2008 IIB 4B15 ADVANCED TELECOMMUNICATIONS NETWORKS DR T D WILKINSON

4B15 cribs – please note they are more verbose than expected in the exam.

Q1 a) A *transparent bridge* is a layer 2 OSI compliant 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*. Such a device on the physical layer is referred to as a *repeater* and on the network layer as a *router*. 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.

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.

b) When a frame is received on any port, the bridge extracts the destination address from the frame, looks it up in the table and determines the port to which the address maps. If the looked up port from the table is the same as the one on which the frame arrived then the frame is discarded, as it assumes that the station on that port will have already received the intended frame. This is referred to as frame *filtering*.

If the frame received by the bridge is mapped to a different port than the one it arrived on, it then forwards the frame to the port and onto the appropriate LAN. Within this forwarding process, the following must be observed by the bridge.

- 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 whist 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.

c) i) Frame is on the same LAN as the sender, hence the bridge reads the frame, updates its address table and assumes the frame has been received.

ii) The bridge receives the frame, looks it up in its table and forwards it to port 3. It also updates its address table from the source address.

iii) The frame is not able to find it within its address table. There are two options at this point: Discard the frame, or send the frame to all ports except the one it arrived on (*flooding*). There are three possible locations for the mystery station.

a) The station could be on port 1, unbeknownst to the bridge
b) The station could be non-existent, unbeknownst to the bridge
c) The station could be on another port, unbeknownst to the bridge

If we assume that the traffic on the LANs and bridge is less that 100%, and that the occurrence of a unknown destination is not too common, then the process of flooding does not carry a significant overhead penalty. However, in case a) flooding does not help as port 1 is not flooded, in case b) it also does not help as the station is not present to receive the frame. It does help, in the case of c). Similarly, if the bridge receives a destination address, which is a multicast address, then it will flood the frames onto all ports, which will be received by those stations on the multicast.

d) In order to allow efficient frame transmission across different protocols, the following must be considered: The access control method, The frame format, The frame semantics (meanings of each field), The allowable data length, Bit ordering

In order to make such a translation possible there are a few mechanisms which must be adhered to in order to prevent chaos from breaking out.

✓ Access control for each LAN is unchanged. CSMA/CD or token passing is normal.
✓ Any special control frames such as tokens or management frames are not passed over the bridge. The bridge will only relays data
✓ The bridge does not have any privileged status on the LANs.

As can be seen in the above table, when data is passed from one MAC to another by the bridge, it must discard and add certain fields which will be expected by the new MAC protocol. Eg when forwarding to a token ring an access control field must be added with token and monitor bits set to zero. Priority is often ignored, however it is possible to create a new high priority token on the token ring network based on higher level functions set within the bridge. In a similar fashion with the token ring and FDDI MACs, the frame access fields and the end delimiter/frame status fields must be created.

Ethernet	Token Ring	
	Access Control	

	Frame Control	
Destination Addr	Destination Addr	
Source Addr	Source Addr	
Length/Type		
Data	Data	
FCS	FCS	
	End delimiter/ Frame status	

Data encapsulation must also be considered carefully.

✓ Ethernet supports native type field encoding. This use a type value to describe higher layer protocols.

✓ Any LAN can use LLC encoding, which involves a SSAP and DSAP being set.

✓ Any LAN can use SNAP encapsulation, which uses a standard header with fixed SSAP and DSAP values (0xAA) and allows expansion of the SAP space using an OUI field and a PiD field.

One of the biggest problems in translating between different LAN protocols is due to the bit ordering used. All LAN protocols send bytes from left to right, however Ethernet transmits each byte in the bit order LSB (bit 0) first which is referred to as little endian format. Token ring and FDDI both transmit bytes in the order MSB (bit 7) which is known as big endian format.

c) Source routing inserts extra fields into the MAC frame, hence it is necessary to indicate presence of a source routed frame. This is done by using the first bit of the 48bit source address in the MAC frame, which normally defines unicast or multicast address format. This bit is normally wasted as multicast source address is meaningless, hence it is used to indicate (set to 1) a source routed frame. This is a good filtering bit for LANs such as Ethernet where source routing is not permitted.

The is no simple solution to this purely at layer 2. The only real way to resolve this problem is to use a layer 3 switching mechanism to overlay the MAC protocols, effectively making them redundant. This is still not trivial as most Ethernets sit below layer 3 protocols such as TCP/IP, however a lot of Token rings were used under Appletalk. Compatibility at layer 3 is still not guaranteed for all possible combinations.

Q2 a) A bridge is a purely layer 2 device, which indicates a bridging function being done at or near wire speed. As with the technological evolution of the bridge into the switch, the advance of technology has allowed the use of network layer routing or *layer 3 switching* to be implemented as part of a LAN. 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. In order to understand how this works, we need to understand the importance of a *subnet.* A subnet is a group of networks within an internet of networks. It is the beginning of a tiered system which allows more complex and larger scalable networks to be implemented. A subnet is defined as a sub-network of all routers, often belonging to a single company or network provider.

The key part of IP is the IP address, as it is globally unique and also capable of defining both the network to be connected to as well as the station within the network.

b) 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.

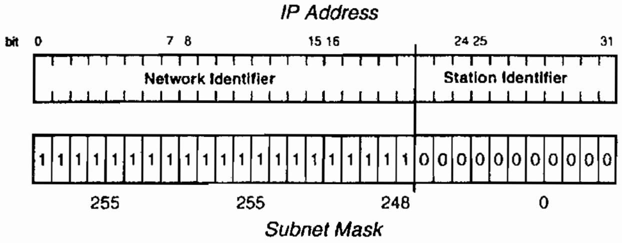
Class A: Sets the first bit as 0, bits 1 to 7 as the network ID and bits 8 to 31 as the host ID. This gives 126 networks and approx 16 million devices on each network.

Class B: Sets the first two bits as 10, bits 2 to 15 as the network ID and bits 16 to 31 as the host ID. This gives 16382 networks and 65134 devices on each network.

Class C: Sets the first three bits as 110, bits 2 to 23 as the network ID and bits 24 to 31 as the host ID. This gives 2 million networks and 254 devices on each network.

Class D: Sets the first four bits as 1110 and is used for broadcasting and multi-cast addressing

Class E: Sets the first five bits as 11110 and is for future use.



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 subnet mask can also be used to allocate a third field in the IP address which denotes a physical subnet section to the station identifier. This allows a group of stations to be located via one subnet mask and then the final section of the station identifier used to locate the individual stations.

A CUED example: Class B address, network identifier 129.169

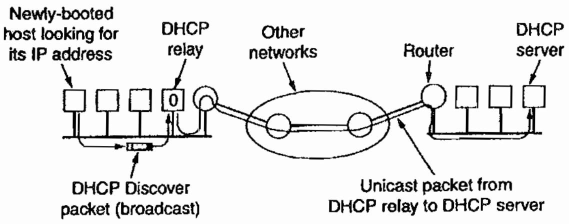
Photonics and sensors group, Inglis 2nd floor subnet mask 255.255.252.0 binary for 252 = 11111100, this means that the first 6 bits of the 3rd byte will define the subnet identifier, and the last 10 bits defines the station.

c) IP (version 4) addresses are 32 bits long and in the older internet system there were five different classes of fixed IP address. Projections of the address space indicated that this limited structure would lead to the running out of addresses by the year 2000. Two mechanisms were proposed to fix this shortage:

