4B15 2006 CRIBS

4B15 Dr T Wilkinson tdw@eng.cam.ac.uk

Q 1.

a) Explain the main concepts behind layer 2 interconnection of local area networks (LANs). What is meant by hard and soft invariants? How can these invariants be both a useful feature as well as a hindrance to the process of interconnecting LANs.

b) Sketch a diagram showing the way in which a bridge connects between the layers in the open systems interconnect (OSI) model. Give two reasons why it is not desirable to use the logical link control (LLC) functions in this bridging process.

c) Describe how the sub network access protocol (SNAP) can be used to expand the number of higher layer clients available in the local link control (LLC) frame format. Show how SNAP can be used to connect a type encapsulated Ethernet LAN to a SNAP based token ring LAN. What other consideration have top be made when making such an interconnection.

d) Give two reasons why interconnection at layer 2 is a limiting factor in creating a global internetwork of LANs. Explain how this limitation can be overcome. What is the cost in terms of performance in implementing such a change to the interconnection strategy?

Q1 crib a) An interconnection at the OSI layer 2 is done through a piece of hardware known as a bridge. 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*. 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

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. It also looks up the source address to see if it is contained within the table. If it is not, then it adds a new line for that address. If the looked up destination address 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. 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

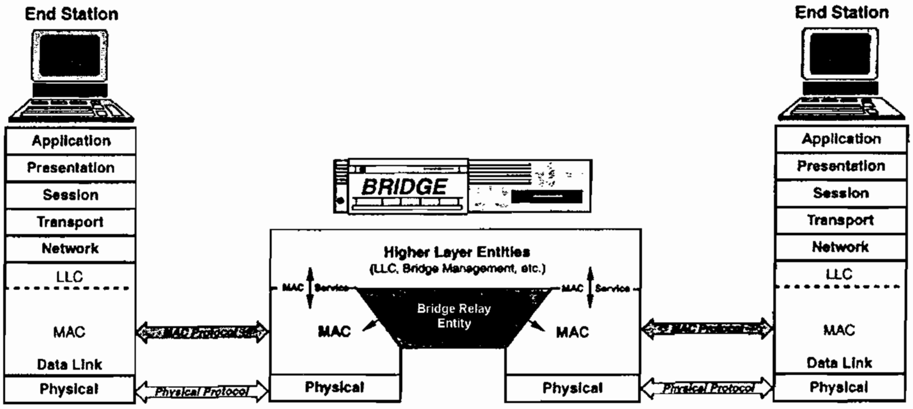
✓ Low error rate

✓ 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. Bridges are very efficient due to the hardware operation, but they are limited in scalability and complexity by the invariants.

b) 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.



If you go out and buy a bridge from a modern manufacturer you are most likely to get a ‘real bridge’ which will have enhanced LLC and network layer functions in included with the basic ‘architectural bridge’ functionality. This will include network monitoring and management functions as well as higher layer routing or the spanning tree protocol. Two drawbacks for using an LLC based bridge are i) There is an added delay and complexity overhead added by the LLC management processes which can cause extra latency and also compatibility between different LLC entities. ii) There is a risk of undetected frame corruption. If a bit error occurs while the frame in a non-LLC based bridge, then it will be sent with the original FCS and the error detected at the receiver. If there is LLC an a bit error occurs in the data, then a new FCS will be generated this will incorporate the error and will be missed by the receiver.

c) There are a limited number of SAPs that can be addressed with the single LLC byte, restricting the multiplex to a maximum of 256 clients, half of which are reserved for multicasts. This is even further reduced, as the second bit is often reserved in the SAP as well. The multiplex can be expanded with a further header sub-division known as sub-network access protocol (SNAP) encapsulation.

LLC Header SNAP Header

MAC Header	DSAP 0xAA	SSAP 0xAA	Ctrl 0x03	**SNAP OUI**	SNAP Pid	Data
Bytes	1	1	1	3	2	

Here the DSAP and SSAP are filled with the value 0xAA followed by the organisationally-unique identifier (OUI) which indicates the organisation for which the protocol identifier (Pid) is significant. This allows any organisation to set up to 65536 private higher layer identifiers. See the MAC section for the OUI and Pid.

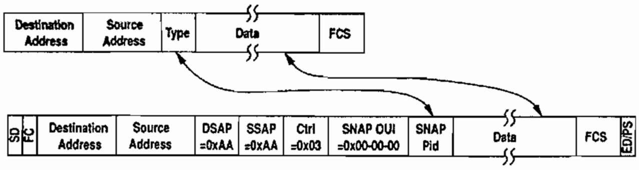
In order to make a MAC translation possible there are a few mechanisms which must be adhered to in order to prevent chaos from breaking out.

