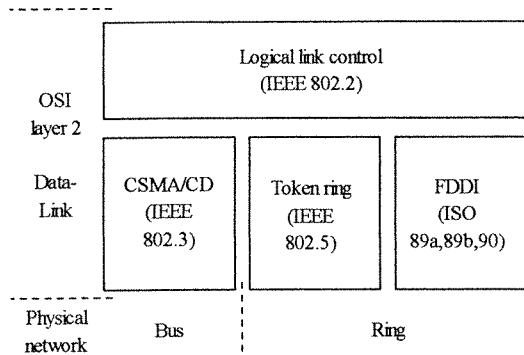


- (a) The data link layer forms layer 2 of the OSI reference model. This provides error detection and correction for a link to ensure that the exchange of data is reliable. Provides error free data to the network layer. Adds control information to the data blocks and a frame check sequence (FCS) for error-checking or bits for synchronisation. Retransmits data if errors have occurred (but it does not arrange for retransmissions). The data-link layer is also responsible for defining the length of data segments to be sent to the physical layer. If the data provided by the network layer is too long, then it is broken into suitable sized packets.



Transparency is preserved for the data bits in the layer 2 frames. Two basic services are offered, *connection oriented* and *connectionless*, usually through the use of a LLC sub-layer. In the case of LANs, the data-link layer provides local area address management on top of that provided by the network layer through medium access protocols (MACs). Different types of LAN are characterised by their distinctive topologies. They all comprise a single transmission path interconnecting all the data terminal devices, with a bit speed typically between 1Mbits/s and 1Gbit/s, together with the appropriate

protocols (called the logical link control (LLC) and the medium access control (MAC)) to enable data transfer. The LLC and MAC split the data-link layer (Layer 2).

The LLC can multiplex amongst higher layer clients through the use of a service access point (SAP) identifier. Both the source client (SSAP) and the destination client (DSAP) can be identified with this header. This is not strictly an OSI compliant function and the information SAP fields are often ignored by the MAC and LLC.

(b) A bridge is a device which allows the interconnection of LANs at the layer 2 level. A *transparent bridge* is a piece of hardware, which allows frames to be passed between LANs that have different geographical locations and even different LAN protocols. A bridge is a data-link layer device and any interconnection of LANs via bridges is often referred to as a *catenet*. A bridge in its true OSI definition is purely a layer 2 entity, which transparently forwards, discards or floods frames to its ports. This pure bridge function (including the bridge address table) is performed by the relay entity in the MAC layer of the bridge. There is no access to this layer from higher layers such as the LLC or network layer. This is often referred to as an architectural bridge. End stations will be totally oblivious to the bridge at the data link layer. Some basic bridge principles:

- There are multiple distinct LAN segments interconnected by the bridge.
- Each station has a globally unique 48 bit unicast address.
- The bridge has a *port* or interface on each LAN to which it connects.
- There is a table within the bridge which maps station addresses to bridge ports, hence it knows how each station can be reached.
- The bridge acts in *promiscuous mode*, it receives (or attempts to) every frame on every port regardless of destination address.
- The bridge must apply the MAC protocol of the intended port. Eg detect collisions on Ethernet or use a token on a token ring.
- A frame can incur a bridge transit delay due to traffic or buffering whilst awaiting transmission onto the intended port.
- When forwarding a frame, it uses the original source address of the sender rather than inserting its own (if it has one).
- The stations are unaware of the bridge, hence it is transparent.

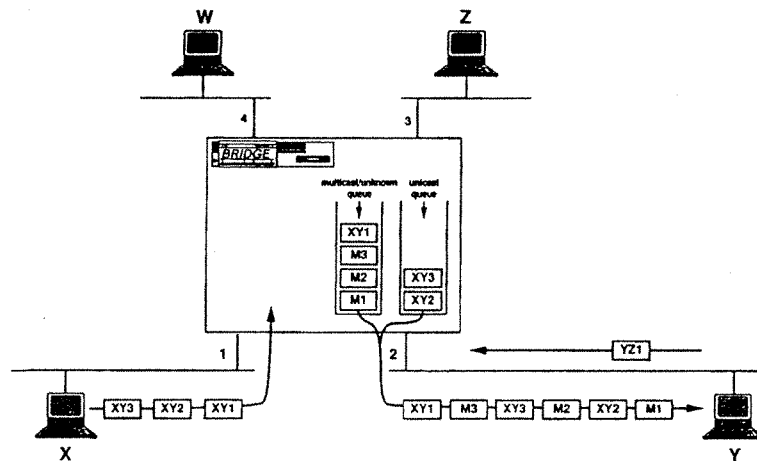
The transparent nature of bridges means that they can be interconnected with each other to make even more complex LAN topologies, without loss of frames or data. There is no limit to how many bridges can be connected in this way, however there are practical limits due to flooding and multicasts as well as bridge delays.