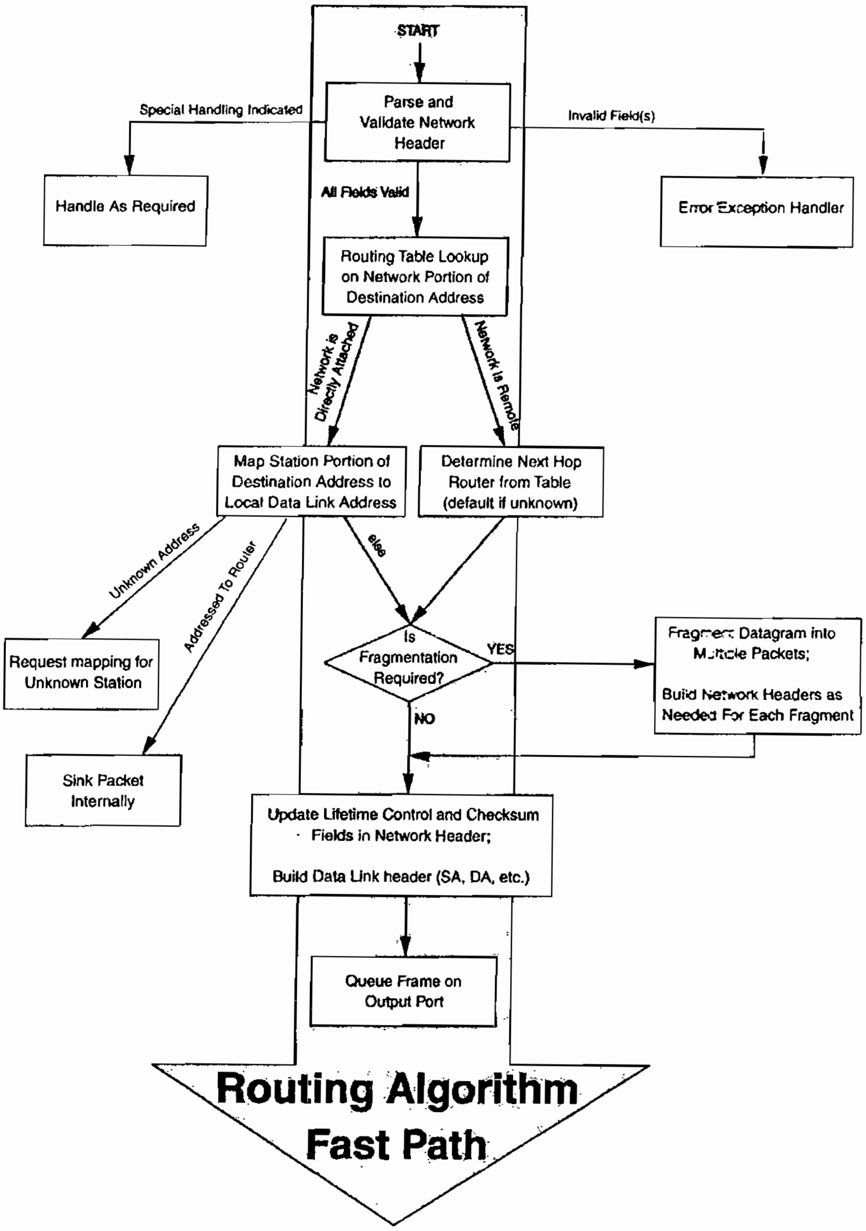
The first was to go to *IP version 6* which along with a host of more sophisticated link management procedures also includes a 64 bit back compatible with version 4, address space.

A second solution was to use the an updated version of the *bootstrap protocol* (BOOTP) which manually assigned IP addresses, called the *dynamic host configuration protocol* (DHCP). DHCP allows both manual and automatic IP address assignment and has largely replaced both BOOTP and the *reverse address resolution protocol* (RARP).

DHCP is based on a special server which assigns IP addresses to stations asking for one. In order to allow the server to be accessible from all stations, a relay agent is used at the edge of each LAN to forward DHCP requests. An IP address can be allocated for the time the station is connected or it can be leased for a fixed period of time.


Newly-booted host looking for its IP address
DHCP relay
Other networks
Router
DHCP server
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server

DHCP was the most successful solution as it required minimal change for the majority of users who were using IP. The problem with IP6, even though there were a whole host of extras included, was how to get users on the internet to adopt it?


START
Parse and Validate Network Header
Special Handling Indicated
Handle As Required
Invalid Field(s)
Error Exception Handler
All Fields Valid
Routing Table Lookup on Network Portion of Destination Address
Network is Directly Attached
Network is Remote
Map Station Portion of Destination Address to Local Data Link Address
Determine Next Hop Router from Table (default if unknown)
Unknown Address
Addressed To Router
else
Request mapping for Unknown Station
Sink Packet Internally
Is Fragmentation Required?
YES
NO
Fragment Datagram into Multiple Packets;
Build Network Headers as Needed For Each Fragment
Update Lifetime Control and Checksum Fields in Network Header;
Build Data Link header (SA, DA, etc.)
Queue Frame on Output Port
Routing Algorithm
Fast Path

d) As pictured left the fast path for unicast IP routing entails: **Packet parsing and validation** – The router need to separate certain fields to determine the type of handling required. Check the IP version number; Check the header length field (>20bytes means routing information present); Calculate the header checksum; Validating the source address.

Two non-fast path operations: **Fragmentation** – Each available output port will have an associated maximum transmission unit (MTU), which is the largest frame length permitted. It is generally a function of network technology or MAC used (Ethernet, Token ring, PPP etc). If the packet is larger than the MTU then it must be fragmented. Layer 2 protocols have no mechanism for fragmentation, however layer 3 protocols such as IP do. This is often a mixed blessing as the processing burden is high. Hence fragmentation is usually avoided or dealt with outside the fast path. Routers often initiate an MTU discovery process to find the most appropriate MTU sizes to use. Fragmentation is often avoided in certain network configurations such as campus wide networks by using a common LAN protocol such as Ethernet. If Ethernet is the dominant layer 2 protocol on the network, then a logical MTU of 1500 bytes can easily be established.

ICMP – the check on the presence of the options field often indicates the use of ICMP, which is not a fast path process as it has a large router process overhead.

e) UDP is a stripped down version of TCP which offers a very simple connectionless and unacknowledged service. TCP on the other hand can offer very complex services including acknowledged connection oriented service and even transmission via a form of virtual circuit to minimise delay and maximise quality of service (QoS). Remember that this on top of IP which is a connectionless protocol where packets are routed as they enter each network node.

Source port	Dest. port	Sequence number	Ack number	Control flags	Window field	Checksum	Urgent pointer	Data
2	2	4	4	2	2	2	2	

The basic structure of the TCP (or UDP) header contains several important data fields. The *source* and *destination ports* are used to identify well known application processes such as FTP or SMTP and can be used in the same way a logical connections or virtual circuits are identified in frame relay or X.25. Some protocols such as FTP use 2 ports, 21 is defined form control and 20 for data transfer. Ports can also be scanned for software vulnerabilities in higher layers and are common targets for hackers.

The *sequence field* is used to identify the order of data segments transported. The *acknowledgement field* depends on the setting of the ACK bit (see below). If it is set, then the field contains the sequence number of the data segment that the receiver will receive next. Otherwise it is ignored

The *control field* contains six flags, URG (urgent), SYN (syntax), ACK (acknowledge), RST (reset), PSH (push) and FIN (finish). The value URG = 1 indicates that urgent data is to follow (a form of priority). The SYN bit is used to indicate a connection request and is used to establish connections. The ACK flag is used to denote an acknowledgment is in progress. The PSH flag indicates that data should be pushed to higher layers and the RST is used to reset connections if needed. Finally FIN indicates the close of a connection.

The *window field* is used to set the number of bytes the sender can accept and functions as a flow control mechanism. Hence if a receiver becomes overloaded then it can reduce the window size for each new connection and reduce the data rate. This is often referred to 'controlling ones destiny'.

The control flags can be used in many ways to set up connection handshaking systems. A common one (until recently) was the three way handshake. A TCP connection starts with the setting of the SYN bit. The receiver returns with a SYN and ACK bit set to acknowledge the connection long with the setting of the window field. Finally the transmitter then completes the three way handshake by setting the SYN bit again. This handshake was used recently by hackers to form a *denial of service* attack by flooding a server with TCP SYN requests and then not responding to the SYN ACK returned. This causes the server to open multiple connections and eventually over loads the server and causes it to crash. The solution was to add further timers to the system to force a time out for no response to an opened port.

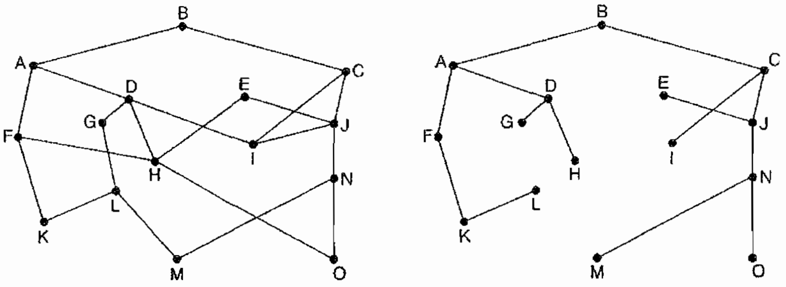
Q3 a) The routing algorithm is the basic process which sets up either an address table or a VC circuit list which allows a router to decide which line to send a packet upon. For either system of connection this is based on finding the best possible route through a given subnet. This can be multiple hops across different network routers with different bandwidths and delays, hence one of the most important decisions in setting up an algorithm is the choice of optimisation criteria.

Regardless of the connection type, there are certain properties which are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness and optimality. Correctness and simplicity are self explanatory, however robustness is less obvious, unless you consider the lifetime, evolution, topology change and reliability of a subnet over many years. Stability means that a network converges to an optimal solution, some algorithms do not do this at all. Fairness and optimality are not entirely mutual as there are often cases where he optimal traffic conditions mean that certain links will not function at all.

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. 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.

The choice of optimisation criteria is also very difficult. Packet delay is one possible criteria, as is maximising total network throughput, however maximum throughput comes when packets are extensively queued, which introduces excessive delays. Minimising the number of hops between end routers is a popular criteria as it tends to reduce delay and increase throughput at the same time.

b) 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. This is referred to as the *sink tree* as demonstrated below. Note that this tree is not necessarily unique for a given destination, however the goal of a routing algorithm is to find the sink trees for all routers on a subnet.



(a) (b)

Since the sink tree is a tree structure, then there will be no loops and each packet arrives in a finite number of hops. However, within a subnet routers may fail and crash causing links to go down and then reappear so different routers may have different ideas about the topology.

c) This algorithm is widely used as it is simple to implement and understand. The idea is to build a graph of the subnet where each node represents a router and then select the shortest possible path through the subnet based on the chosen link criteria. The choice of route will depend on the chosen criteria. If number of hops is chosen, then it will probably give a different series of routes than if geographical distance were chosen. There are many other criteria such as queue delay, transit times etc. The graph must be updated every time period to keep track of any changes in the subnet's performance or topology. In fact, it is most likely that the weighting factors given to each link on the graph will be a function of many different metrics.

There are several algorithms for choosing the shortest path in a connected graph, but one of the commonest was invented by Dijkstra in 1959. Each node is labelled in brackets with the distance from the source node along the best known path. Initially no paths are known so all nodes are labelled with infinity. As nodes and paths are found then the labels are updated with the current best known path to that point. Initially labels are tentative (open circle) but once it is known that the node is on the shortest path from the source it becomes permanent (full circle).

d) 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.

In the Fug 2 example the distance is in terms of delay. Once every T msec each router sends to its neighbours its delay estimates to each destination and at the same time receives the delay estimates from its neighbours. Suppose the router J wants to find the best route to router G. It knows the delays to its neighbours A,I,H and K as 8,10,12 and 6msec respectively. It also knows the estimated delays from those four routers to all other destinations. Hence it can calculate the delay to G via those 4 nodes as (18+8)=26, (31+10)=41, (12+6)=18 and (31+6)=37msec respectively. Hence it chooses the router H to send packets to G via as it has the minimum delay (18msec) estimate. A similar calculation is repeated for all destinations.

e) 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. The link state discovery process can be used to discover the structure of the MPLS network and then optimal routes can be found accordingly. These then translate to different labels. Also when a packet arrives, the destination address must be mapped on tot a suitable label and the link state routing can be used to build this table. 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.