✓ 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.

The other main consideration when connecting between different MACs is the maximum transmission unit (MTU) which should be adopted for each data frame.



Notes: Arrows indicate that the literal contents are copied to the appropriate field(s).
Fields without arrows are created as needed for the transformation.
The Frame Check Sequence must be recalculated after the transformation.

d) Layer 2 interconnection is limited by the invariants as well as the size of the address look up table for all stations. A better mechanism is to go to layer 3 where there is more facilities for managing network structures on a global scale through structures addressing and address tables in routers. The main limitation of this is the added complexity and processing overhead at layer 3. Any complex interaction should be a boundary event and dealt with n a low priority queue. This is the concept behind the fast path.

Q2.

a) Explain what is meant by the terms *wirespeed* and *full duplex* in the context of an Ethernet based local area network (LAN). How have these two concepts radically changed the way in which LAN protocols operate at layer 2? Use a sketch to show how this development has changed the topology of Ethernet LANs.

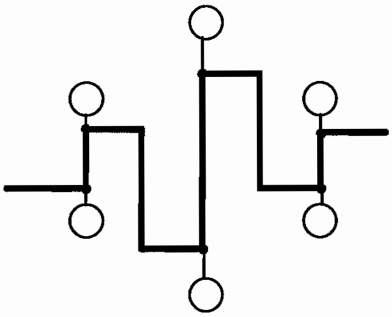
b) Sketch a diagram showing how the two concepts in part (a) can be used to perform link aggregation. Why is this a useful feature when considering interconnection in a LAN? Suggest two ways in which link aggregation can be implemented without breaking the layer 2 hard invariants. How are medium access control (MAC) addresses managed during the process of link aggregation?

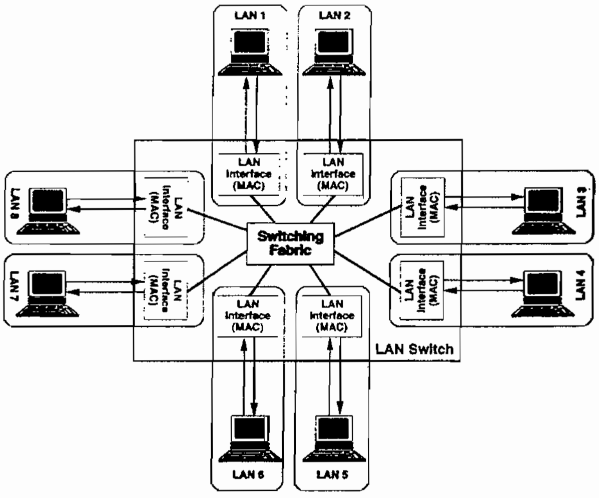
c) Explain how a simple form of flow control can be implemented at layer 2 by using the medium access control (MAC) protocol to compensate for congestion. How does this technique compare with flow control implemented in the higher layer transport control protocol (TCP)?

d) Link aggregation is normally implemented at the layer 2 level. How could it be implemented at layer 3? Is this really a useful mechanism for enhancing the throughput of a link or does the added complexity of the layer 3 processing negate any serious benefit from the aggregation?

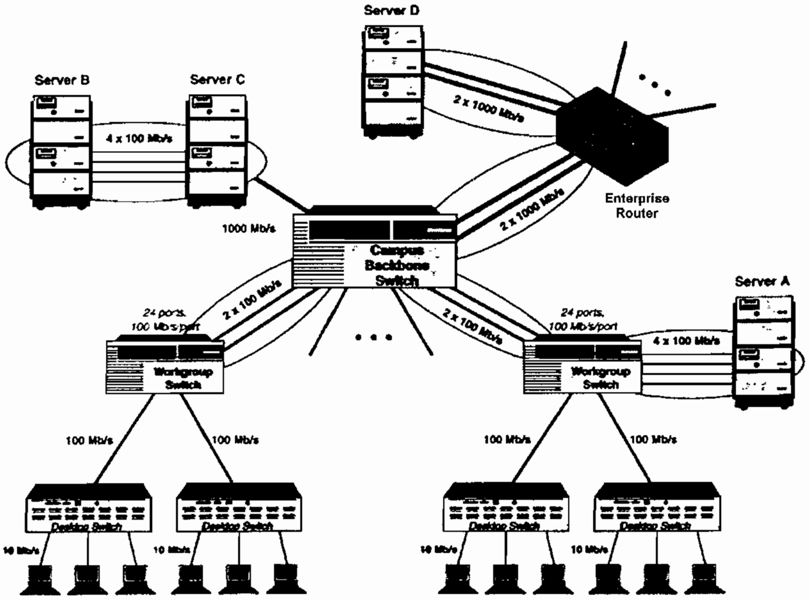
Q2 crib a) The term *wirespeed* refers to when a bridge or switch is capable of operating all of its ports at the fully specified data rates. One of the main advantages of LAN MAN and even WAN connections is the ability to operate at high data rates. This is especially the case with LANs, however their operation is often hampered by shared media or *half duplex* operation. A more desirable means of operation would be to operate in *full duplex*, where each direction of traffic to or from a station occupies its own physical channel. This has the effect of negating MAC protocols such as CSMA/CD and token rings. There are two main factors which allow full duplex operation.

