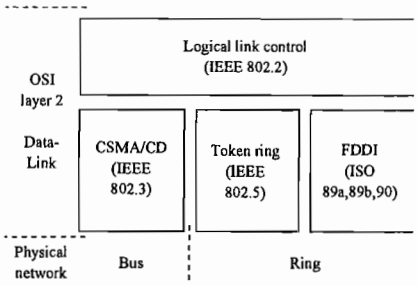
2009 PART IIB 4B15 ADVANCED TELECOMMUNICATION NETWORKS

Dr T D Wilkinson

4b15 cribs – more verbose than expected

Q1 a) The 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)



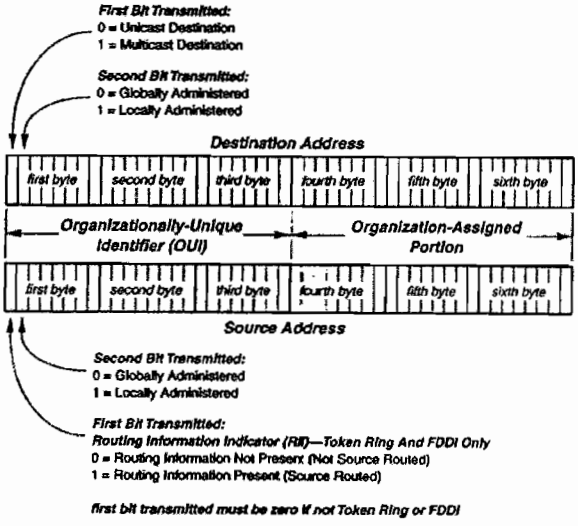
Most LANs conform to one of the different types specified in the IEEE 802 or ISO 8802 series of standards. IEEE 802.2 defines a logical link control (LLC) protocol that can be used with any of the below. IEEE 802.3 defines a MAC protocol called carrier sense multiple access with collision detection (CSMA/CD) which may be a bus or star topology. IEEE 802.5 defines a MAC protocol suitable for use on a token ring topology.

The LLC provides direct (OSI layer 2) connection between any two end devices locally connected to the LAN. It made sense to define a standard interface between the LAN and the application intended to communicate across it. This standard is the logical link control (LLC) defined by IEEE 802.2 or ISO 8802.2.

LLC Type 1 – *Connectionless service*. A simple best efforts delivery with no flow control. The only service is the ability to multiplex the Data-Link to higher layer clients.
LLC Type 2 – *Connection oriented service*. This was derived from the high level data-link control protocol (HDLC). This uses a series of control primitives such as those in X.25 for acknowledgement, retransmission and flow control using fixed length sliding widows of up to 8 frames. LLC-2 systems use LLC-1 to set up connections.
LLC Type 3 – *Acknowledged connectionless service*. This is really a bit of a bodge between LLC-2 and LLC-1 designed to use a mechanism known as 'request with response' that was part of the IEEE 802.4 Token bus protocol. Its is not very common.

The LLC protocol uses an associated frame format which is defined as part of the IEEE 802.2 standard The frame header is either 3 or 4 bytes long for LLC supervisory or information frames.

b) The key to MAC layer protocols is a uniform addressing structure so that LAN hardware can easily identify stations on the network and transmit packets between them. By definition, a LAN MAC address needs only be 'locally unique', especially as it resides within the data-link layer, where routing protocols are not strictly part of the OSI model. Since the development of the 10Mbit/sec Ethernet in 1979, there has been a standardised 48 bit address 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*) and are usually referred to as the *MAC address* of the station on the LAN.



The 48bitMAC address space is split into two halves: A *unicast address* identifies a single device or network interface. When frames are sent to a single station, the unicast address is used as the destination address. The source address is always unicast. Often referred to as *individual, physical* or *hardware addresses*. A *multicast address* identifies a group of logically related devices. They provide a means of one-to-many communication allowing multiple destinations to be addressed by a single communication. Often referred to as *group* or *logical addresses*.

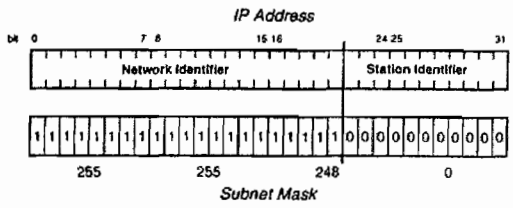
The first bit of the destination address defines if it is a unicast (0) or a multicast (1) address. Source addresses are always unicast so the first bit is zero except in token ring or FDDI, where is describes how the packet is routed. The second bit of the address defines whether the address is globally 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. A 48-bit address allows for approx 281 million million addresses.

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. If a manufacturer wants to build devices with globally unique addresses, it first must obtain and OUI from the IEEE. 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 as discussed later.

c) 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. Eg. [Network 1 | Station 4] can communicate with [Network 3 | Station 4] even though they have the same data-link address.

The problem with this system is that there was no standard way of assigning a MAC address to a particular LAN. A bridge could have an address but there was no logical way of searching for the address as each MAC address is globally unique. The combination of OUI and PiD severely limits the structure of the address space created by the LAN addresses set, making searching very inefficient.

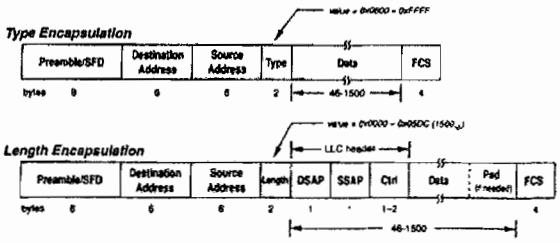
A better way was to encapsulate the MAC address with a layer 3 address which has the correct properties for separating networks from stations. A mapping such a ARP can then be used to map the higher layer address onto the MAC address using a local cache. A good example is the IP address. IP (version 4) addresses are 32 bits long and in the older internet system there were five different classes of fixed IP address. This has been superseded with more dynamic IP allocation system to conserve the numbers of IP addresses in use at one time. An IP address contains fixed-length fields which comprise of two main portions:



- ✓ The network identifier, which indicates the network on which the addressed station resides.
- ✓ The station identifier, which denotes the individual station within the network to which the address refers. IP station identifiers are locally unique, only being meaningful in the context of the identified network.

Each IP address has an associated subnet mask of the same length (32 bits). The network identifier portion of the address is defined by the portion of the subnet mask set to 1's. The rest is the station identifier. The convention is that the network bits and the station bits are set in a contiguous fashion. The contiguous subnet can be stored as a 5bit word, which greatly simplifies the router look up process.

d) An Ethernet frame can take two possible forms. The preamble/SFD, address and frame check sequence (FCS) are common to both types. They are referred to as *type* and *length encapsulation* frame formats. Due to the requirements of back compatibility Ethernet has expanded beyond the LAN into the MAN and WAN. When installed as part of a structured cabling scheme (nowadays the most common realisation of Ethernet), twisted pair or coaxial cabling provides for the transmission medium. Multiple twisted pair cables are usually installed in each individual office and near each desk, and are wired back to a wiring cabinet. Next to the wiring cabinet is a LAN hub which replaces the coaxial backbone, so the arrangement is often called a collapsed backbone. The bus topology still exists, but only within the hub itself. If new devices are added, then it is patched through the wiring cabinet.

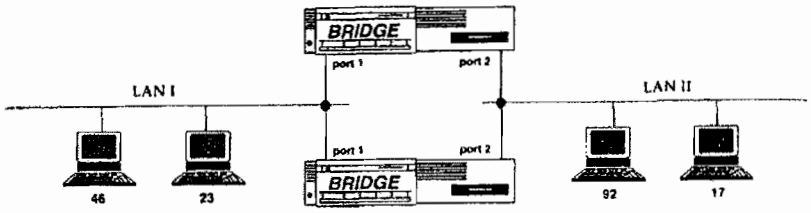


The implementation of Ethernet across WAN and MANs uses very few of the original features. All of the access control mechanisms are gone along with the features of LLC (including SAPs as this is done by a higher layer such as TCP or SIP). All that remains is a data payload, addresses and FCS plus delimiters. This leads to very efficient low latency transmission with protocol compatibility guaranteed all the way through the WAN from LAN to LAN. Potential problems – 1500 byte MTU limit, no latency control or QoS.

Token ring has not been taken up due to the rise of Ethernet, however it does offer some very desirable features, such as inherent source routing capabilities, priority, reservation and QoS mechanisms as well as data link monitoring functions.

Q2 a) Interconnecting LANs and stations is not without its perils. The methods outline so far have assumed we started laying out the network from scratch, however this is very rarely the case, as LANs tend to 'evolve' from previous

incarnations. Because of this process of evolution, the topology or interconnection of the LANs and stations could be sub-optimal, which often leads to loops in the network.



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.

So how can this happen, surely such situations are obvious??

- Historically bridges were slow, so the temptation was to double up bridges to speed up network operation.
- Complex networks are often very difficult to keep tabs on especially over several years of change and evolution.
- Redundancy is often added to a network by planners to guarantee bandwidth at a given cost.

Since loop contention has been an issue for many years, a protocol has evolved to remedy it by inter bridge communication and a tree structure. The original spanning tree protocol (STP) was designed by DEC and later became an IEEE standard (IEEE 802.1D).

1. **The tree topology.** Think of a tree, it has a root, branches and eventually leaves which we can apply as an analogy to the STP system. There are no loops in a tree, all links between the root and the leaves follow a single path which encompasses all of its parts.
2. **The root bridge.** Just as in a tree, we have a *root bridge* from which the remainder of the tree branches out. It is the logical centre of the tree, but not necessarily the physical centre of the tree.
3. **Designated bridges.** A simple way to avoid loops on catenets is to make sure only one bridge is responsible for forwarding traffic from the root onto any branch or link. If there is only one link for any station or group of stations, then loops are automatically avoided. The bridge responsible for forwarding from the root is a *designated bridge*.
4. **Designated ports and root ports.** For a given designated bridge there are three types of ports:
   - A designated port. This is a port in the active topology used to forward traffic away from the root onto the link(s) for which the bridge is the designated bridge.
   - A root port. This is the port in the active topology that provides connectivity from the designated bridge toward the root. There is one exception (the root bridge has no root port).
   - All other ports of a designated bridge will be inactive (disabled or blocking) in the steady state. A bridge only forwards data on its root and designated ports.
5. **Bridge identifiers and port identifiers.** In order for bridges to properly configure, calculate and maintain the STP, there needs to be a way of uniquely identifying each bridge in the catenet and each port within each bridge.
6. **Links and link costs.** Each port on a bridge connects to a link, which could be a high speed LAN or wide area connection. The STP attempts to configure the catenet such that every station can be reached at minimum cost, where the cost is set to be inversely proportional to data rate on the links.
7. **Path cost.** In order for the STP to select paths from the root to a station, a path cost is used. This is the sum of all the link costs used to transmit the data across the desired path.

The above system would eliminate on of the bridges using the STP protocol, making it designated. The other would wait as a standby bridge in case something goes wrong.

b) The STP operates on the principle that all designated bridges (including the root) advertise their current understanding of the spanning tree and their internal state by emitting, on a regular basis, through their designated ports, configuration messages (encoded as bridge protocol data units (BPDU)). All bridges listen to these configuration messages and compare them with their own internal information. When a bridge internally feels it has a better claim to be root or a designated bridge, it initiates a topology change. The regular transmission of these control messages maintains the steady state. If a

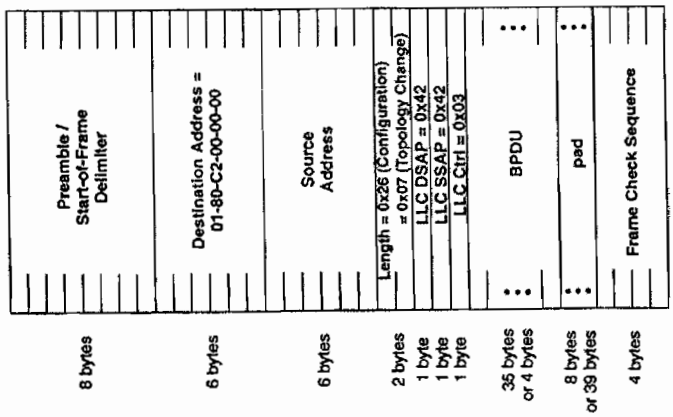
link fails and the messages stop, then previously inactive ports may become active to re-route data. In the steady state, the STP follows:

1. Once every Hello time (usually 2 seconds), the root bridge transmits a configuration message encoded as a BPDU. This identifies the root bridge.
2. All bridges sharing links with the root bridge receive the message and process it internally. BPDUs are never forwarded.
3. Designated bridges (or those preparing to be designated) use the information received from the root bridge, and update the identifier, path cost and port identifier and then transmit it out of its designated ports.
4. This process will repeat from bridge to bridge until there are no bridges left before the stations.

Every bridge receiving the configuration messages compares the information received with its own internal state and knowledge. Specifically the bridge compares:

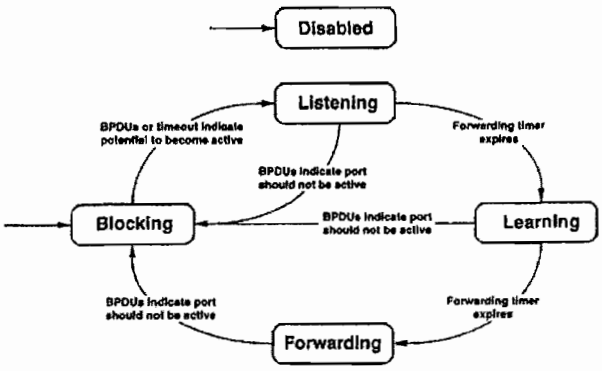
- The root identifier with its own identifier. If it is numerically lower then it initiates topology change and transmits configuration messages with itself as root.
- The path cost in received messages to the path cost available to this bridge through any other ports. Hence if it can offer a lower cost path, then it initiates a topology change. If costs are equal then it compares bridge identifiers or port identifiers as required.

BPDUs are a special frame format used to transmit STP information between bridges. The information is included in a series of frame fields, including timing information. The BPDU includes both a protocol identifier (the STP takes the value 0x0000) and a version number (currently 0x00) to allow future expansions.



When transmitted on a LAN, BPDUs are further encapsulated in MAC frames using LLC type 1, with a DSAP and SSAP of 0x42 (!!!) The MAC source address is the MAC address of the port through which the frame is being transmitted. The MAC destination address is the multicast address 01-80-C2-00-00-00. The use of a multicast address means that the source bridge does not need to know the destination address of other bridges. The multicast address is in the range 01-80-C2-00-00-00 to 01-80-C2-00-00-0F which are addresses defined not to be forwarded by IEEE 802.1D compliant bridges.

c)



Each bridge port can be in one of five possible states.

- **Disabled.** When disabled, a port will neither receive nor transmit frames or STP messages. This is usually due to a port fault or duff link.
- **Blocking.** A port that is enabled but is neither designated nor root port will be in the blocking state, which is in standby or not currently required by the STP. It will not receive or transmit data frames nor transmit BPDUs, but it will listen for others BPDUs (or time out if none heard) to determine if and when the port should be consider becoming active in the spanning tree. The blocking state is usually entered on power up. Upon learning (by either BPDUs or time-out) that it needs to become active, it will go through three stages: listening, learning, forwarding.
- Listening. In this state, the port will not forward data, but is listening to (and possibly sending) BPDUs to determine the spanning tree. If the port decides it is not suited to becoming designated it will return to the blocking state.

- Learning. In this state, the port is preparing to start forwarding data. Since the ports address table will be empty, it needs to wait (known as the forwarding delay) as a port with an empty table will flood all data frames.
- Forwarding. Once the bridge has spent time learning addresses, it is allowed to forward data frames. This is the steady state for an active port on the spanning tree.

If the topology changes it is usually due to component or link failure, management intervention or network evolution. STP treats these as boundary events and can be slow to react to changes. Hence a smoothing procedure is employed to prevent possible shocks.

- Provides for explicit topology change and notification.
- Provides for acknowledgement of the changes.
- Uses timers to:
  1. Prevent rapid transition between blocking and forwarding states. Hence minimise transient loops
  2. Allow bridges to participate in the election of new designated bridges or ports. This prevents recursive changes

Before transitioning to the forwarding state as a result of a topology change, a port will wait for the topology change information to propagate through the catenet.

When a bridge that is not the root changes the active topology of the catenet, it transmits a topology change message through its root port. This is repeated until the bridge receives an acknowledgement from the designated bridge from that link (through the TCack bit). The designated bridge similarly transmits a topology change message through its root port until the message ultimately reaches the root bridge. When the root bridge is informed of the change, it sets the topology change bit in the field for some time so that all bridges become aware of the change.

d) Three protocols which use the STP.
1) Source routing systems - 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.
2) Non-adaptive routing - in order to understand routing, we must understand the optimality principle as this is how routes are decided. The principle states that if J is on the optimal path from router I to router K, then the optimal path from J to K also occurs along the same route. As a direct consequence of this, the set of optimal routes from all sources to a given destination form a tree structure rooted at that destination. This is referred to as the sink tree which is effectively the result of a spanning tree.
3) MPLS - The purpose of label based and MPLS systems is that they strive to avoid the address table look-up in each router. The basic principle behind MPLS is that an extra field is added to the front of the packet, normally by modifying the PPP frame format. This field contains a label, which identifies the VC and gets the flow of packets on to the next router in the VC in a given QoS. Hence each flow appears as a single large packet with a common defined VC to a given destination. Each VC is determined by a very complex route discovery and maintenance process not dissimilar to that used in source routing systems (spanning tree).

Q3 a) The network layer of the OSI reference model is predominantly concerned with the moving of packets from source to destination. This may take many hops between different layer 3 devices (routers) along the way. This requires a more sophisticated approach than that used at layer 2 so far as the network layer provides end to end transmission across the entire network. A subnet is defined as a sub-network of all routers, often belonging to a single company or network provider. We will assume a station on LAN connected to a router, which is part of a group of routers which forms a subnet, connecting via the subnet to another station on the other side of the network.

All of the routing and switching discussed so far has assumed that the bridge switch or router will have some form of transparency and that the switch has full knowledge of the network and its arrangement within its destination address look up table. This is the technique which suits connectionless systems like LLC-1 Ethernet and the internet. This process is a very efficient way on setting up a subnet and it is inherently scalable assuming the relevant technology and a suitable addressing structure to allow coarse and fine searches. The main drawback of this approach is that there is a lot of complexity and cost assumed at each router node in the subnet. The routing table is a major asset and problem as the larger the network, the more entries it must contain and the longer it will take to search efficiently.

There is however, another way... It is possible to configure and operate a network, which has no knowledge of its interconnection or the positions of stations upon it. All of the route configuration and set-up processing is performed by the stations at either end of the route or link(s). This is referred to as source routing. The roles are completely reversed in source routing. The end stations are totally responsible for the transportation of traffic across the subnet and the nodes are totally unaware. Essentially source routing is a connection oriented process, with a call setup procedure followed by data transmission. The chosen path is then used for the entire transmission process as if it were connected via a virtual circuit. Once the data transmission has been completed, the path is closed and no record of the route is kept by either end station.

One of the main advantages of source routing is that there is no address table required as the route is embedded within the packet itself, hence it is a very fast and efficient procedure to manage and maintain across the network.

b) Routing algorithms are set into two classes: nonadaptive and adaptive.

Nonadaptive routing algorithms do not base their routing decisions on any measurements or estimates of traffic or topology. The routes are calculated off-line and then downloaded to the router at boot-up this is often called static routing. This is often favoured by cheaper routers which only route over a small subnet. It is inherently inflexible as the routes are only as good as the table downloaded, however it is cheap simple and stable to implement. It is likely to be found in routers close to the end stations such as those found in LANs as the system does not change quickly nor need to adapt fast. There is also little need for global routes as more distant nodes will be routed by larger routers.

Adaptive routing algorithms, in contrast will update their list of routes dynamically to reflect the current snapshot of the network performance, including both topology and traffic conditions. Information is exchanged between routers to build up a picture of the network's performance and structure. This is a very versatile system but requires a lot of sophisticated electronics and algorithms to maintain the router table locally. It is also vulnerable to attack and instability with sudden large changes in node numbers.

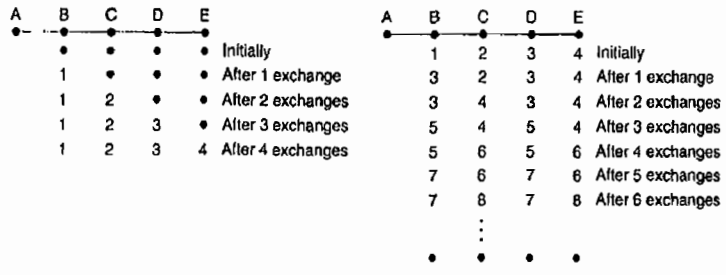
In order to understand either type of routing, we must understand the optimality principle as this is how routes are decided. The principle states that if J is on the optimal path from router I to router K, then the optimal path from J to K also occurs along the same route. As a direct consequence of this, the set of optimal routes from all sources to a given destination form a tree structure rooted at that destination.

c) Modern routers use more adaptive algorithms than those used for more static subnets such as the shortest path. This is because graphical algorithms are slow to adapt the changes in the routing conditions. *Distance vector routing* algorithms operate by having each router maintain a table giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbours. This algorithm is often referred to as the *Bellman-Ford* or *Ford-Fulkerson* algorithm and was the basis for the original ARPANET experiment and later became the *routing information protocol* (RIP).

In distance vector routing ach router maintains a routing table indexed by and containing one entry for each router in the subnet. This entry contains two parts, the preferred out going line for that router and an estimate of the time, distance or delay to that router. The router is assumed to know the distance to its neighbours. If the distance is hops, then they are all 1 hop away, if it is queue length, then it measures the queues to each line. If the distance is either delay or transmission time, then the router uses special ECHO packets which contain timestamps to record transit times.

The problem with this algorithm is that although it converges the optimal route, it may do so very slowly. This is referred to as the count to infinity problem as it reacts quickly to good news by slowly to bad news. If we consider the 5 node linear subnet below where the distance metric is number of hops. Suppose A is down initially and all the others know this (ie record number of hops as infinite). When A comes back up, the other learn about it through vector changes. Initially only B knows that A is up, but after four time periods all four routers know A is running.

If we consider the same subnet, but with A running and then crashing without warning, hence the line to B is cut. After one timeslot B has not heard from A, however it does hear from C that A can be reached after 2 hops, even though this route passes through B itself. Hence B thinks it can reach A via C. D and E do not change. On the next exchange C now thinks from its neighbours that A can be reached by 3 hops through either B or D and chooses a random one with a link metric of 4. Hence we see why the fact that A is down will take a long time to propagate. All routers will count eventually to infinity (however that is defined). There have been attempts to fix this problem such as RFC 1058: *split horizon with poisoned reverse*, but none work well in general as a router has no way or knowing if it is on a path itself suggested by another router.

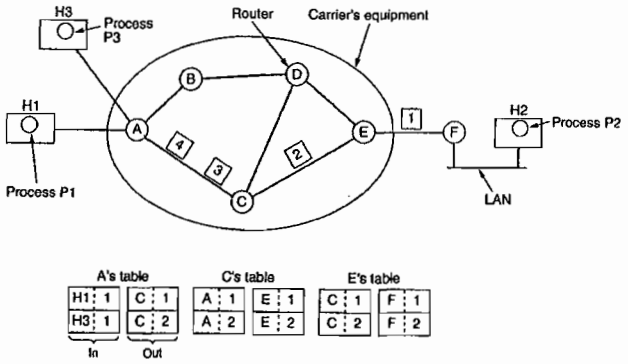


Distance vector routing was used in ARPANET until 1979, when it was replaced by *link state routing*. Variants of this are still widely used today. The process in each router can be described in 5 stages:

- Discover its neighbours and learn their network addresses
- Measure the cost or delay to each neighbour
- Construct a packet telling all learned in 1 and 2
- Send the packet to all other routers
- Compute the path to all other routers.

Essentially this is a process of mapping the entire network in terms of its performance and then choosing the best path with an algorithm such as Dijkstra's optimal paths.

d) The purpose of label based and MPLS systems is that they strive to avoid the address table look-up in each router. In fact they get perilously close to a virtual circuit (VC) as would be defined in the older X.25 or ATM. They are often referred to as tag switching systems as well. The basic principle behind MPLS is that an extra field is added to the front of the packet, normally by modifying the PPP frame format. This field contains a label, which identifies the VC and gets the flow of packets on to the next router in the VC in a given QoS. Hence each flow appears as a single large packet with a common defined VC to a given destination. Each VC is determined by a very complex route discovery and maintenance process not dissimilar to that used in source routing systems. Hence a MPLS router will receive a frame, read its label and look it up in a simplified table to determine the next hop in the route. The VC label is not always constant across the route as it may have to be updated in order to manage similar MPLS packets using the same routes from different flows.



One of the advantages of link state routing is that once the routers have each discovered their neighbouring routers and set up an optimal plan of the internetwork, this data can be used to assign combinations of routes to different MPLS labels and VCs to speed up the process. The edges of the MPLS network then contain the combined routes and are used to translate addresses at both ingress and egress from the network. The central nodes are then effectively bypassed from the main address management process and just maintain and pass on corresponding labels and VCs.

This is in fact a rather convoluted and inside out implementation of the source routing procedures. The rout discovery process can in fact be totally managed by the ingress and egress point to the network and hence can greatly simplify the complexity of the central network nodes as they no longer need to perform the link state management processes and more,