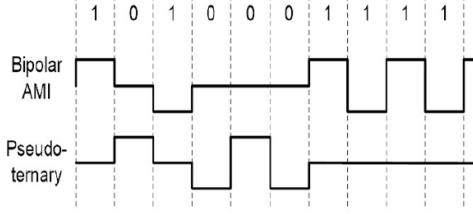
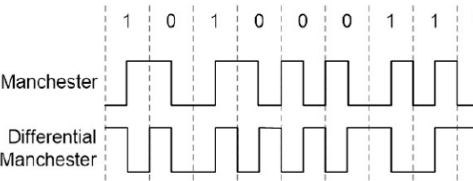
**Single-wire** (plus signal ground) bus. **Single master, multi-slave** (all message from master so **no arbitration**).

LIN used in some parts of cars. Byte oriented (data sent one byte at a time, each byte field contains: dominant start bit, 8 data bits, recessive stop bit & ID field is one byte long including parity bits, message ID and sender/receiver info)



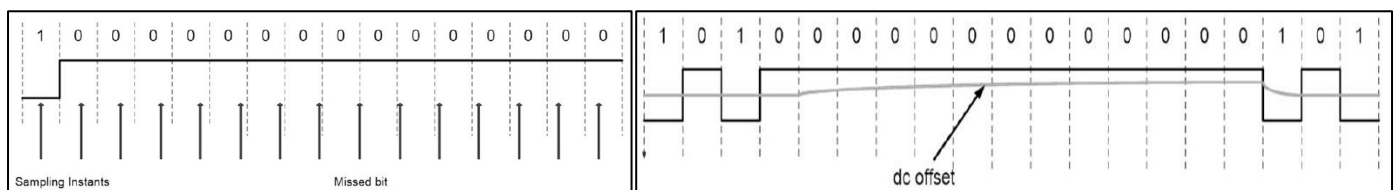


## B) Coding of Digital Data and Error Detection

Coding scheme	Encoding	Notes
Return to Zero (RTZ)		Requires twice bandwidth of NRZ Can also use 0V to represent 0
Non-Return to Zero (NRTZ)		<b>Unipolar:</b> (0 to +AV) [ $P=A^2/2$ ] <b>Bipolar:</b> (+A/2 V to -A/2 V) [ $P=A^2/4$ ]
Non-Return to Zero Invert on Ones (NRTZI)		Invert on 1s, maintain level on 0s As with all RTZ, simple & efficient use of bandwidth (but suffers from long 0 - DCshift/sync loss)
Multilevel Binary (MB)		<b>Bipolar-AMI</b> (alternate mark inversion): 0 no signal, 1 +ve OR -ve alternately <b>Pseudoternary:</b> 1 no signal, 0 +ve or -ve alternately
Manchester (M)		<b>Manchester:</b> +ve trans. for 1, -ve trans. for 0 (always mid-bit trans.) <b>Differential Manchester:</b> transition at beginning of bit for 0, no transition for 1 (mid-bit trans.)

Sampling must be done in the middle of the bit period which causes issues:

- Loss of synchronisation (bit can be missed or sampled twice if clocks are not running at same rate)
- Low frequency content (long streams of 1s and 0s will cause problems for synchronisation & DC offset)



**Bandwidth considerations:** NRTZ signal elements have same duration as data elements, **data rate = modulation rate** (bandwidth related to modulation rate - number of symbol elements per second in **baud**) thus NRZ needs less bandwidth than RTZ

**Spectral considerations:** lack of DC component means AC can be used reducing interference (distortion and interference is related to spectral properties as transmission characteristics usually worse at edge of band thus should concentrate most power at centre for best performance). MB has no DC drift as average signal value is 0V. Some schemes such as ISDN introduce additional bits to force transitions and avoid long runs of 0s or 1s. Detecting 3 levels at the receiver is harder than 2 for noisy environments (greater BER for given SNR).

**Tolerance to wiring inversion:** sometimes inversion is required for complex wiring installations (NRZI is tolerant of polarity inversion)

Manchester (used in some Ethernet) has error detection as absence of a transition indicates error, is polarity tolerant, no DC component, resistant to long 0s.