- The use of dedicated media such as structured cabling
- The use of microsegmentation about a switch.





b) Wirespeed operation and full duplex means that multiple physical links can be combined or aggregated to increase the nett bandwidth between high traffic systems such a routers and servers.



The key question in an aggregate link is how to assign the data to each link in the aggregate. A *striping* technique could be used where a frame is split between links and sent in parallel, however this is not possible with LANs. Hence whole frames are sent along each link and the problem is how to manage this process without breaking the LAN hard invariants. This is particularly difficult when trying to maintain frame sequence. The secret to maintaining frame sequence is to find why frame order must be maintained and which frame need to be kept in order. We can then relax the strict invariant requirement of LAN transmission. Not all traffic across the same LAN link will be from the same stations or applications so not all order is essential. In an aggregate link, a *conversation* (sometimes called flows) is defined in traffic when order must be maintained. Hence the distributors job means that frames from the same conversations must be sent down the same link.

So how does an NIC or switch interface decide on individual conversations? This will depend on the applications used to send the data across the LAN. An example could be in a switch to switch link aggregate where destination MAC addresses make a very good means of determining conversations. This technique does not work well in switch to server connections as all frames will carry the same MAC destination address. A better system might be to use MAC source addresses.

In a traditional non aggregated link, each network interface controller (NIC) has a globally unique 48bit MAC address. This is used as the source and destination address for the station. When an aggregated link is set up, the link should appear to higher levels to have a single MAC address, however this is not case in hardware as each NIC has its own address. Hence the software driver which is controlling the aggregated NICs must take a single address and assign it to an aggregated group of links. This can be done by overwriting the MAC address register in software.

c) The basic mechanism that can be used to control congestion is to send some sort of signal to the sending station to reduce its rate of transmission or throttle back its frames. There are mechanisms for both half and full duplex links which can avoid higher layer interactions in order to reduce congestion. For half duplex links, the access control mechanism can be used to force a sending station to reduce its output in order to conserve buffer overflow in the switch.

- **Backpressure** – A switch uses the access protocol to slow the data arriving at its input ports.
- **Aggressive transmission** – A switch tries to remove data quickly at its exit ports by shortening the transmission procedure.

Many higher level features can be specified for the switch to operate on such as network management, congestion control and delay sensitive traffic priority such as video streaming. The control of layer 4 operation using protocols such as TCP is often done on a stream of packets referred to as an *application flow*. 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.

d) The case for link aggregation at layer 3 is not so clear cut as the majority of the processes on this layer are implemented in software and do not always lead to efficient link operation ate wirespeed rates. There is normally a tradeoff for using layer 3 functions between data rate and flexibility and therefore low latency. Although layer 3 aggregation works in principle as there are no invariant limitations and there is the possibility to sequence, fragment and offset packets. Most of these features are in fact non-fastpath based processes which negates the advantage of link aggregation. Is there any real point in running two links together at half the data rate due to the complexity of the processing overheads.

Q3.

a) Explain what is meant by shortest path routing in an internetwork. How does Dijkstra's algorithm select an optimum path through such an internetwork. Use the internetwork shown in Figure 1 to demonstrate Dijkstra's algorithm.

b) What are the limitations of using shortest path routing? How do distance vector routing algorithms help mimimise these limitations? Give an example of how a vector routing algorithm operates.

c) Define the 5 main stages in implementing a link state based routing algorithm. Give two examples of possible link state metrics. Why is this choice important? Use a simple example to demonstrate how link state routing can be simplified by using a hierarchical structure.