A bridge tries to make a catenet appear transparent to end stations, as if it were a single LAN. Hence higher layer services will expect a LAN-like performance from the catenet below it. A LAN data-link must exhibit certain properties.

Hard Invariants: Non duplication of frames, Sequential delivery of frames

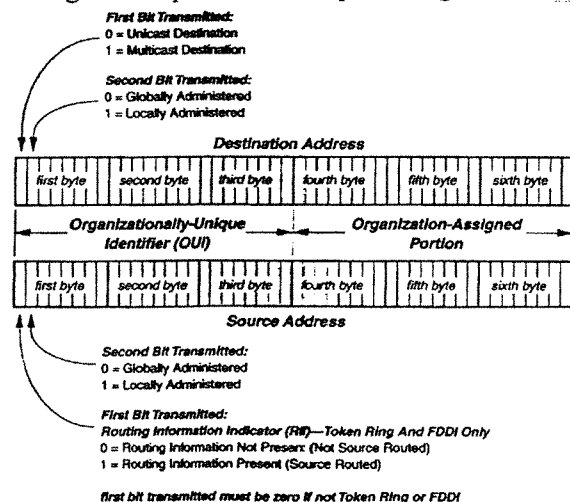
Soft Invariants: Low error rate, High bandwidth (or utilisation), Low delay (or latency)

The hard invariants are absolute and cannot be compromised in any way as they are fundamental to the operation of a layer 2 process in the OSI model. The soft invariants are more flexible and can be traded off for more sophistication in certain areas of the LAN performance. Note that there is no protocol mechanism to guarantee these invariants. Bridges can complicate these restrictions and care must be taken, especially when considering the hard invariants.



As shown above, the bridge receives a frame (XY1) for destination Y from source X, however station Y is currently unknown and so the bridge treats it like a multicast. This results in the frame appearing in the same queue as other multicasts which could be awaiting transmission (M1, M2, M3), hence frame XY1 must wait. While XY1 is in this queue, station Y sends a frame to station Z (YZ1) which is received by the bridge. This YZ1 frame means that the bridge now knows which port station Y is on due to its source address, hence it can update its address table. This means that when the next frame XY2 arrives at port 1, it is queued with the unicast traffic for port 2 as station Y is now known. These two queues for port 2 will eventually be emptied and could lead to XY2 being received before XY1, breaking a hard invariant. There is no clear solution:

- ✓ Use a single queue for both unicast and multicast traffic (complicated)
- ✓ Use a time synchronisation mechanism between queues (slow)
- ✓ Discard the later unicast traffic until frames are clear from the multicast queue.
- ✓ Ignore the problem and hope nothing will break.



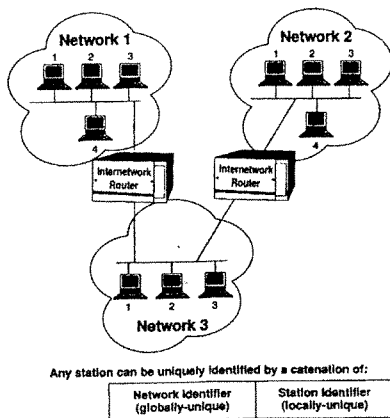
(c) The MAC address has a standardised 48 bit structure which has been adopted as part of the IEEE 802 set of LAN protocols. The address can identify both the transmitting station (*source address*) and the receiving station (*destination address*).

The 48bitMAC address space is split into two halves:

- A *unicast address* identifies a single device or interface. The source address is always unicast.
- A *multicast address* identifies a group of logically related devices.

The first bit of the destination address defines if it is a unicast (0) or a multicast (1) address. Source addresses are always unicast so the first bit is zero except in token ring or FDDI, where it describes how the packet is routed (equals 1 for source routing).

The second bit of the address defines whether the address is globally unique (0) or locally unique (1) to the LAN. Globally unique addresses are assigned by the manufacturer, whereas locally unique addresses are assigned by the LAN administrator. The 48-bit MAC address is divided into two parts. The first 24 bits constitute the organisationally unique identifier (OUI), which indicates which organisation (typically the manufacturer) is responsible for assigning the remaining 24 bits of the address. A MAC address is usually expressed in canonical byte form *aa-bb-cc-dd-ee-ff* where the first 3 pairs of bytes are the OUI and the last 3 pairs of bytes are set by the manufacturer. An address is read *aa* byte first, however there are two bit conventions, little and big endian.



The MAC address for a NIC maker must be globally unique and contain a valid OUI registered to that company. Hence there are 3 bytes left giving $2^{24} = 16777216$ addresses.

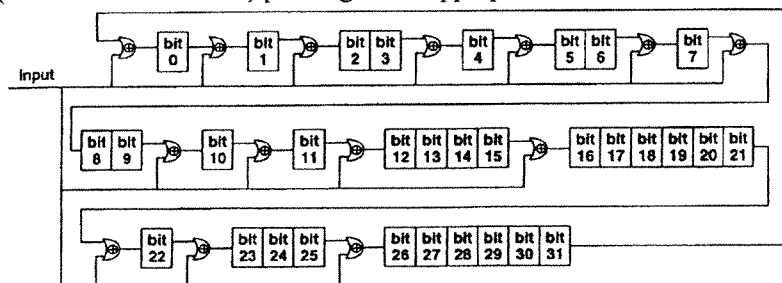
The maker could buy another OUI, or the net software could be adapted to use higher layer address structures as well. Within the data link layer, only stations on the same LAN need to have unique addresses, as more complex network addresses can be made by catenation of a station address with the address of the LAN which contains it. This could be a higher layer address such as an IP address. Eg. [Network 1 | Station 4] can communicate with [Network 3 | Station 4] even though they have the same data-link address.

(d) In order for a bridge to locate a port to which a destination station is attached it must search for it in the bridge address table. This should be as efficient as possible to minimise potential delays in the catenet. There are three main techniques: The hash table, binary search and content addressable memory (CAM).

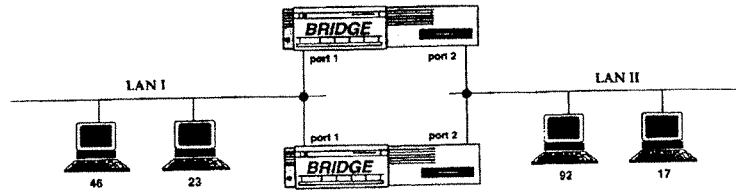
A way to avoid huge search areas is to map the address onto a small pointer space, using what is known as a hash function. Hashing is taking the 48bit address and producing a shorter field, which will then point at a subset of locations in memory. The hash function must:

- ✓ Produce a consistent value for the same address (be repeatable)
- ✓ Produce a relatively uniform distribution of pointers for a given set of inputs.

Hence, a hash function should not cause uneven spreading of addresses to output values. Each output value should have roughly the same number of addresses mapping to it in a random manner. This is important as most MAC addresses on a LAN will be a subset as they will have come from the same manufacturer and have the same 24bit OUI. There are many different hash functions in the literature, however there is an automatic generation algorithm, which will perform the hash function on the data in the form of the FCS generator and checking circuitry. As the frame passes through the FCS circuit, there is a point when the 48bit destination address has passed through it, hence we only need to sample the output from the FCS at this point in time to generate a suitable hashed address. This is usually done with a linear feedback shift register (LFSR). The address table is organised as hash buckets, with the hash function (the bits from the LFSR) pointing to the appropriate bucket.



- ✓ The destination address is simple to calculate, but as the source address follows straight after it, it is not so simple to hash. Either a second LFSR are needed or a software option is required.
- ✓ Depending on the addresses in the catenet, the buckets may not fill uniformly leading to buckets overflowing. This can be altered by changing the bits chosen by the LFSR.



Q2 (a) If we consider the above bridge connection, using ordinary transparent store and forward bridges which have up to date address tables, what will happen when frames are sent? If station 46 sends a frame to station 17, each bridge receives the frame and sends it to the appropriate port (port 2) and after queuing, the frames appear at station 17. Station 17 will receive two frames from station 46 which violate the non-duplication invariant. In fact every station on LAN II will receive duplicate unicast frames from LAN I.

What happens if LAN 46 sends a multicast frame? Both bridge A and B will forward the frame, however bridge A will receive the forwarded frame from bridge B (and vice versa) on port 2. Upon receiving this frame, the bridge will look at the source address of the frame and reconfigure station 46, thinking it is now connected to port 2. This process will continue indefinitely with both bridges continually updating their address tables. The same situation will occur in the first scenario, with non-convergence of the address table in the unicast case as well.

This problem can be fixed by careful network planning, however in complex structures, loops are inevitable. The problem above can be fixed by disabling the ports on one of the two bridges, effectively removing it from the network. It is still listening, but no longer routes frames. This disabling process can be done manually for small catenets, however for large complex structures this is impractical and often dangerous. There are several automatic mechanisms which can be implemented in bridges, the best of which is the spanning tree, which allocates ports on bridges and therefore routes by measuring the speed and efficiency of transport. Hence in the above example the slowest bridge is always eliminated.

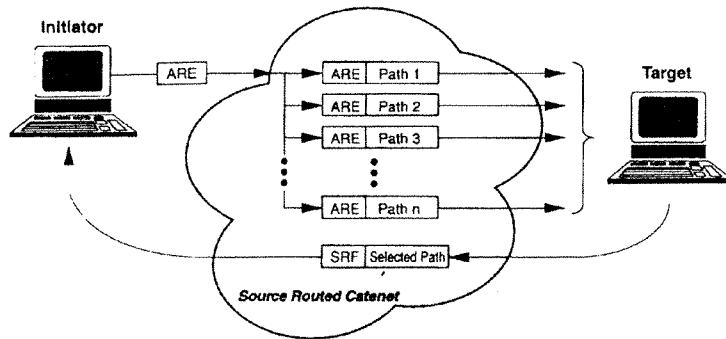
(b) **Routing type** – This field in the source routed frame specifies the type of routing frame used

1. *Specifically routed frames (SRF)* – Carry routing type 0b0XX. A SRF carries a list of route descriptors and is forwarded along this route. This is used for the bulk of data frames.

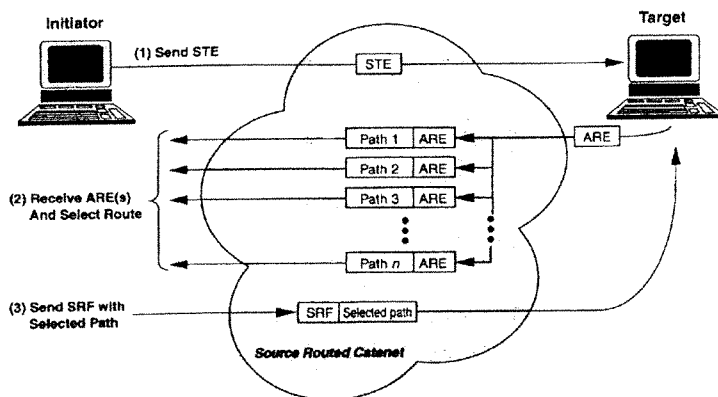
Source frames which are not specifically routed are called *explorer frames* and have the first bit of the routing type set to 1

2. *Spanning tree explorer (STE) frames* – Routing type 0b11X. A SPE frame is only forwarded by bridges that are designed as part of the spanning tree. One copy of these frames will appear on each ring in the catenet (non-duplicated). They can be used for either multicast traffic or route discovery. STE frames are originally sent with no route descriptors. Forwarding bridges insert descriptors to identify the route taken.
3. *All routes explorer (ARE) frames* – Routing type 0b10X. An ARE frame is forwarded by all bridges along every possible path between source and destination. Thus the destination may receive multiple frames for different routes taken. ARE frames are sent with no route descriptors. These are added by each bridge as it forwards the frame.

(c) Before an SRF can be sent, the route discovery process must be done to find a suitable route. There are several possible ways of doing this with STE and ARE frames, but here are two most common.



an SRF for every ARE it receives and let the sender select the route.



The route selection process can be a complex one, as it is unknown how many ARE responses will occur. Is a better route yet to come?? There are two basic mechanisms.

- ✓ *Take the first route* – Just select the first ARE which arrives and discard the rest. The first ARE indicates the route of least delay. This is what the majority of stations will do.
- ✓ *Take the first route which meets a specific requirement* – select the route with a suitable MTU of with a minimum number of bridge hops.

(d) The relative merits of the two types of routing are quite subtle and buried in historical evolution. Source routing was developed by IBM as part of their token ring protocol and is in fact found as part of the IEEE token ring and FDDI standards. The logic behind it was that source routing required more computing power at the stations and IBM were good at making computers... There is no source routing mechanism for Ethernets (Xerox and IBM didn't get on). There are a whole host of different arguments about link efficiency, traffic management, link redundancy, latency control etc which can be used to justify or reject source routing, however a more important lesson is it parallels with the switching protocols used in the synchronous digital hierarchy (SDH) and asynchronous transfer mode (ATM) switching.

Source routing is a more efficient as long as the network can cope well with multicast frames. This may seem contradictory as multicasting is normally associated with clogging up networks. This is not the case in source routed networks, as the bridges are now very simple transfer and record devices with no real intelligence above layer 2, hence the overhead of multicasting is not very large. This is clearly evident in both SDH and ATM, where the excess cost of multicasting is not that considerable.

The main reason in favour of destination routing is flexibility. In a network that is always growing or evolving, the ability to 'learn' the routes can be a big advantage, however this flexibility is at the cost of complexity in both hardware and protocol(s). It is interesting to note that today's destination routed networks all require at least 3 protocols to work efficiently (ie TCP/IP + Ethernet). This is a result of the growing complexity of the requirements of routing frames. This has a huge impact on the hardware as well, with large backbone routers being extremely complex and expensive.

If we consider the internet, then flexibility is more important than complexity, hence destination routing has evolved in the way it has. Source routing would be difficult with so many users and is probably not sufficiently scaleable to be effective. If we consider other issues such as priority services and quality of service across the internet, then source routing could be much more effective.

ARE request, specifically routed response – The sending station send an ARE frame which will arrive at the destination by multiple recorded routes. The receiving station selects a route and sends a SRF back to the sender to indicate the chosen route. This places a processing burden on the receiver, so a variation of this is for the receiver to send

Spanning tree request – The sending frame sends an STE frame which is forwarded by the bridges on the spanning tree on all rings in the catenet and eventually to the target destination. The receiving station responds by sending an ARE frame back to the sender which will indicate to the sender all of the available routes. Hence the processing burden is returned to the sender who must select a route.

QUESTION 1

- a) i) What is a "strictly non-blocking" switching network. Give two examples?
- ii) What is the probability $P(x)$ that x calls arrive in time T in a telephone network in which all call arrivals and terminations are random events and in which the generation of traffic is a stationary and random process? Sketch this probability distribution.
- iii) By analogy with ii) write down the probability that in a packet switch, k packets arrive at a given output port in time t . What does this assume?
- Reduce the mean fraction of packets that make it to the output of a fully loaded, strictly non-blocking switch.
- b) i) What problem does this result pose and how in principle may this problem be reduced?

QUESTION 2

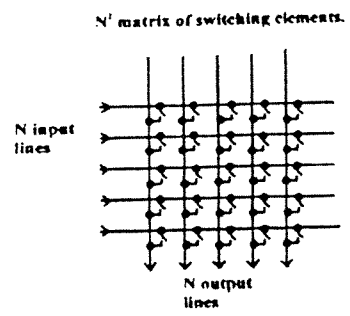
- a) i) Briefly outline the differences between plesiochronous and synchronous transport modes.
- ii) Sketch an STM-1 frame and describe how a 2Mbit/s plesiochronous channel maps into it.
- iii) Describe the functionality of a cross-connect and of an add-drop multiplexer. How do these modules differ in the synchronous digital hierarchy ^(SDH) and in a wavelength division multiplexed (WDM) network.

Question 3

a) In a strictly non-blocking switching network all permutations and combinations of ^{direct} connections can be made without re-arranging existing switch paths, i.e. any available path can be taken

Example 1 A crossbar switch

CROSSBAR SWITCHES.

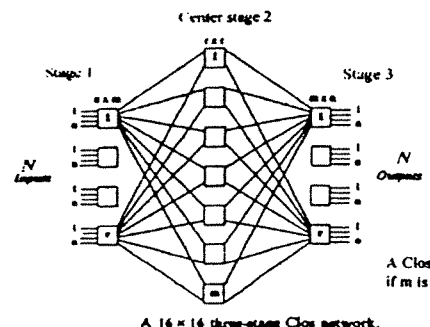


Strictly Non-Blocking.
No shared paths
All N connections can be made independently.
Full broadcast & multicast.

(Router-selector switches can also be considered as topologically equivalent to a crossbar, though in some instances multi-casting is not possible)

Example 2

Multistage Networks: The Clos Network.



A Clos network is strictly non-blocking if m is greater than or equal to $2n-1$.

A 16 x 16 three-stage Clos network.

- N = no. of inputs and outputs ($n \times n \times r$)
- n = no. of inputs per switch in stage 1 & no. of outputs in stage 3.
- m = no. of outputs per switch in stage 1 & no. of inputs in stage 2.
- r = no. of switches per stage in stages 1 & 3.

It can be shown that a symmetrical 3-stage Clos network is strictly non-blocking if $m > 2n-1$

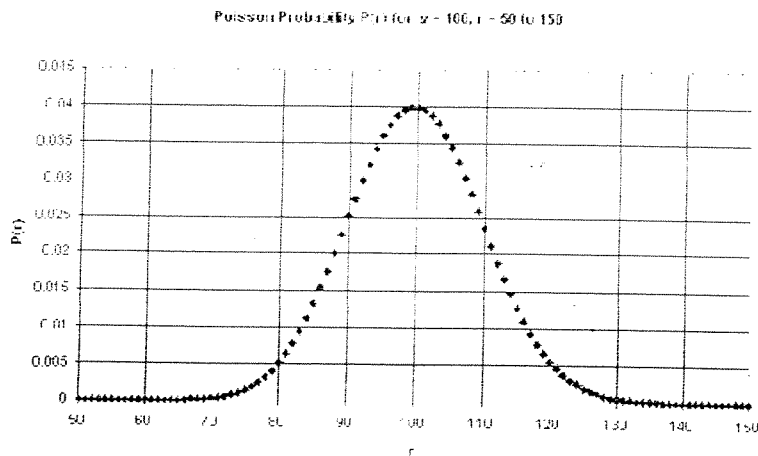
b) Random arrival of calls and statistical equilibrium results in a Poisson distribution

$$P(x) = \frac{\mu^x}{x!} e^{-\mu}$$

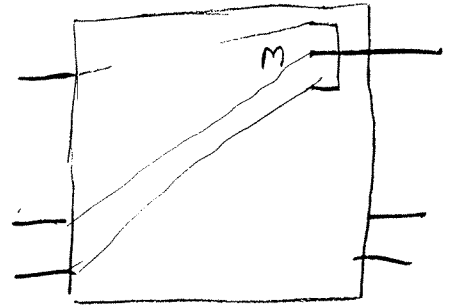
Where x = the number of message arrivals in time T

μ = the mean number of arrivals in time T

Example of Poisson Distribution.



c) Consider a packet switch in which all of the N inputs are copied to every output. However only M packets (or cells) are selected, (the others are "knocked out")



The probability that k packets arrive at a given output port in a time T may be given by the Poisson distribution

$$\text{i.e. } P(k)_{N \rightarrow \infty} = \frac{\rho^k}{k!} e^{-\rho}$$

where ρ = the average number of packets arriving in one time slot (i.e. in time T)

This assumes : ① N is large ($N \rightarrow \infty$)

② $M \ll N$

③ the packets have an independent, uniform, address distribution

The mean fraction of cells that make it to the output is therefore given by the summation over k

$$\rho_{\text{var}}(N \rightarrow \infty) = \left[\frac{\rho e^{-\rho} + \rho^2 \frac{e^{-\rho}}{2!} + \rho^3 \frac{e^{-\rho}}{3!} + \dots}{\rho} \right]$$

$$P_{av} = \left(\frac{1 - e^{-p}}{p} \right)$$

For a fully loaded (i.e. $p=1$), fully non-blocking switch therefore 63.2% of the input traffic gets through and 36.8% is lost.

d) The packet loss rate needs to around 10^{-10} , not 36%

This problem may be alleviated in the design of real packet switches by the following measures

- increasing the number of paths through the switch, i.e. M

- decreasing the switch loading (setting $p < 1$)

- grouping the N outputs in groups of n ($n \ll N$), which are treated as if they were one output.

Further switching must be carried out in the output stages of the switch to direct packets to the correct output within the group of n .

However this grouping allows buffers to be shared and greatly reduces the value of n required to achieve an acceptable rate of packet loss

Question 4

a) In a plesiochronous transmission hierarchy, small differences in the bit rate of different tributary channels are brought to a common bit rate by the addition of dummy information bits (that we called 'justification' bits). This process of 'bit rate adaptation' is necessary to allow the channels to be interleaved to form a higher level time division multiplexed data stream.

e.g. 2 Mbit/s channels (each equivalent to 30 64 kbit/s channels plus control etc.) might be multiplexed up to a 8 Mbit/s data stream.

The plesiochronous digital hierarchy (PDH) has two main problems:

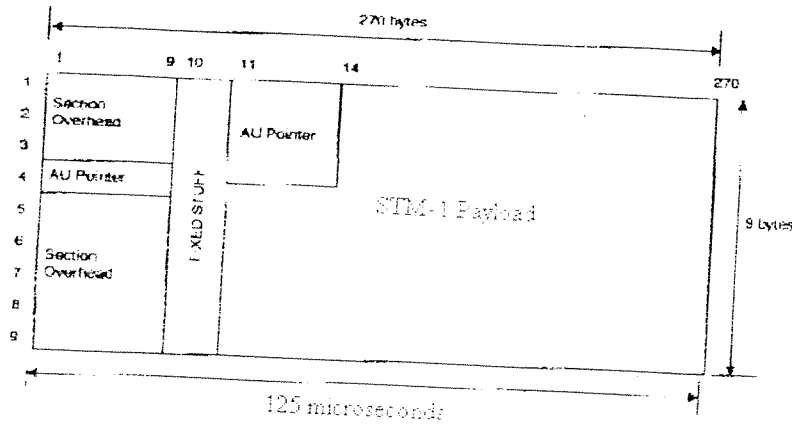
- 1) Justification bits are present at each level in the PDH. This makes it impossible to identify the frames for a single 2 Mbit/s channel in a higher level PDH multiplex.
- 2) There is little provision for network management.

The synchronous digital hierarchy (SDH) solves these problems by:-

- 1) Synchronising all elements to the same master clock to avoid the necessity to justify signals.
- 2) Information about the signals that are multiplexed together is included in fixed known positions in the multiplexed frame, e.g. pointers to indicate where a signal may be found. This accommodates delays.
- 3) Provision is made in the multiplexed frame for monitoring and network management information.

b)

The Synchronous Digital Hierarchy: STM-1 Frame Structure.



Information is packaged into the 155 Mbit/s STM-1 signal as a series of frames that are continuously repeated.

The basic structure is shown above. Each 8-bit byte is equivalent to one 64 Kbit/s voice circuit.

The time slots allocated to a particular plesiochronous rate within such an SDH frame are called "containers". A container that includes some control information is called a "virtual container" or VC.

A 2 Mbit/s PDH channel maps into the STM-1 frame as a "VC 12"

A VC 12 therefore requires 32 bytes, which is equivalent to 2048 Kbit/s. This is used for a payload of 30 voice circuits plus 2 bytes used for justification and overhead.

To accommodate delays VCs must "float" in the frame, so pointers (located at a fixed position in the frame) are used to indicate the start of a VC in the frame. (i.e. A VC with its pointer is a "tributary unit").

c) Cross-connects are used to reconfigure a network by providing semi-permanent interconnections. They are controlled by the network manager and their purpose is to manage the network efficiently and provide new services.

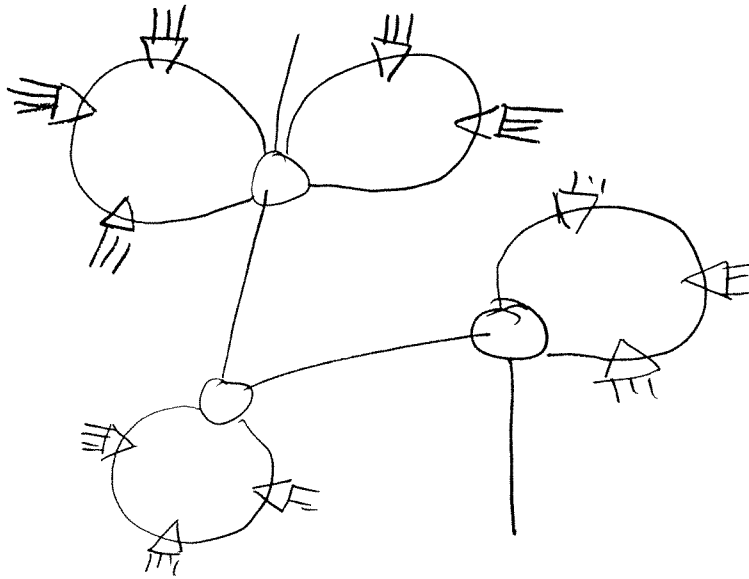
Add-drop multiplexors allow channels to be added or dropped at nodes or stations in the network. These are the two building blocks of optical networks. (Fig A)

In the synchronous digital hierarchy these are essentially electronic modules fitted with a wide variety of optical and electronic interfaces.

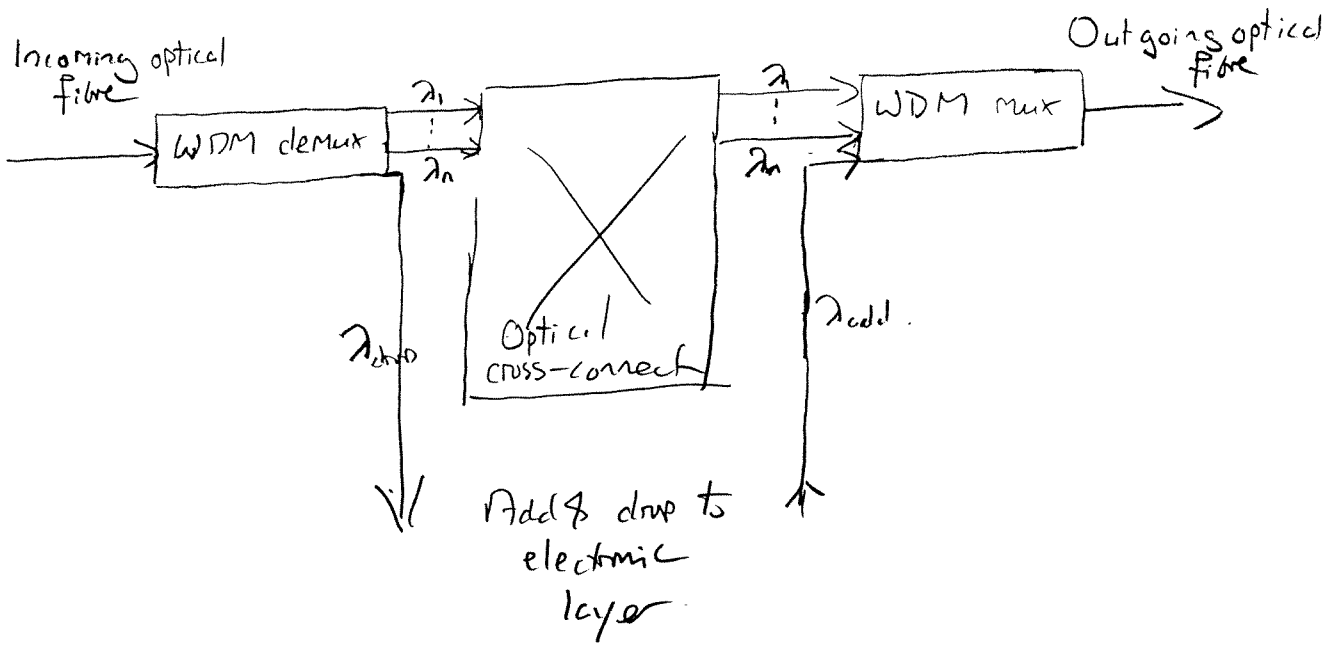
SDH cross-connects manage and reconfigure the network at the level of virtual containers (VCs). This is substantially a time domain operation i.e. it involves time-slot re-arrangement as well as some space switching. e.g. cross-connects connect VC4s (140 Mbit/s) or VC3s (2 Mbit/s) in STM-4 data streams (622 Mbit/s).

SDH add drop multiplexors are designed for a wide variety of tributaries e.g. synchronous, plesiochronous, ATM or from LANs. All may need to be added or dropped from high capacity optical aggregates.

In a similar manner WDM networks may also built from cross connects and add drop multiplexors. However these are optical cross connects that switch individual wavelength channels. They operate at the highest levels in the present telecommunication transmission network. (Fig B)



A) A network built from cross-connects (O)
and add drop multiplexers (◀E)



B) A fully optically transparent cross-connect which
does not have wavelength conversion