**Block codes:** combine ability of positive attributes in biphasic codes with a reduction in bandwidth (done by transmitting  $m$  data bits from  $n$  data bits such that  $m > n$  to maximise transitions and minimise DC component). Ultimate goal is to achieve a DC balance, provide clock sync with low signal rates.

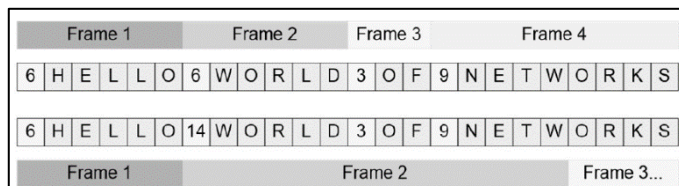
Coding scheme	Encoding	Notes
Biphase block code (4B/5B-NRTZI)		Modulation rate is $5/4 \times$ data rate (1010=10110, 0011=10101...) Using table & NRTZI within blocks
Ternary block code (4B/3T)		Codes chosen with frequent trans. (1010=+++ or ---, 0011=+-0...) Modulation rate is $3/4 \times$ data rate

**Scrambling:** used to violate simple codes to give no dc component, no long sequence, no reduction of error rate, error detection capabilities

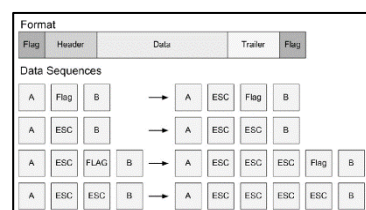
Coding scheme	Encoding	Notes
Bipolar alternate mark inversion (AMI)		Flips on every 1
Scrambling techniques (B8ZS) (HDB3)		<b>B8ZS:</b> bipolar with 8-zeros substitution (code exception) <b>HDB3:</b> high density bipolar-3 zeros (4 zeros code exception)

**Frame construction:** must mark **start** and **end** of **frame** (4 methods - **character count**, **flag bytes** with **byte stuffing**, **starting and end flags** with **bit stuffing**, **physical layer coding violations**)

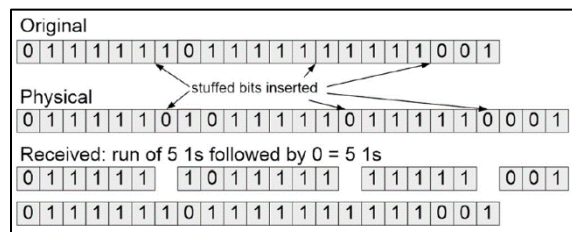
- Character count:** include field in frame containing character count (character count error can cause error at frame level, error checking can detect errors but since sync is lost destination can no longer identify boundaries thus cannot retransmit  $\therefore$  rarely used on its own)



- Flag bytes (byte stuffing):** use *special byte* called **flag** byte to mark start and end of frames (here 01111110). If special byte occurs in data, this data must be prefixed with the special escape bytes ESC. Used in Point to protocol.



- Starting & end flags (bit stuffing):** flag used again as in byte stuffing but instead of adding an escape byte ESC, a 0 bit is inserted. When 111110 is found in the data it is replaced by receiver as 11111. Allows arbitrary bit patterns to be inserted in the data (**data transparency**).



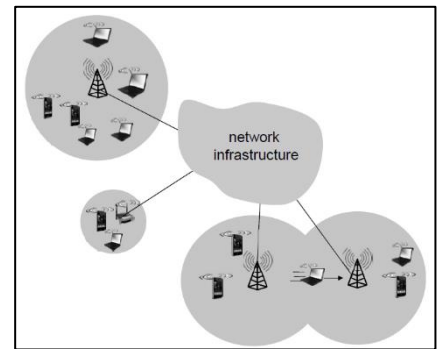
- Physical layer (coding violations):** ME requires a transition in the middle of data bit hence a data bit with no transition could be used to signal frame end. In 4B/5B from 32 possible codes only 16 used so others could be for frame boundaries.

## C) Wireless networks

Wireless network is composed of: **wireless hosts** (mobiles, laptops...), **base stations** connected to a wired network (relays, cell towers, access points...) and **wireless links** used to connect the hosts to the base stations.

Typically, **802.11ac** used for 10-50m while **802.15** used from 10-30m.

Two modes: **infrastructure mode** (base station connects mobiles into wired network), or **ad hoc mode** (no base stations, nodes only transmit to other nodes within link coverage and try to organise themselves into a network and route among themselves)

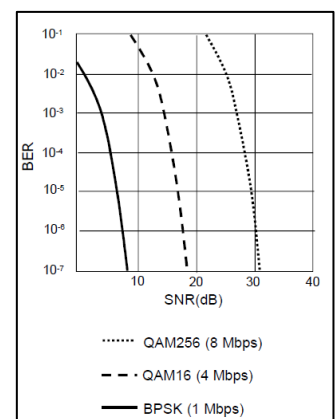


	Single hop	Multiple hops
<b>Infrastructure (e.g. APs)</b>	Host connects to base station (WiFi, WiMAX, cellular) & to the larger Internet	Host may have to relay through several wireless nodes to connect to Internet (mesh net)
<b>No infrastructure</b>	No base station, no connection to larger Internet (Bluetooth, ad hoc nets)	No base station, no connection to Internet. May have to relay to reach other node (MANET)

Differences from wired: **decreased signal strength** (attenuation by propagation through matter due to **path loss**), **interference** from other sources, **multipath propagation**... see DMC)

SNR vs BER tradeoff: **increase power** → **increase SNR** → **decrease BER**. Choose physical layer that meets BER requirement giving highest throughput (SNR changes mobility - dynamically adapt physical later)

Multiple wireless senders and receivers cause issues: **multiple access**, **hidden terminal problem** (A & B hear each other, B & C hear each other, A & C do not, A is unaware of the interference from C), **signal attenuation** (A & B hear each other, B & C hear each other, A & C do not hear each other interfering at B)



Code Division Multiple Access (CDMA): **unique code assigned to each user**, all users **share frequency** but each user has **chipping sequence to encode data** allowing multiple users to coexist

$$\text{encoded signal} = \text{original data} \times \text{chipping sequence}$$

Decoding: **inner product** of encoded signal and chipping sequence

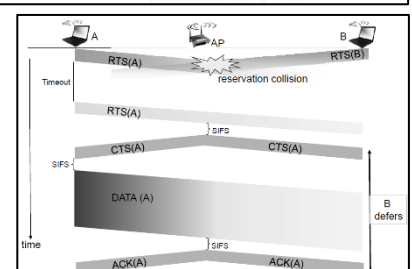
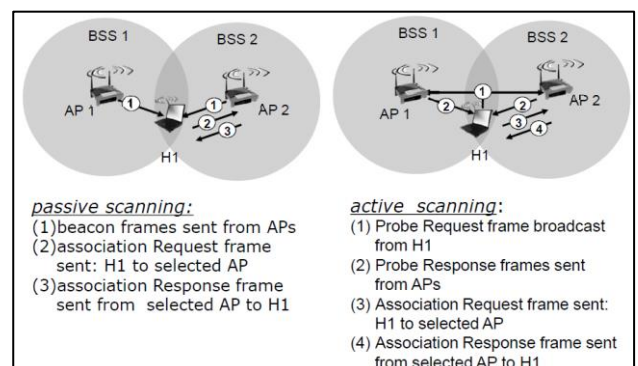
**802.11 wireless LAN:** 802.11b (2.4-5GHz unlicensed 11Mbps), 802.11a (5-6GHz, 54Mbps), 802.11g (2.4-5GHz, 54Mbps), 802.11n (multiple antennae 2.4-5GHz, 200Mbps) all using **CSMA/CA for multiple access** & all have infrastructure and ac hoc modes.

802.11b has **11 channels at different frequencies** (AP admin chooses AP frequency - possible interference with neighbours), host must associate with an AP by scanning channels, listening for **beacon frames** containing AP name (**SSID**) and **MAC address**, **requesting AP to associate** with and **authenticate**, run **DHCP to get IP in AP subnet**.

**Multiple access** - avoid collisions as 2+ nodes transmit at the same time using CSMA/CA (collision avoidance)

**MAC Protocol** - CSMA/CA states if sense channel idle for DIFS then transmit entire frame (no CD), if sense channel busy the exponential backoff. Receiver ACK after SIFS.

**Avoid collisions** - allows sender to *reserve* channel rather than random access of data frames (avoids collisions of long data frames). Sender first transmits a small request-to-send (RTS) packet to CSMA base station. CSMA response is clear-to-send CTS, heard by all nodes.



## MAC Coordination function:

Mechanism used to **control access to wireless medium** within BSS (ethernet would use CSMA/CD)

**Wireless cannot use CD since variations in signal strength are large**, many radio systems do not support concurrent listen and transmit operations

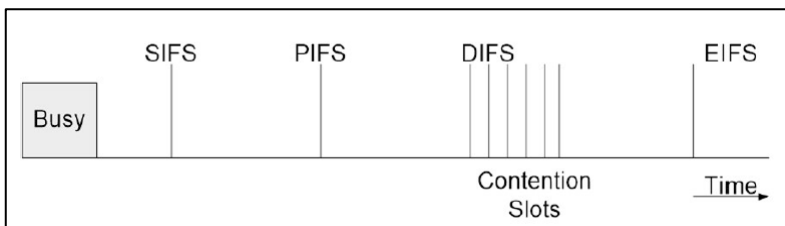
**2 modes** - **DCF** (distributed coordination function - all stations take part in CTS-RTS as before), **PCF** (point coordination function - AP manages contention free system by polling, on top of DCF)

**DCF issues** - **collisions occur** (when two parties send RTS at same time, no CTS is received and RTS is retried), **errors occur in frames** (more prevalent than in wired Ethernet thus error recovery at MAC level is needed using ACK frames leading to process restart if no ACK is received)

**Probability of bit being in error  $p$**  thus probability of frame to be error-free in a  $n$ -bit frame is  $(1-p)^n$  thus the longer the frame, the greater the probability of error.

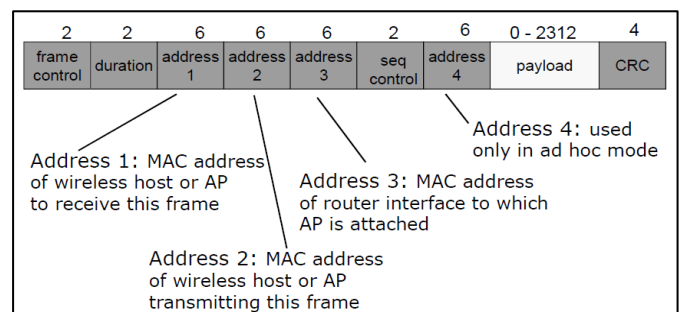
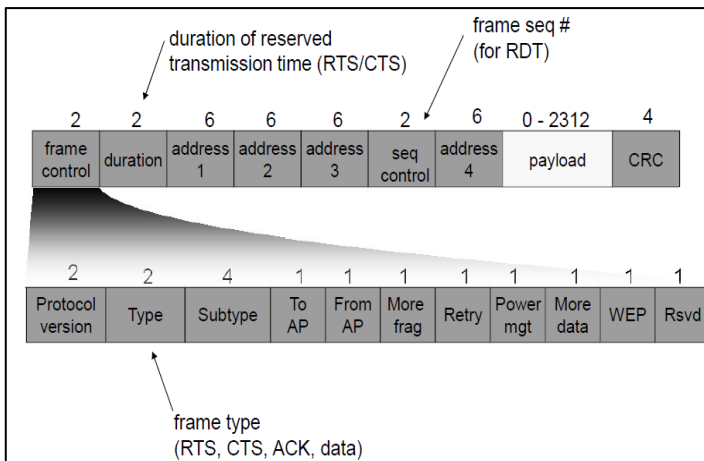
**Reduce retransmission due to bit errors by:** splitting frame in smaller fragments (each ACKed, retransmission reduced at the expense of additional bandwidth used for the ACKs), each fragment has own checksum (stop and wait protocol must be used), station non-collision **PCF** mechanism (allows repeated fragments to be sent without a new RTS/CTS handshake).

**PCF solutions** - base station periodically sends Beacon Frame (broadcasts system parameters, allows hosts to sign up for PCF service). Base station polls all hosts signed up to PCF and accepts their frames. Supports time-critical requirements, causes stations to sleep saving power. Gaps allows DCF & PCF to coexist.



**Interframe spacing** - SIFS (short Inter-Frame Spacing - ACK/CTS data frame, fragment, PCF response) has higher priority than PIFS (Point coordination IFS - beacon frame or polling of a host by base station), DIFS (DCF IFS - RTS), EIFS (Extended IFS - bas or unknown frames)

**Frame addressing** - see images aside and below



**Mobility within subnet** - if H1 remains in same IP subnet then **IP address can remain the same**. Switch must find out **which AP is associated with H1** via **self-learning** (remember which port used to reach H1).

**Rate adaptation** - dynamically change transmission rate (physical layer modulation technique QAM, BPSK...)

**Power management** - node-to-AP can send sleep until next beacon frame. Beacon frame contains list of mobiles with AP-to-mobile frames waiting to be sent. Node stays awake if AP-to-mobile frames need to be sent else sleep.

**802.15 Bluetooth:** less than 10m diameter, no infrastructure, master/slaves, 2.4-2.5GHz up to 721 kbps.



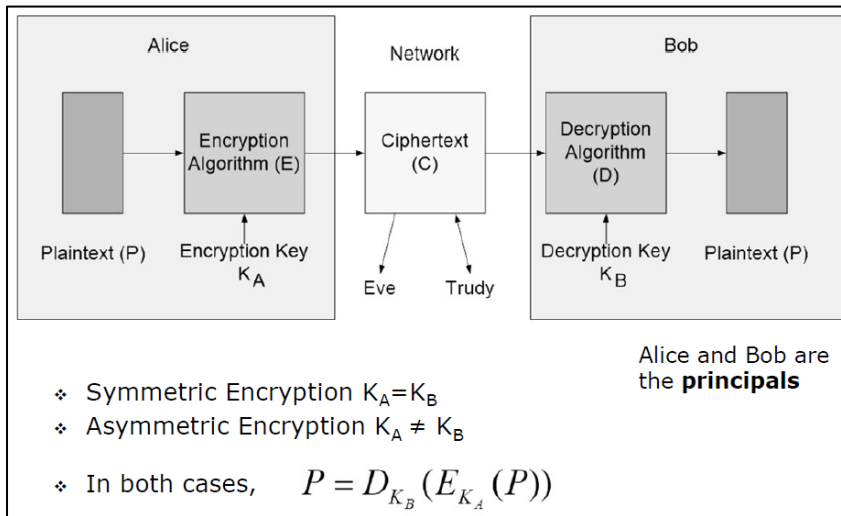
## VI) Network security

**C - Confidentiality** (secrecy - keeping information out of the hands of unauthorised parties) [e.g. AES-256]

**I - Integrity** (verification - ensure data is not modified by unauthorised 3<sup>rd</sup> parties) [e.g. SHA-256]

**A - Authenticity** (identification - ensure user is indeed the same person they claim to be) [e.g. 2FA]

**Non-repudiation** - ensure sender of a message cannot subsequently deny having sent it [e.g. blockchain]



### Types of attack:

- **Passive** (Eve the eavesdropper)
- **Active** (Trudy the intruder)

Algorithm known to attacker but not the key (long keys increase **work factor** required for a brute force attack).

Assumptions: attacker knows full ciphertext, some ciphertext matched with plaintext

Some encryption uses both symmetric & asymmetric cryptography (SSL, TLS, SSH)

### A) Symmetric Cryptography (DES, AES)

Shared secret (private) key used only. Both Alice and Bob must use the same key.

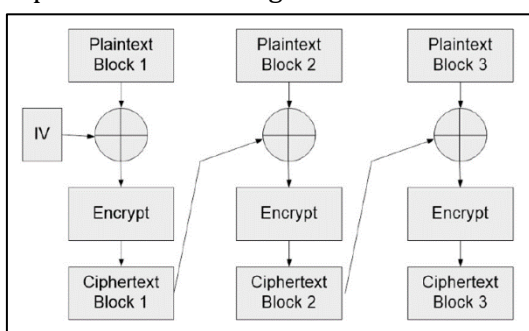
Block-cipher: common method where plaintext block is sent as the same-sized ciphertext block (sizes usually 64-256 bits)

Symmetric cryptography using XOR and scrambling techniques with parts of the key to create ciphertext.

Several rounds of XOR & scrambling are used to encrypt & their reverse operations must be done to decrypt thus key must be known for decryption.

Standard	Data Encryption Standard (DES)	Advanced Encryption Standard (AES)
Use	1970-2000	2000-present
Vulnerability	audited, broken	audited, secure
Key size	56-bit	128-bit, 192-bit, 256-bit
Block size	64-bit	128-bit
Rounds	16 (19 total) R0: permute, R1-16: <b>key-functions</b> , R17: swap 2 32-bit word halves, R:18 inverse R1	10, 12, 14 (11, 13, 15 total) R0: XOR Round Key, R1-10/12/14: Sub bytes, Shift Rows, Mix Columns, XOR Round
Other information	Strengthened using triple DES (slower) but later abandoned	Decryption by repeating with different tables, algorithm is complex but highly performant

### Cipher-block chaining



**Initialisation vector (IV)** used to remove the property that the same plaintext block always results in the same ciphertext block for a given key. Works by **XOR each plaintext block with preceding ciphertext block before encryption**. First block uses a random IV (like a seed).

Main issue with Symmetric Cryptography: secret key distribution without compromising/revealing the key



## B) Asymmetric Cryptography (RSA)

Secret (private) key is not shared thanks to the use of public keys being combined with private keys (a.k.a. public key cryptography).

Standard	Rivest-Shamir-Adelman (RSA)
Use	1977-present
Vulnerability	audited, secure
Key size	512-bit $\times$ 512-bit = 1024-bit
Block size	1024 bits
Rounds	1

### Algorithm:

- Choose two large (512 bits) primes  $p$  and  $q$  and keep them secret.
- Calculate  $n = p \times q$  and  $z = (p - 1) \times (q - 1)$
- Choose a number  $e$  relatively prime (does not share common factors) to  $z$ .
- Find  $d$  such that  $(e \times d) \bmod z = 1$
- Divide plaintext into blocks so that value of plaintext  $M$  is in the range  $0 \leq M < n$  (block  $k$  bits  $n < 2^k$ )

### Encrypt:

- Ciphertext  $C = M^e \bmod n$

### Decrypt:

- Message  $M = C^d \bmod n$

Thus  $e$  and  $n$  are needed to encrypt (public key:  $n, e$ ) and  $d$  and  $n$  needed to decrypt (private key:  $d, n$ ).

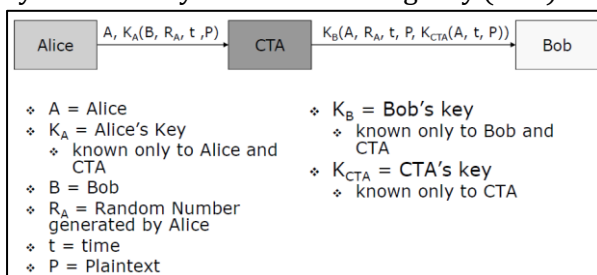
### Security:

Attacker can use Euclid's algorithm to get  $d$  if  $z$  and  $e$  are known but this is extremely computationally expensive. In practice  $p$  and  $q$  are 512 bits so  $n$  is around 1024 bits hence block size can be 1024 bits. Some form of chaining may be required as in AES. RSA is very slow compared to AES and generally used to encrypt a one-time session key for use with a symmetric key algo (like in SSH).

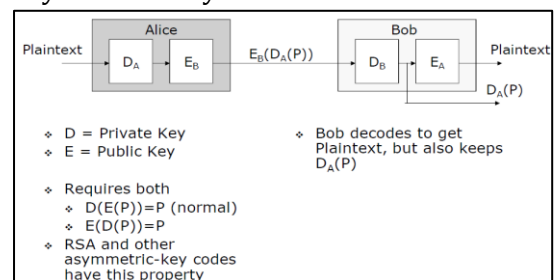
## C) Digital Signatures, Key Distribution & Authentication

Digital signatures (sender must be verified, message cannot be repudiated, message cannot be altered):

- Symmetric-key central trusted agency (CTA)



- Asymmetric-key

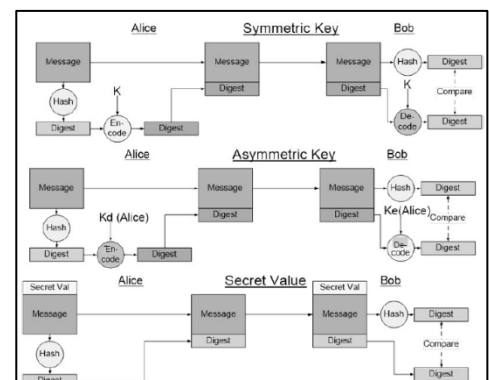


- Message digests with asymmetric-key for signing

Generate a one-way hash that guarantees the contents of the message (SHA-256)

To protect against verification of sender, repudiation and message concoction, the digest itself must be modified (aside)

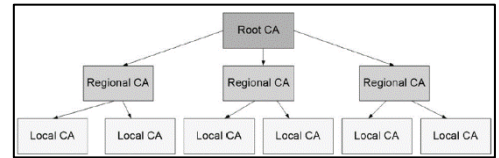
**Birthday attack:** repeatedly finds the hashes of certain values to find a collision for use in compromising integrity (64-bit digest can lead in a collision in around  $2^{32}$  attempts)



## Key distribution:

Asymmetric cryptography requires **distribution of demonstrably authentic keys**. Public keys must be **traced back to owners to prevent attacks**.

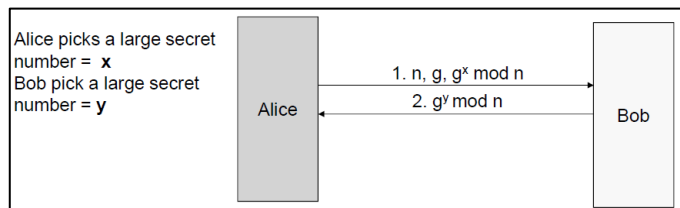
**Public key certificates** - used to verify authenticity of keys. Done via centralised **certification authority (CA)**. CA **signs certificate using CA private key**, certificate is validated by Alice using a well known public key.



To spread the work, this CA is placed in a hierarchy with **root, regional** and **local CAs** in a **hierarchy of certificates**. This is called a **Public Key Infrastructure (PKI)**. In practice, Alice does not need to go up the chain, Bob will include the set of all CAs up to root in his certificate. There are multiple root CAs so you can check with multiple reputable ones for security (most web browsers, Microsoft... protocol used is X.509).

## Authentication:

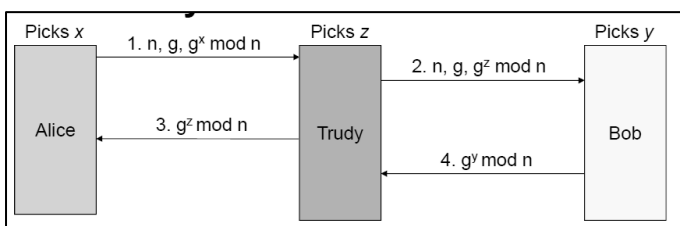
First, we must **generate shared secret keys**. This is done via **Diffie-Hellman key exchange**:



Key is  $g^{xy} \bmod n$

This cannot be cracked computationally fast.

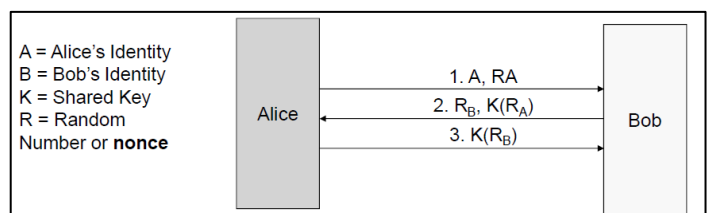
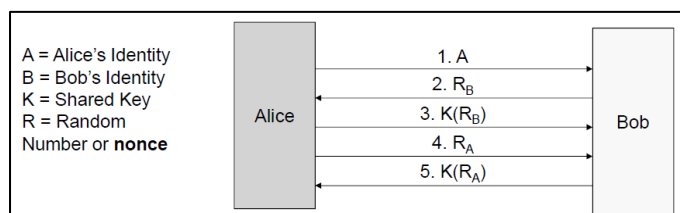
This is however vulnerable to a **Man-In-the-Middle attack (MIM)**:



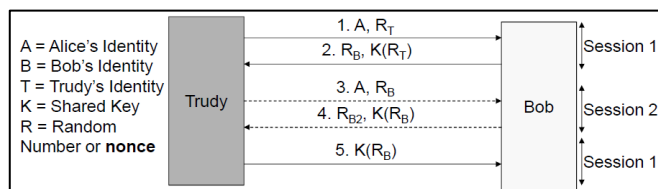
Trudy can establish secret keys with both Bob and Alice provided she can intercept all messages.

Diffie-Hellman alone is also not efficient long term as a private key is needed for each communication partner.

We can now use the **shared secret key** (second version reduces number of messages) for authentication:

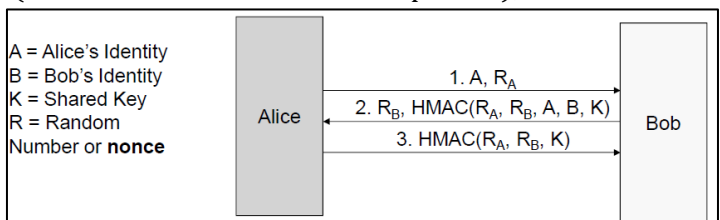


Both methods require key to be established *a priori* and both are vulnerable to **reflection attack**:

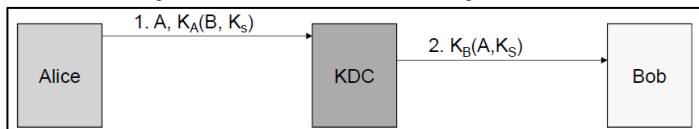


**different sets, ensure session info stays within a session.** This is done in HMAC (Hashed Message Authentication Code - uses one-way hash function). See aside.

This can be addressed by: **making the initiator prove his identity before the responder**, use a **pair of shared keys** (one for initiator and one for responder), use **nonces** from

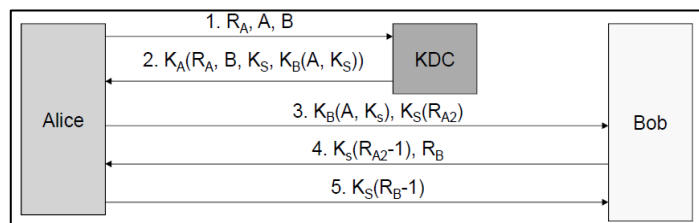


Alternatively, authentication with key centre can be used:



Trusted KDC (key distribution centre) shares private key with each party. Alice contacts KDC indicating her identity and sending Bob's identity and a session key.

However, this is susceptible to a replay attack where Trudy can **replay message from KDC to Bob** and any message that followed it. Trudy is limited to replaying actions of Alice but this may be of benefit. Time stamping can severely restrict the window of opportunity for attack. Can also place a nonce. More sophisticated algo **Needham-Schroeder Protocol** developed.



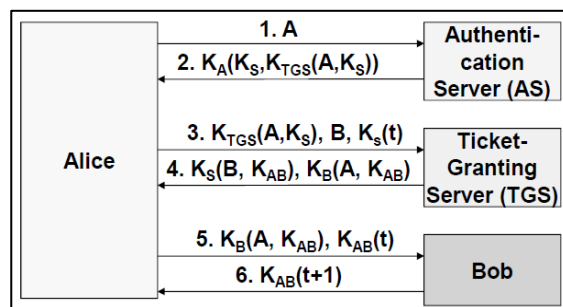
KDC gives Alice's challenge encrypted with Alice key.

Bob receives his challenge, sends it back, again encrypted, and issues his own challenge which is then solved.

**Interception is nullified.**

**Kerberos** is used as a variant to the Needham-Schroeder Protocol. Alice logs into an **authentication server** to be verified as Alice. Then using a ticket granting server to gain a ticket proving her identity. Then she can access Bob.

AS provides a session key in an encrypted message containing ID and session for the TGS. Workstation then asks for password. This generates  $K_A$  to decrypt the message. Alice's password can not be deleted from workstation. TGS is then sent info and returns Bob's identity with a timestamp using encrypted session key. ASs and TGS are distributed in multiple realms.



Or you can just use Public-Key Cryptography (RSA):