Q3 crib a) **Shortest path routing.** 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). In the example on the next page we start at permanent node A. We then find the paths to B and G with the appropriate distances. We then make B permanent as I is the shortest from A. We then move to B and repeat the process, each time locating the shortest possible distance. After all the tentative nodes have been labelled, then the graph is searched to find the shortest path from any permanent node to the next tentative node. This is then made permanent.

1. Discover its neighbours and learn their network addresses
2. Measure the cost or delay to each neighbour
3. Construct a packet telling all learned in 1 and 2
4. Send the packet to all other routers
5. 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.

1. When a router is first booted it sends a special HELLO packet to all its neighbours to find out who they are. The addresses used must be globally unique.

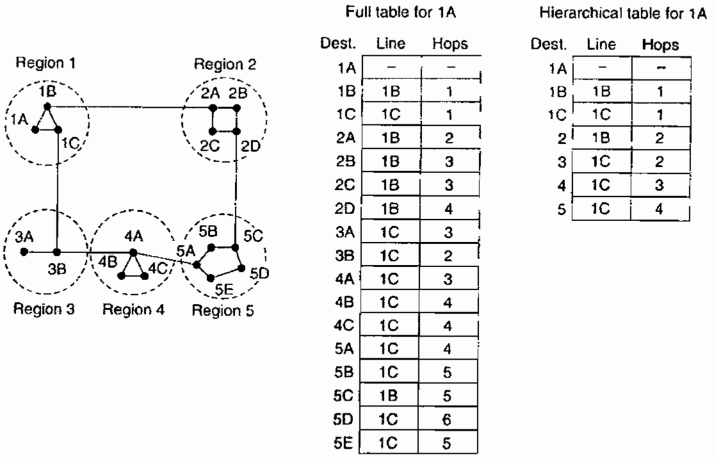
2. Measure the line costs to each neighbour or at least have a reasonable estimate. This is normally done through ECHO packets which must be returned as quickly as possible to measure the time stamp. Other metrics include number of hops (good for long distances), data rate, throughput, delay and error reliability.

3. Once the information has been gathered, then the link state packets are formed including a sequence number and an age field (see later).

4. The most difficult part is to distribute the link state packets reliably. As packets are received, the router will update its table, which could lead to different routers having different versions of the packets and tables. The fundamental idea is to use flooding to distribute the packets. To keep this in check, each packet has a sequence number which is incremented for each new packet sent. To avoid these problems the age field is also included and packets are aged with time until they reach zero when all information about that router is discarded.

5. Once the router has a complete set of link state packets, it can construct a map of the subnet and choose the optimal routes using Dijkstra's algorithm.

When a subnet increases in size, it puts more demand on the memory and processing overheads for the routing algorithm. This can be exceeded in very large subnets, however there is a simple solution to break up the subnet into regions and then route each region as a separate subnet. This is often done on a metropolitan level, with regions based within a city then interconnected at a higher level between cities.

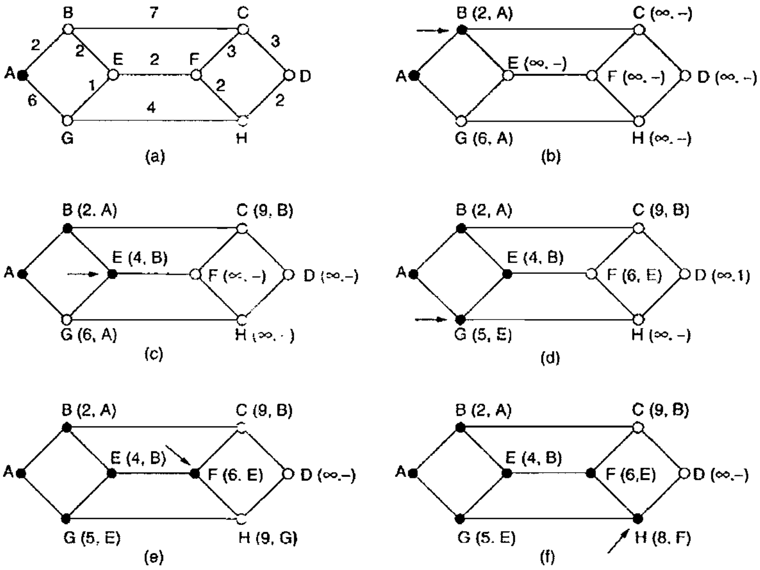


Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

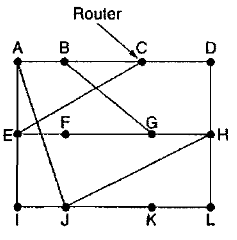
Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4



b) **Distance vector routing.** 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.



(a)

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

In the above 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.

c) **Link state routing.** 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: