9.17 State the meaning of the following terms relating to the routing of packets over the Internet:
(i) line cost,
(ii) path cost,
(iii) hopcount,
(iv) routing metric,
(v) shortest path.

9.18 With the aid of the routing table entries shown in Figure 9.11, explain the meaning of the terms:
(i) static routing tables,
(ii) next-hop routing,
(iii) optionality principle,
(iv) alternative paths.

9.19 With the aid of the broadcast diagram shown in Figure 9.12, explain:
(i) why the broadcast following the route via R2 is assumed to arrive first
(ii) how duplicate copies of a packet are determined by R3
(iii) how the number of copies of the packet produced is limited
(iv) why flooding is an example of an adaptive/dynamic routing algorithm.

9.20 In relation to the distance vector algorithm, with the aid of the example shown in Figure 9.13, explain:
(i) the meaning of the term "connictivity/adjacency table" and how the table's contents are obtained
(ii) how the final routing table entries for R3 are built up
(iii) how a packet from a host attached to netid3 is routed to a host attached to netid1
(iv) the limitations of the algorithm including how looping may arise.

9.21 Assuming the connectivity/adjacency tables given in Figure 9.14(a), show how the overall network topology is built up by router R3 using the link state algorithm. Hence derive the contents of netid location table for R3.

9.22 Assuming the initial network topology shown in Figure 9.15(a), use the Dijkstra algorithm to derive the shortest paths from R3 to each other router. State the meaning of the terms "tentative" and "permanent" relating to the algorithm and the implications of alternative paths/routers.

9.23 Using the set of link state, routing, and connectivity tables for R1, R2, and R3, explain how a packet received by R3 from G3 with a destination netid of 1 is routed using hop-by-hop routing.

9.24 Explain how a packet received by R3 from G3 with a destination netid of 1 is routed using source routing. Include how the routing tables you use are derived.

9.25 In relation to the link-state algorithm, explain why each link-state message contains a sequence number and a timeout value. How are these used?

9.26 With the aid of the generalized Internet architecture shown in Figure 9.18, state a suitable routing algorithm for use
(i) within a single bridged LAN,
(ii) within a single site comprising multiple different LANs interconnected by subnet routers,
(iii) within a single autonomous system such as a regional/national network,
(iv) within the core network of the Internet.

9.27 In relation to the BGP, with the aid of the message types shown in Table 9.1, explain how the following functions are performed:
(i) neighbor acquisition and termination,
(ii) neighbor reachability,
(iii) routing update.

9.28 State the reason why classless inter-domain routing (CIDR) was introduced.

9.29 Derive the range of class C addresses that are available. Hence, using the list of continental addresses given in section 9.6.6, explain how the routing of packets across the global backbone network is carried out.

9.30 State why the lack of a fixed division point within class C addresses with CIDR means a router/gateway is unable to route a packet. Hence explain how this is overcome.

9.31 An example of how when using CIDR a host attached to one network can produce a match with the mask of a second network was illustrated in Example 9.5. How would this be overcome in practice?

9.32 Explain the term "tunneling" and when it is used. Hence with the aid of the schematic diagram shown in Figure 9.20, explain how the host on the lift of the diagram sends an IP datagram/packet to a host attached to the Internet. Include in your explanation the role of the two multiprotocol routers.

State an application of tunneling IP packets over an IP network.

9.33 What are the aims of both the reverse path forwarding algorithm and the spanning tree broadcast algorithm?

9.34 Use the final routing tables and broadcast sequence relating to the reverse path forwarding algorithm shown in Figure 9.21 to explain why only the (broadcast) packet received by SR6 from SR3 is broadcast at the fourth stage. What is the number of duplicate broadcasts that occur?

9.35 Assuming the network topology shown in Figure 9.22(a) and that SR3 is access gateway, determine the spanning tree derived by each subnet router. Use this to derive the broadcast sequence.

9.36 In relation to multicasting over a LAN, describe how IANA controls the allocation of multicast addresses. Also explain how the 48-bit MAC address and 28-bit IP address of a host are derived from the allocated address. Hence with the aid of the schematic diagram shown in Figure 9.23(b), describe how a host joins a multicast session that is taking place over the LAN. Include the role of the multicast address table and group address table held by each member of the group.

9.37 What is the meaning of the term "multicast router"? Outline the sequence of steps that are followed to route an IP packet with a multicast address over the Internet.

9.38 Assume the same topology, multicast address table contents, routing table contents, and routing table entries as shown in Figure 9.24. Assuming the DVMRP, explain how a packet arriving from one of its local networks with a multicast address of C is routed by MR3 to all the other MRs that have an interest in this packet.

9.39 Repeat Exercise 9.37 but this time using the MOSPF routing protocol and the spanning tree shown in Figure 9.25(b).

9.40 What is the role of the IGMP protocol?

With the aid of the example shown in Figure 9.26, explain how a host that is attached to a local network/subnet of an MR joins a multicast session. Include in your explanation the table entries retained by both the host and the MR and how multicast packets relating to the session are then routed to the host.

9.41 With the aid of the example shown in Figure 9.26, explain the procedure followed when a host that is attached to a local network/subnet of an MR leaves a multicast session.

9.42 The multicast backbone (m-bone) network shown in Figures 9.24 and 9.25 comprised a set of multicast routers interconnected by single links. In practice, these are logical links since each may comprise multiple interconnected routers that do not take part in multicast routing. Explain how a multicast packet is sent from one mrouter to another using IP tunneling.

Section 9.7

9.43 Explain briefly the role of the ICMP protocol and the different procedures associated with it. Hence explain how the path MTU discovery procedure is used to determine the MTU of a path/route prior to sending any datagrams.

Section 9.8

9.44 Discuss the reasons why improved levels of QoS support are now being used within the Internet.

9.45 Describe the role and principle of operation of the following control mechanisms used within Internet routers:
(i) token bucket filter,
(ii) weighted fair queuing,
(iii) random early detection.

9.46 Define the three different classes of service used with the Intserv scheme. With the aid of the network topology shown in Figure 9.28 (a) describe the operation of the resource reservation protocol (RSVP). Include in your description the meaning/role of the following:
(i) path, reserve, and path-tear messages
(ii) path-state table,
(iii) cleanup timer,
(iv) soft-state.

9.47 Define the usage of the type of service (ToS) field in each packet header with the DiffServ scheme including the meaning of the term "DS packet codepoint".

9.48 With the aid of the general architecture shown in Figure 9.28(b), describe the operation of the DiffServ scheme. Include in your description the meaning/role of the following components of an ingress router;
(i) behavior aggregate,
(ii) traffic meter module,
(iii) MF classifier,
(iv) marker module,
(v) shaper/dropper.

Also explain the meaning/role of the following components of a core router:
(i) BA classifier,
(ii) per-hop behavior,
(iii) expedited forwarding,
(iv) assured forwarding.

Section 9.9

9.49 Identify a selection of the applications of the point-to-point protocol (PPP) within the Internet. With the aid of the PPP frame format shown in Figure 9.29, explain the meaning/use of the following fields:
(i) opening and closing flags with zero bit insertion,
(ii) the byte stuffing rules used with asynchronous transmission lines.

9.50 State the role of the protocol ID field in the header of a PPP frame. Hence describe in outline the features supported by the link control protocol (LCP) associated with PPP.

Section 9.10

9.51 Discuss the reasons behind the definition of IP version 6, IPv6/IPng, including the main new features associated with it.

9.52 With the aid of the frame format shown in Figure 9.30(a), explain the role of the following fields in the IPv6 packet header:
(i) traffic class,
(ii) flow label,
(iii) payload length (and how this differs from the total length in an IPv4 packet header),
(iv) next header,
(v) hop limit,
(vi) source and destination addresses.

9.53 In relation to to IPv6 addresses, with the aid of the prefix formats shown in Figures 9.31(a) and (b), explain the meaning/use of:
(i) address aggregation,

(ii) prefix formats,
(iii) embedded IPv4 addresses.

9.54 With the aid of the frame format shown in Figure 9.31(c), explain the meaning/use of the following IPv6 fields:
(i) registry,
(ii) top-level aggregators,
(iii) next-level agrregators
(iv) site-level aggregators
(v) interface ID.

Comment on the implications of adopting a hierarchical address structure.

9.55 With the aid of examples, explain the use of a link local-use address and a site local-use address.

9.56 Explain the format and use of
(i) a multicast address,
(ii) an anycast address.

9.57 With the aid of examples, show how an IPv6 address can be represented:
(i) in hexadecimal form,
(ii) with leading zeros removed,
(iii) when it contains an IPv4 embedded address.

9.58 Explain the role of the extension headers that may be present in an IPv6 packet. List the six types of extension header and state their use. Also, with the aid of examples, state the position and order of the extension headers in relation to the main header.

9.59 The fields in an options extension header are encoded using a type-length-value format. Use the hop-by-hop options header as an example to explain this format.

9.60 In relation to the routing extension header, explain
(i) the difference between strict and loose source routing, and the associated bit map,
(ii) the use of the segments left field.

9.61 In relation to the packet formats shown in Figure 9.33, explain:
(i) the meaning and use of the identification field and the M-bits in each extension header,
(ii) why the hop-by-hop and routing headers are present in each fragment packet.

9.62 In relation to the encapsulating security payload header, with the aid of diagrams, explain:
(i) the difference between transport mode and tunnel mode encryption,
(ii) the meaning and use of the term "steel pipe".

9.63 State the aim of the autoconfiguration procedure used with IPv6 and the application domain of
(i) the neighbor discovery (ND) protocol and
(ii) the dynamic host configuration protocol (DHCP).

9.64 With the aid of Figure 9.35, explain the operation of the ND protocol. Include the role of the router solicitation and router advertisement messages and how a host creates its own IP address.

9.65 Explain how an IPv6 address is obtained
(i) using a DHCP address server,
(ii) a DHCP relay agent.

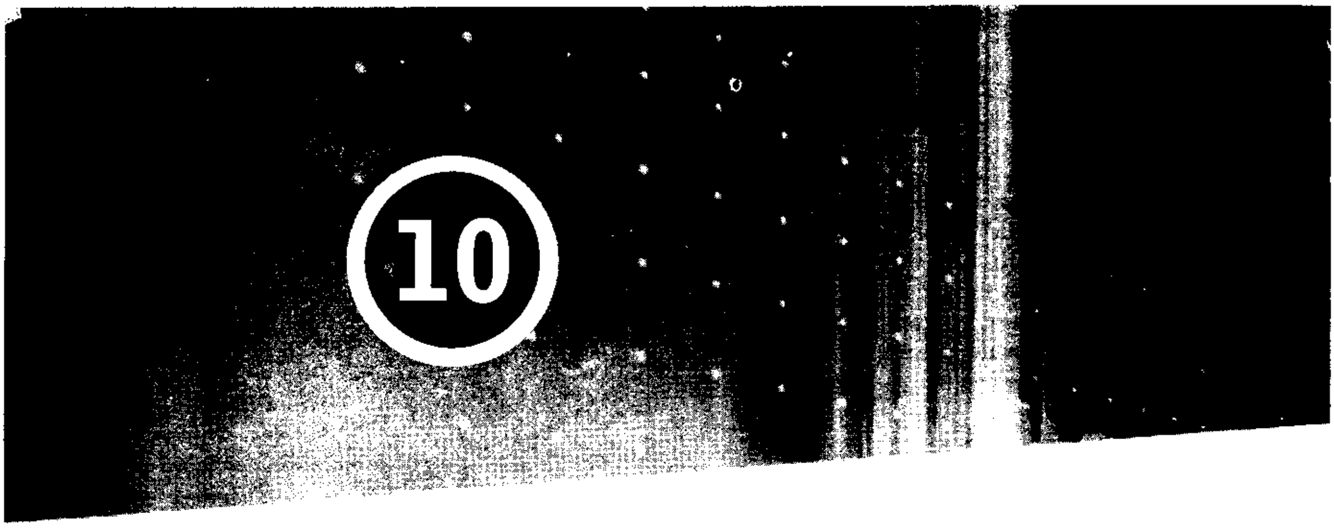
Section 9.11

9.66 With the aid of Figure 9.36, explain how a LAN server can respond to requests from both an IPv4 and an IPv6 client using dual protocols.

9.67 With the aid of Figure 9.37, explain how two hosts, each of which is attached to a different IPv6 network, communicate with each other if the two IPv6 networks are interconnected using an IPv4 network. Include the addresses that are used in each message transfer.

9.68 State the meaning of the terms "network address translation" (NAT) and "protocol translation" (PT). Hence, with the aid of the schematic diagram shown in Figure 9.38(a), explain the role and operation of a NAT-PT gateway. Include what the source address in each IPv6 packet should be.

9.69 Identify when the use of a NAT-PT gateway is not practical. Hence, with the aid of schematic diagram shown in Figure 9.38(b), explain the role of an application level gateway.



Broadband ATM networks

10.1 Introduction

As we explained in Section 1.3.5, broadband ISDN (B-ISDN) was designed from the outset to support multimedia communication applications involving text, high-resolution images, speech, audio, and video, either singly or a number integrated together in some way. At the time of the design (and standardization) of B-ISDN, the prevailing compression algorithms associated with the different media types were such that the anticipated peak channel bandwidth per call would be well in excess of the 2 Mbps offered by a (narrowband) ISDN.

Since each call may involve different types of media, the transmission and switching system chosen for the network is independent of both the bit rate and whether the source information is in the form of blocks or a variable rate bitstream. The chosen method involves the source information associated with each call being first converted into small fixed-sized packets known as **cells**. And since the rate of generation of the source information varies, so the rate of entry of cells into (and through) the network may also vary. Hence instead of allocating a fixed portion of transmission bandwidth per call, the cell streams relating to different calls are multiplexed together on a statistical basis. The switching units then operate using a form of packet switching called **cell switching**.

The adoption of a small fixed-sized cell means that cell switches can operate at a much higher rate than variable-length packet switches, so cell switching is also known as **fast packet switching**. Also, because the cells relating to different calls have varying time intervals between them during both transmission and switching, this mode of operation is known as the **asynchronous transfer mode (ATM)** and networks that operate in this way (broadband) ATM networks.

Having adopted a small cell size and a packet-switching mode of operation, the next decision to be made was whether the network should operate in a connection-oriented (CO) or a connectionless (CL) mode. As we showed in Figure 1.22 and explained in the accompanying text, in a CL mode network, each packet requires the full networkwide address of both the source and destination end systems in its header. The adoption of a small cell size, however, meant that the header may then be disproportionately large compared with the actual cell contents. So in order to keep the header small, the CO mode of operation was chosen. In a packet-switched network, this involves a path/route through the network being established prior to the transfer of any information packets and, on completion of the call, the path being closed down. The resulting path is called a virtual circuit and, in order for each packet-switching exchange to relate incoming packets to a specific virtual circuit, a unique identifier is assigned to the call on each line making up the virtual circuit as this is set up. The identifier used on each line is called the virtual circuit identifier (VCI) and, by assigning a new VCI on each line, only sufficient identifiers are required to identify the packets relating to the different calls on each line rather than the different calls in the total network. This same mode of operation is used in ATM networks.

The final decision was the size of the payload of a cell. Small cells have advantages for constant bit rate traffic since only a short delay is experienced as successive bytes relating to the same call are assembled and disassembled into and from cells at the network interface. Conversely, since each cell must contain additional routing information, small cells have the disadvantage that the overheads associated with each cell (in terms of transmission and switching) are higher. A compromise was reached by the various international standards bodies and a cell size of 53 bytes/octets was chosen. This comprises a 48-byte payload (information) field with a 5-byte header for the VCI and other fields. No error control is performed on cells within the network and hence no sequence numbers are required for retransmission purposes. However, the header does contain error check bits to detect the presence of transmission errors in the various header fields.

Because of the predicted high bit rates per call, an optical fiber access network was to be used to connect subscriber premises to the core network. As we indicated earlier, however, the rapid advances that have taken place in compression mean that the bit rate required for each call is significantly less than originally thought. As a result, existing access networks such as those associated with an ISDN and a PSTN can now support a range of multimedia communication applications. Hence the introduction of a B-ISDN access network has been postponed by most telecom providers. Nevertheless, the asynchronous transfer mode of working has been adopted for a number of other networks in which high bandwidth is required. As we showed in Figure 1.5, examples include ATM LANs and MANs, the latter providing a switched high-speed LAN interconnection facility. Also, as the bit rate of the transmission lines used in the Internet backbone network has increased – owing to the increasing number of multimedia applications – the cell-switching mode has been incorporated into the design of some of the high-speed routers used in the core backbone. In this chapter, we present further details of the operation of cell-switching networks in Sections 10.2 to 10.4 and the operation of ATM LANs in Section 10.5. We then describe the operation of ATM MANs in Section 10.6 and show how they are being used as the basis of broadband wide area networks in Section 10.7.

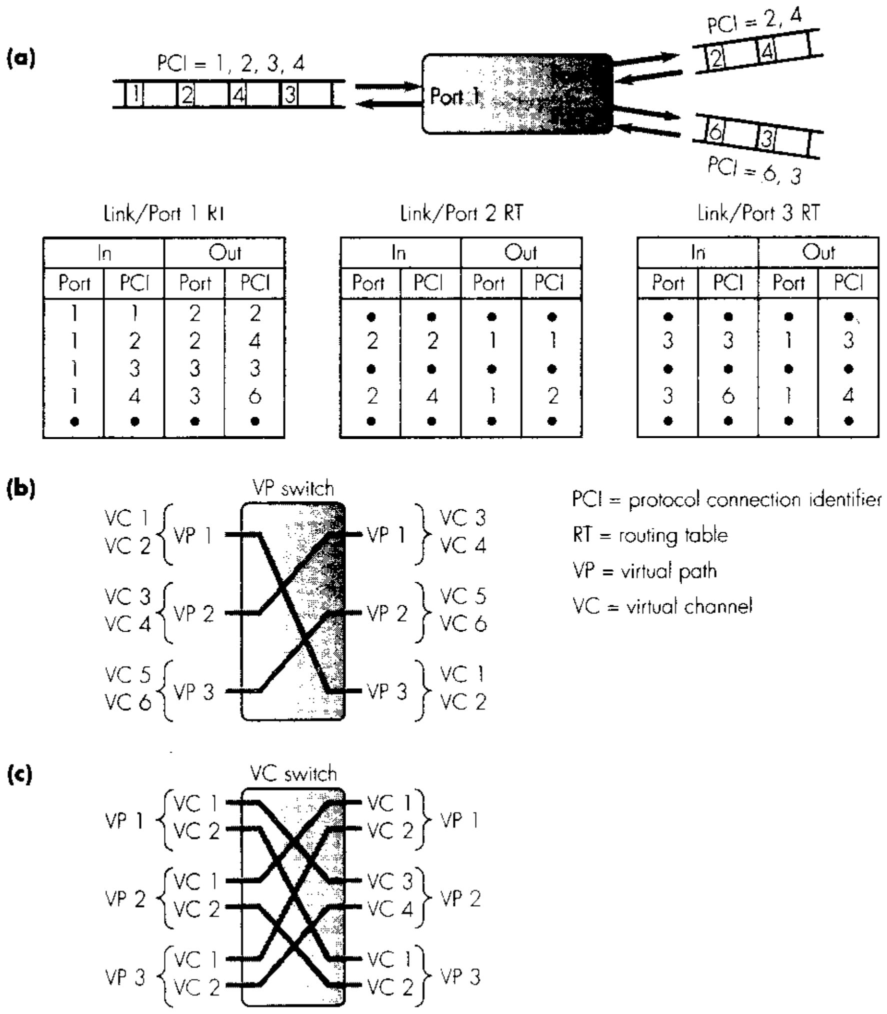
10.2 Cell format and switching principles

As we have explained, prior to any information cells being sent, a virtual circuit is first established. In an ATM network, the virtual circuit identifier used on each link is known as the **protocol connection identifier (PCI)**. The principle of the routing scheme used is shown in Figure 10.1(a).

Associated with each incoming link/port is a routing table that contains, for each incoming PCI, the corresponding outgoing link/port and the new PCI to be used. The routing of cells in both directions along a route is thus very fast as it involves a simple look-up operation. As a result, cells from each link can be switched independently and at very high rates. This allows parallel switch architectures to be used and high-speed transmission lines in the gigabit range, each operating at its maximum rate.

In practice, the PCI is made up of two subfields: a **virtual path identifier (VPI)** and a **virtual channel identifier (VCI)**. Routing can be performed using either one or a combination of the two. Two examples are shown in Figure 10.1. In part (b), switching is performed on virtual paths and the VCIs within each virtual path remain unchanged. In part (c) switching is performed on the virtual channels within each virtual path independently and the virtual paths simply terminate at each switch port.

An example of the use of virtual path switching is when multiple calls originating at the same network entry point are all intended for the same destination exit point. Each individual call is assigned a separate VCI and the calls are multiplexed together at the source interface onto a single virtual path. The multiplexed set of calls are then switched using the VPI field only and hence all follow the same path through the network. In this way, the set



Link/Port 1 RT

In		Out	
Port	PCI	Port	PCI
1	1	2	2
1	2	2	4
1	3	3	3
1	4	3	6
•	•	•	•

Link/Port 2 RT

In		Out	
Port	PCI	Port	PCI
•	•	•	•
2	2	1	1
•	•	•	•
2	4	1	2
•	•	•	•

Link/Port 3 RT

In		Out	
Port	PCI	Port	PCI
•	•	•	•
3	3	1	3
•	•	•	•
3	6	1	4
•	•	•	•

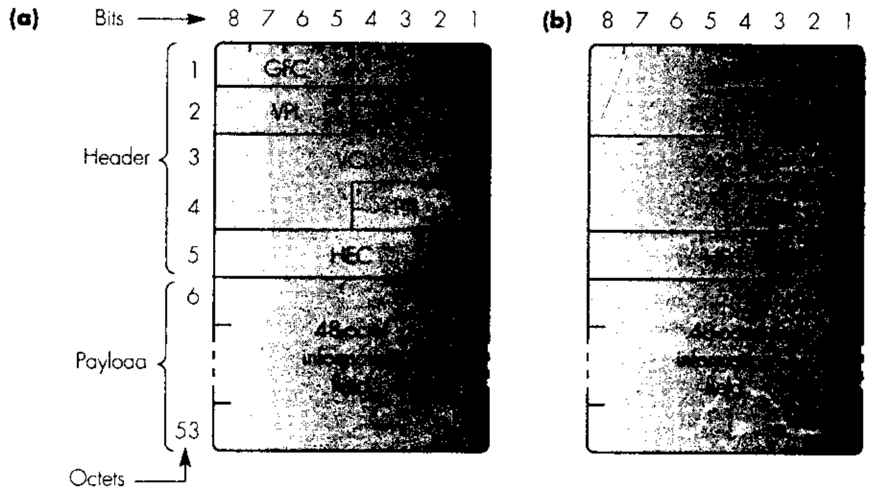
Figure 10.1 Cell switching principles: (a) routing schematic; (b) VP routing; (c) VC routing.

of VCIs remain unchanged and are used at the destination to identify (demultiplex) the individual calls.

An example of the use of virtual channel switching is when each call at the network entry point is intended for a different destination. In this case, each call is again assigned a different VCI but switching is also performed using the VCI field. The VPI field then has local significance to each link only and is used, for example, to allow calls to be multiplexed together for transmission purposes.

The format of each cell is shown in Figure 10.2 and, as we can see, the header is made up of six fields. Their functions are as follows:

- **generic flow control (GFC):** this is present only in cells transferred over the user–network interface (UNI) and is included to enable a local switch to regulate – flow control – the entry of cells by a user into the network. Within the network, cells transferred over the interexchange links – known as the network–network interface (NNI) – do not contain this field and the four bits are part of the VPI field;
- **virtual path identifier (VPI):** this is 8 bits at the UNI and, as just indicated, 12 bits at the NNI. As described previously, it is used for identification/routing purposes within the network;



PTI: 000 – user data, no congestion, SDU type 0
001 – user data, no congestion, SDU type 1
010 – user data, congestion, SDU type 0
011 – user data, congestion, SDU type 1
100 –
101 –
110 – Network control
111 –

GFC = generic flow control
VPI = virtual path identifier
VCI = virtual channel identifier
PTI = payload type identifier
CLP = cell loss priority
HEC = header error checksum

Figure 10.2 ATM cell formats: (a) user–network segment; (b) within network, network–network interface.

- **virtual channel identifier (VCI):** a 16-bit field used for identification/routing purposes within the network;
- **payload type indicator (PTI):** indicates the type of information carried in the cell; the different types are shown in Figure 10.2. All cells containing user data have a zero in the most significant bit. The next bit indicates whether the cell has experienced excessive delay/congestion or not, and the third bit the service data unit (SDU) type – 0 or 1. We shall discuss its use in Section 10.4.1 in relation to the AAL5 service. The four remaining cell types are used for network control purposes;
- **cell loss priority (CLP):** within the network, the statistical multiplexing of cells on each link may occasionally result in cells having to be discarded during heavy load conditions. This field has been included to enable the user to specify a preference as to which cells should be discarded; CLP = 0 high priority, CLP = 1 low priority and hence discard first;
- **header error checksum (HEC):** generated by the physical layer and is an 8-bit CRC on the first 4-bytes of the header.

10.3 Switch architectures

The general structure of an ATM switch is shown in Figure 10.3(a). Each input link is terminated by an **input controller (IC)** which performs the routing of cells arriving at each link (port) to their required output link. This involves a simple look-up and mapping operation of the VPI/VCI in the header of the incoming cell into the corresponding output VPI/VCI. Normally, the output port number obtained from the routing table is used to determine the path to be followed through the switching fabric to the required output controller.

Because there is no reservation of slots on the output links, cells may arrive simultaneously at two (or more) input ports that require the same output port/link. This is handled in one of two ways: either the input controllers contain a set of cell buffers that hold the additional cell(s) or the buffering is provided in the **output controllers**. In both cases the buffers are organized in the form of a FIFO queue to ensure the cells from each input controller are output in the same order as they arrived. The output controllers simply forward received cells at the appropriate link bit (and hence cell) rate.

The main role of the **control processor** is to download routing information into the routing tables in each input controller. Normally, the routing information is received through the network either from a network management station or, if switched VCs are being used, from a signaling control point processor. In both cases, semipermanent VCs are used to relay the cell containing the routing information. Hence on arrival at the switch, the related cells are routed through the switching fabric directly from the input controller that receives the cells to the control processor. In addition, the

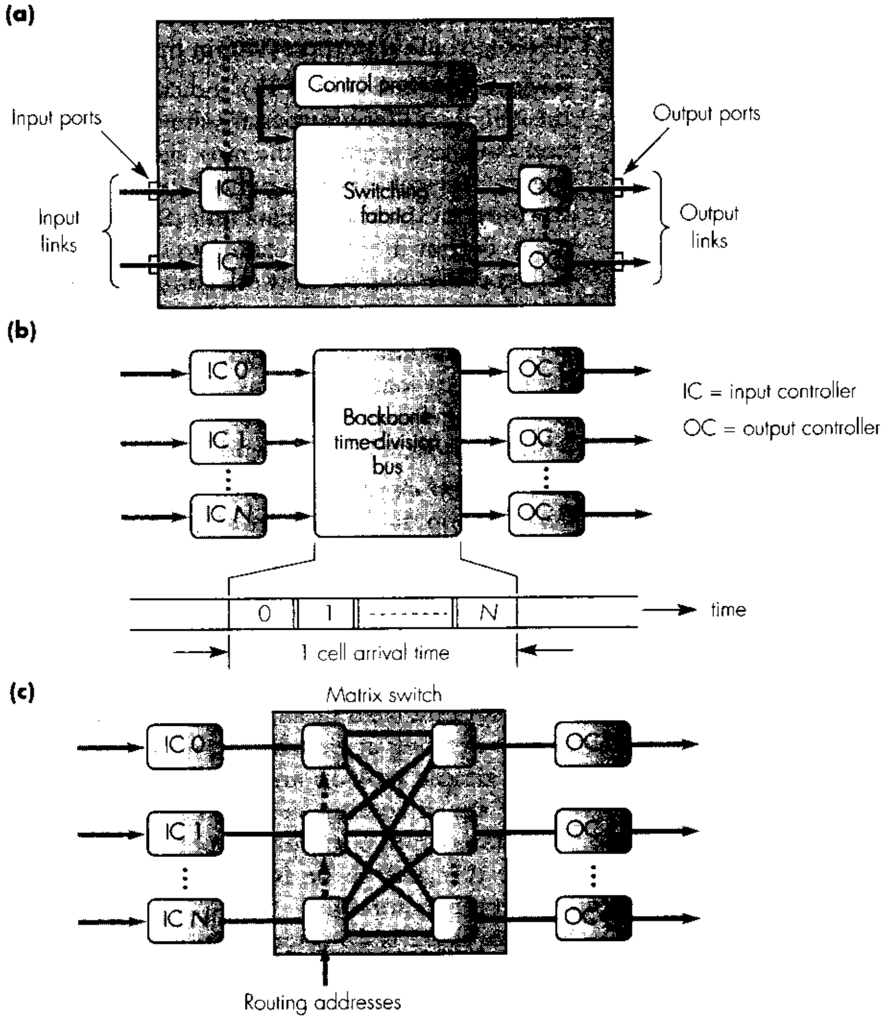


Figure 10.3 ATM switch architectures: (a) general structure; (b) time-division bus schematic; (c) fully-connected matrix switch.

control processor may generate network management messages itself – fault reports, performance statistics, and so on – and these are also routed through the switch fabric to the required output controller for onward transmission to the network management location again using a semipermanent VC.

A number of alternative switch fabrics are used in ATM switches. They can be classified as either time division or space division. Normally, all the input controllers are synchronized so that each set of incoming cells from all controllers is presented to the switch fabric in synchronism. The switch fabric also operates synchronously which means that the cells from each input controller are transferred to their required output controller in a single cell time.

A schematic diagram of a **time-division switch** is shown in Figure 10.3(b). In such switches we use a time-division backplane bus that is capable of transferring N cells – where N is the number of input ports – in a single cell arrival time. Each input controller is assigned its own cell (slot) time to transfer a cell over the backplane bus. The input controller appends the required output port number to the head of the cell and this is used by the set of output controllers to determine which output controller should read and buffer the cell. If more than one cell is received by an output controller in a single cell arrival time then these are queued in the controller. Also, for one-to-many communications (multicast), more than one output controller may be specified to receive the cell. Typically, this type of switch fabric is used in switch designs which have a relatively small number of ports, the number being limited by the speed of operation of the backplane bus and also the output controllers; for example, a 2.5 Gbps bus can support 16 × 155 Mbps or 4 × 622 Mbps duplex links/ports.

In a **space-division switch** the switch fabric comprises a matrix of interconnected **switching elements** that collectively provide a number of alternative paths through the switch. An example is shown in Figure 10.3(c). This is known as a **fully-connected switch matrix** since a path is provided from all input ports/controllers to all output ports/controllers. In the example shown, each input switching element is capable of passing a copy of each received cell to any of the output switching elements. The latter then receive the cells offered and pass them on to the output controller(s) for transmission. Although queuing is necessary in the input or output controllers, normally the aim is to avoid additional queuing within the switching fabric itself. To avoid internal queuing with this type of switch the cell transfer operation must be performed N times faster than the cell arrival rate, where N is the number of input ports. First the cell from the first input switching element is transferred, then the cell from the second element, and so on. Providing this can be done, no additional buffering is required within the switching matrix and the switch is said to be internally **non-blocking**.

We can deduce from Figure 10.3(c)that in a fully-connected switch the number of interconnection paths required through the switch – and hence output/input circuits associated with each switching element – grows as a function of N^2 and the speed of operation of the output switching element by N. In practice, this limits the maximum size of such switches. Hence most practical matrix switch designs use multiple switching stages, each made up of a number of smaller switching elements interconnected in a regular matrix. This also simplifies considerably the implementation of the switch fabric in integrated circuit form.

A switching fabric that comprises multiple switching stages is the **delta switch matrix**, an example of which is shown in Figure 10.4. As we can see, these switches are made up from an interconnected set of identical switching

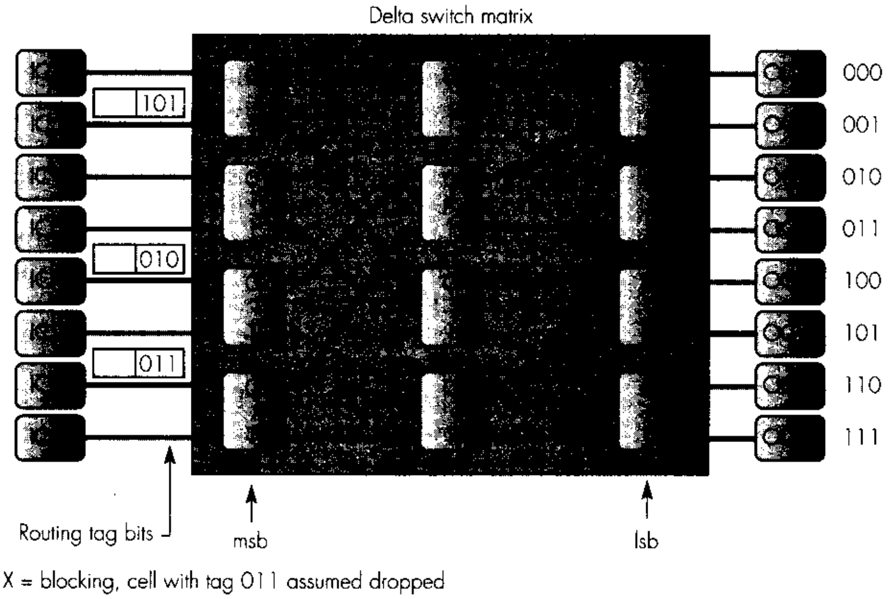


Figure 10.4 Delta switch matrix example.

elements. In this example, a 2×2 switching element is used although larger sizes are also used. In general, the number of switching elements per stage, X, is determined from the expression $X = M/N$ where M is the total number of input lines and N is the number of inputs per switching element. Also, the number of stages required, Y, is determined from the expression $N^Y = M$. In the example, $N = 2$ and $M = 8$ and hence three stages are required each comprising four switching elements.

The internal interconnections between switching elements are such that there is a path through the switch from any input to any output. Associated with each switching element is a routing control bit called the **routing tag**. If the tag bit is a binary 0, then a cell arriving on either input is routed to the upper output, while if the tag is a binary 1, then it is routed to the lower output. As we can deduce from Figure 10.4, the same set of three routing tag bits will route a cell through the matrix from any input port to the same output port. Such networks are said to be **self-routing**.

With this type of switch, to route cells through the switch matrix each input controller simply reads the new PCI and output port number – the routing tag – from its routing table, writes the PCI into the cell header, and then appends the routing tag to the head of the cell. The switching element at each stage along the path through the matrix then uses its own bit from the routing tag – most significant bit first – to perform its routing operation. In this way, routing is very fast and each cell arrives at its intended destination port regardless of the switch port on which it arrived.

The disadvantage of this type of switching fabric is **blocking**. Three example paths through the switch are shown as bold lines in Figure 10.4. As we can see, although the cell addressed to port 5 (101) has an unimpeded path through the matrix, those addressed to ports 2 (010) and 3 (011) both arrive at the second switching element simultaneously. Both require the same output line and blocking is said to occur.

There are a number of ways of overcoming blocking. One approach is for the switching element to discard one of the two cells. It is assumed that this approach is used in the example and that the cell with a tag of 011 is dropped. In support of this, note that not all ports will enter a cell into the matrix during each cell arrival time since only in the heaviest load conditions do cells arrive contiguously at all inputs. This means that under normal loads, several ports will receive idle/empty cells which do not need routing and hence in practice the probability of blocking is small. Nevertheless, in general, discarding cells in this way can lead to an unacceptably high cell loss rate in large switches.

A second approach is to perform the switching operation several times faster than the cell arrival rate, for example by allowing each cell into the matrix after the preceding cell leaves the first stage switching element. There is a limit to the speed of operation of the switching elements and their interconnecting links and hence for larger switching fabrics this becomes impractical on its own. A third approach is to introduce buffering into each switching element but this has the disadvantage of introducing additional delay to the switching operation. In practice, a combination of all three approaches is used in practical switch designs.

An example of a switching fabric that avoids internal blocking is shown in Figure 10.5. It is known as the **Batcher–Banyan switch**. With a delta switch, blocking occurs when either different inputs all require a path through the switch to the same output or, the paths through the switch between different input and output ports involve a common output line from a switching element. In the Batcher–Banyan switch, blocking is avoided firstly, by ensuring that no two cells entering the switching matrix require the same output port and secondly, within the switching matrix itself, by ensuring that there are no common interconnecting links within the paths through the switch.

To satisfy the first condition, the buffering of cells is performed in the input controllers instead of the output controllers. Then, if two (or more) cells arrive simultaneously at different input ports that require the same output port, just one cell is selected for transfer across the switching fabric and the other is queued in the input controller until the next cell transfer time. To satisfy the second condition, as Figure 10.5 shows, the (Banyan) switching matrix is preceded by a **(Batcher) sorting matrix** and the two are interconnected using what is called a **shuffle exchange**. The combined effect is that all cells arriving at the switching matrix are ordered such that they each follow a unique path through the switching matrix. Hence, providing independent routing paths are available within each switching element, the

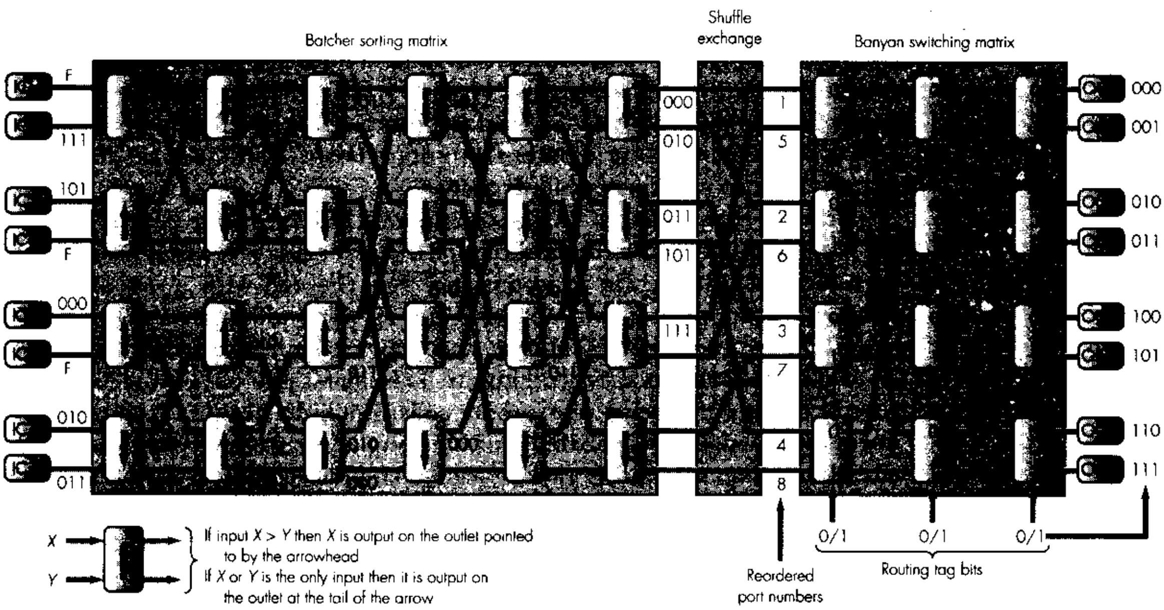


Figure 10.5 Batcher–Banyan switch matrix.

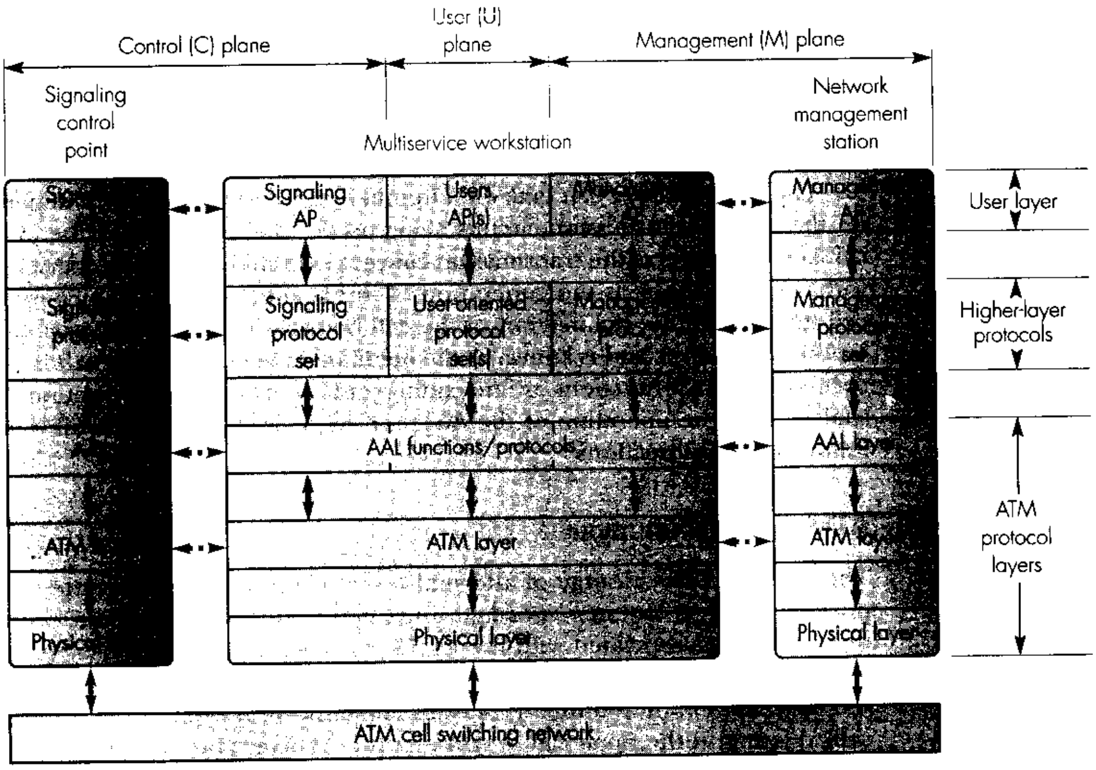
cells from all input ports can be switched simultaneously. The size of a Batcher sorting network grows by N ($\log N^2$) and can be considerable for large switches.

Another approach used in practical switch designs to reduce the blocking probability is a switching fabric that has multiple paths between each pair of input and output ports. For example, with the simple delta switch, this can be achieved by replicating the total switch matrix an appropriate number of times. Each input controller uses a different matrix to transfer each cell. Alternatively, the basic Banyan switching matrix can be extended by adding extra switching stages. In both cases, the number of additional stages can be selected so that the blocking – and hence cell loss – probability is at an acceptable level.

Another issue which has an impact on switch design is multicasting. Recall from Section 9.6.9 that this requires all the cells from each workstation in a (multicast) group to be sent to all the other workstations in the group. In practice, the most efficient way of achieving this is to use the switches to route copies of cells to multiple destinations. Although this can be done readily with both the time-division and fully-connected switch designs, with matrix switches, because of their self-routing property, additional switching stages are needed if multicasting is to be supported. Further details relating to both these issues can be found in the bibliography for this chapter.

10.4 Protocol architecture

As we show in Figure 10.6, the ATM protocol architecture supports three separate application functions (planes). These are the control (C) plane, the user (U) plane and the management (M) plane. The protocols associated with the **C-plane** are concerned with signaling; that is, the setting up and clearing of on-demand VCs. Typically, these are set up and cleared using a signaling protocol set in the station communicating with a similar protocol set in the network **signaling control point**. The protocols in the **U-plane** depend on the application and, in general, communicate on a user-to-user (peer) basis with a similar protocol set in the destination station. The protocols in the **M-plane** are concerned with the management of the station; examples include reporting any error conditions that may arise during normal operation to the network management station and receiving notifications of the



AP = application process
AAL = ATM adaptation layer
ATM = asynchronous transfer mode

Figure 10.6 ATM protocol architecture.

virtual path/channel identifiers that have been allocated for permanent VCs (PVCs). These three application functions then use the services provided by the three lower ATM protocol layers for the transfer of the associated messages over the cell-based ATM switching network.

The ATM network supports a range of different services. The use of cell switching and transmission within the network is transparent to the upper application protocols which view the ATM network simply as a flexible facility for the transfer of information relating to any media type. To achieve this transparency, the highest of the three ATM layers is known as the **ATM adaptation layer (AAL)**. As the name implies, it performs an adaptation (convergence) function between the service provided to the user layer above – for example the transfer of a data frame between two legacy LANs – and the cell-based service provided by the underlying ATM layer.

To support the various information sources, the AAL layer provides a range of alternative service types known as **service classes**. Associated with each service class is a different adaptation function/protocol which converts the source information into streams of 48-octet segments. It passes these to the **ATM layer** for transfer across the network. The ATM layer is concerned with adding the correct cell header information to each segment and multiplexing the cells relating to different connections into a single stream of cells for transmission over the network. It is also concerned with the demultiplexing of received cell streams and relaying their contents to the appropriate AAL protocol at the destination.

The **physical layer** can take on a number of different forms and depends on the type of transmission circuits being used. The upper part of the physical layer is known as the **transmission-convergence sublayer** and is concerned with such functions as the generation of the header check sequence in the cell header and the delineation of the cell boundaries. The lower part takes on different forms and is known as the **medium-dependent sublayer**. It is concerned with such functions as line coding and bit/clock synchronization. We shall discuss the operation of the AAL and ATM layer in more detail in the following two subsections.

10.4.1 ATM adaptation layer

The AAL provides a range of alternative service types/classes for the transport of the byte streams/message units generated by the various higher protocol layers associated with the U-, C-, and M-planes. It converts the submitted information into streams of 48-octet segments and transports these in the payload field of multiple ATM cells. Similarly, on receipt of the stream of cells relating to the same call, it converts the 48-octet information field contained within each cell into the required form for delivery to the particular higher protocol layer.

The service types are classified according to three criteria: the existence of a time relationship between the source and destination users (for example voice), the bit rate associated with the transfer (constant or variable), and the connection mode (connection-oriented or connectionless). Currently, five service types have been defined. They are referred to as AAL 1–5 and their interrelationship, based on these criteria, is illustrated in Figure 10.7(a).

Both AAL 1 (Class A) and AAL 2 (Class B) are connection-oriented and there exists a timing relationship between the source and destination users. The difference between the two is that AAL 1 provides a **constant bit rate (CBR) service** while AAL 2 provides a **variable bit rate (VBR) service**. An example use of AAL 1 is for the transfer of the constant bit rate byte stream associated with a voice call, for example, 1 byte per 125 μs. AAL 1 is also known as **circuit (switched) emulation**. An example use of AAL 2 is for the transmission of the variable bit rate stream associated with compressed video. Although video produces frames at a constant rate, a video codec will produce frames containing a variable amount of compressed data.

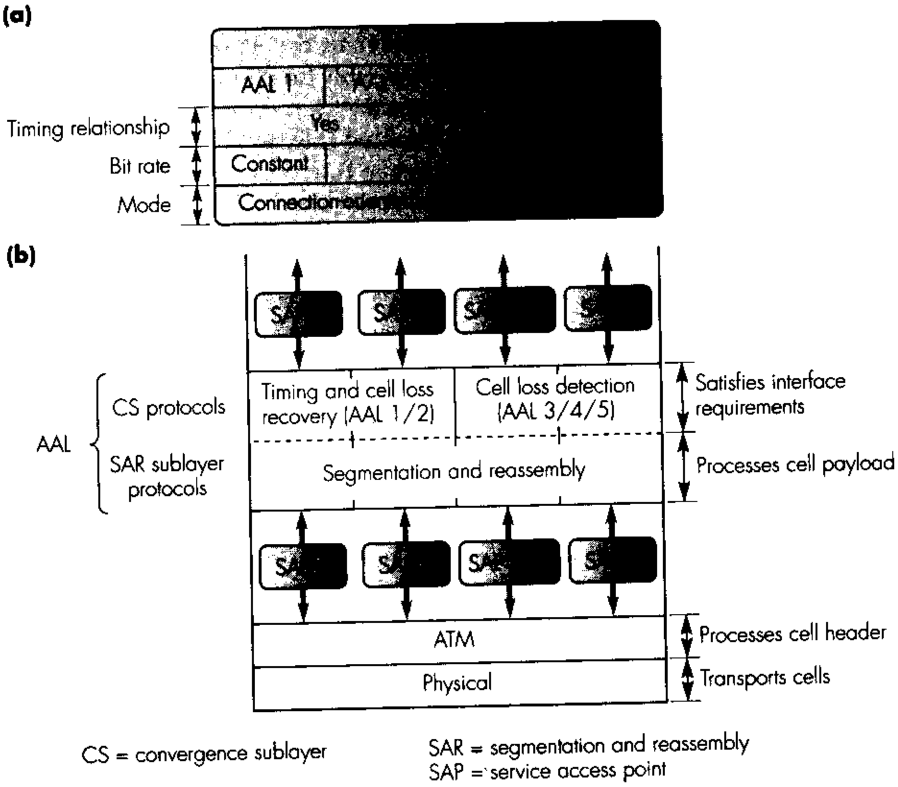


Figure 10.7 ATM adaption layer: (a) service class relationship; (b) sublayer protocols and their functions.

There is no timing relationship between source and destination with either AAL 3/4 (Class C/D) or AAL 5. Initially, AAL 3 was defined to provide a connection-oriented, VBR data service. Later, this service type was dropped and it is now merged with AAL 4. Both AAL 3/4 and AAL 5 provide a similar connectionless VBR service. An example use is for the transfer of data frames between two legacy LANs – through a remote bridge or router for example – or for the transfer of frames containing multimedia information between a workstation and a server. It is also used for the transfer of the message units associated with the signaling and management protocol sets in such workstations. In all of these cases, as we shall see in Section 10.5.1, although the service offered is connectionless, the resulting cell streams produced by the AAL layer are transferred over previously established PVCs, for example, to the signaling control point or network management station.

In order to implement this range of services, the AAL comprises two sublayers as we show in Figure 10.7(b). The **convergence sublayer (CS)** performs a convergence function between the service offered at the layer interface and that provided by the underlying ATM layer. The **segmentation and reassembly (SAR)** sublayer performs the necessary segmentation of the source information ready for transfer in the 48-octet payload field of a cell and also, the corresponding reassembly function at the destination prior to the delivery of the source information.

Since the submitted information differs for each service type, there is a different convergence function – and hence CS protocol – associated with each service type. Associated with each protocol is a **service access point (SAP)** which is used to direct all information submitted for transfer – the service data unit (SDU) – to the appropriate CS protocol. Similarly, there are four different types of SAR protocol, each with its own PDU structure. As we shall now see, each utilizes the 48-octet information field in each cell in a different way.

AAL 1

For this type of service the CS protocol endeavors to maintain a constant bit rate stream between the source and destination SAPs. The bit rate can range from a few kilobits per second – for example, for a single compressed voice call – to tens of megabits per second – for example, for a compressed video. However, the agreed rate must be maintained, even when occasional cell losses or cell transfer time variations occur. Cell losses are overcome in an agreed way, for example, by inserting dummy bits/bytes into the delivered stream. Cell transfer delay variations are compensated for by buffering segments at the destination: the output of the bits/bytes relating to a call is started only after a predefined number of segments have been received, this number being determined by the user bit rate. Typical figures are two segments at kilobit rates and 100 segments at megabit rates. The use of buffering at the destination also provides a crude way of overcoming any small variations between the input rate at the source interface and the output rate at the destination; for example, if each is based on a separate clock.

The format of each PDU associated with the SAR protocol is shown in Figure 10.8(a). To detect segment losses, the first octet contains a 4-bit *sequence number* (*SN*) and an associated 4-bit *SN protection* (*SNP*) field which is used to protect the sequence number against single bit errors. The sequence number is itself made up of a 3-bit *sequence count* field – for the detection of lost cells – and a single *convergence sublayer indication bit.* The latter can be used for the transfer of timing and/or other information relating to the payload field. The sequence number protection field comprises a 3-bit CRC, generated by the polynomial x^3+x+1, and an even parity bit. The latter is used to detect errors in the CRC.

AAL 2

With this type of service, although there is a timing relationship between the source and destination SAPs – determined by the frame rate, for example, for compressed video – the amount of information associated with each compressed frame may vary from one frame to the next. The CS protocol at the source receives bursts of information at the frame rate with each burst containing a variable amount of information. Hence the peer CS protocol at the destination must endeavor to output the received information in this same way even when occasional cell losses or cell transfer time variations occur. With AAL 2, the time variations are overcome using similar techniques to those described for AAL 1.

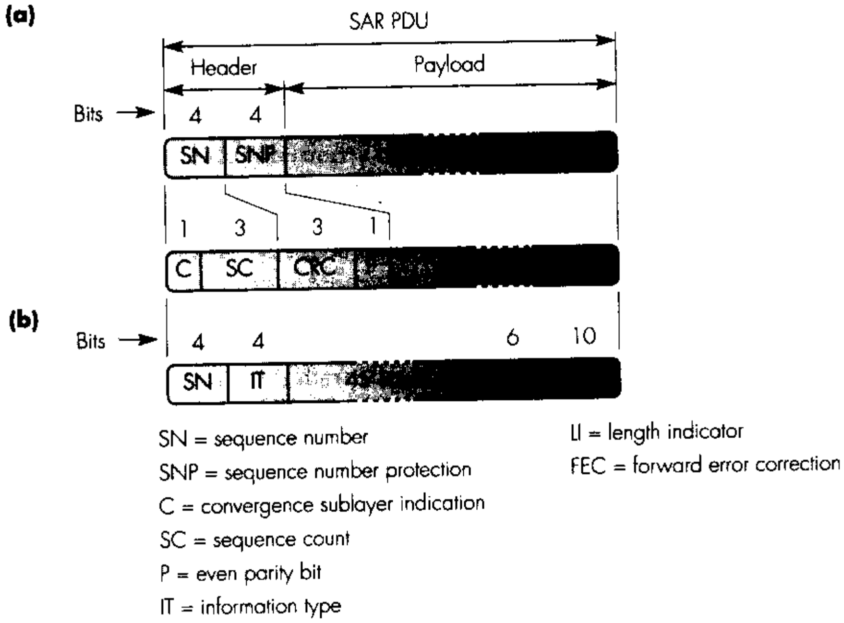


Figure 10.8 SAR protocol data unit types: (a) AAL 1; (b) AAL 2.

The format of each PDU associated with the corresponding SAR protocol is shown in Figure 10.8(b). As with AAL 1, the sequence number is present to detect (and recover from) lost cells and to carry timing information. The *information type* (*IT*) field indicates either the position of the segment in relation to a submitted message unit – for example, a compressed frame – or whether the segment contains timing or other information. The three segment types relating to positional information are beginning of message (BOM), continuation of message (COM), and end of message (EOM). Also, because of the variable size of submitted message units, the last (EOM) segment may not be full and hence the length indicator (LI) at the tail indicates the number of useful bytes in the segment. Finally, the forward error correction (FEC) field enables bit errors to be detected and a selection corrected.

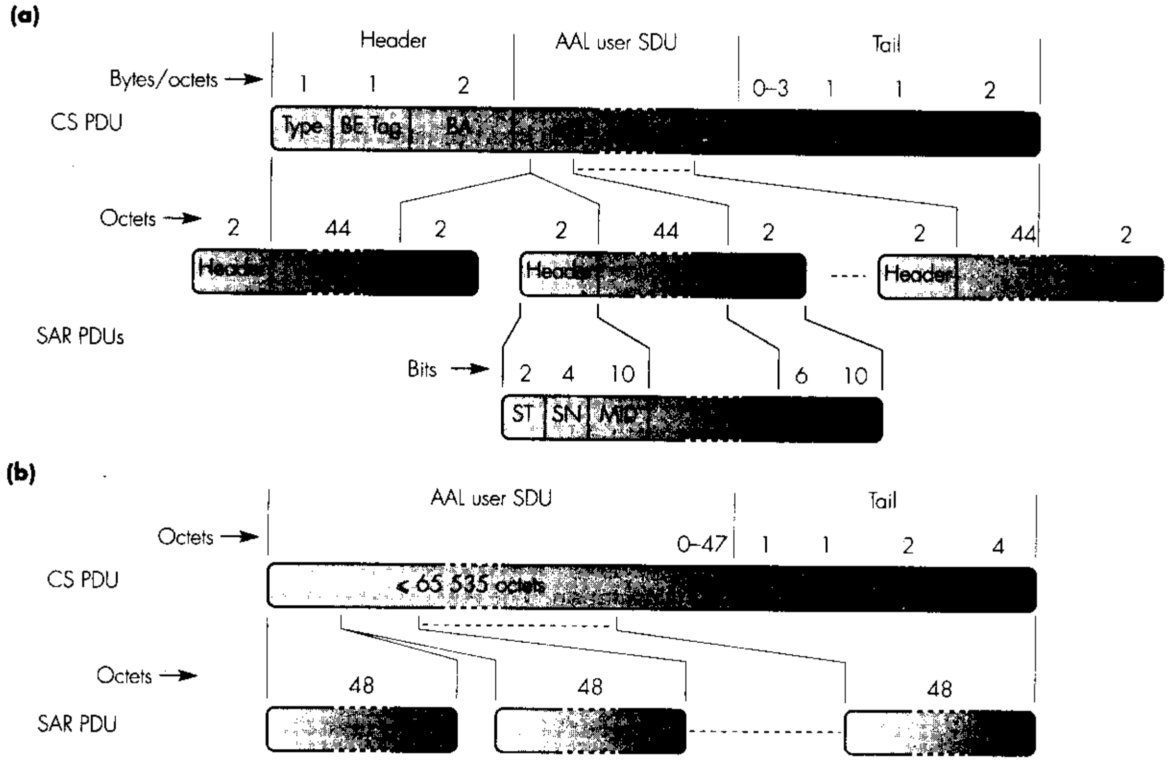
AAL 3/4

AAL 3 was defined initially to provide a connection-oriented data transfer service. Later, this type of service was dropped and combined with AAL 4. AAL 3/4 provides a connectionless data transfer service for the transfer of variable length frames up to 65 535 bytes in length. Error detection and other fields are added to each frame prior to its transfer and the resulting frame is padded so that it is an integral multiple of 32 bits.

The operation of AAL 3/4 is best described by considering the format of the PDUs associated with both the CS and SAR protocols. The format of the additional fields added by the CS protocol to each submitted user SDU – at the corresponding SAP – is shown in Figure 10.9(a). The figure also shows how the resulting CS PDU is segmented by the SAR protocol into multiple 48-octet SAR-PDUs.

The header and trailer fields added to the submitted SDU by the CS protocol at the source are used by the peer CS protocol at the destination to detect any missing or malformed SDUs. The *PDU-type* field is a legacy of the earlier AAL 3 which required multiple types of PDU. It is set to zero with AAL 3/4. The *begin-end* (*BE*) *tag* is a modulo-256 sequence number and is repeated in the trailer for added resilience. It enables SDUs to be delivered at the user interface in the same sequence as they were submitted although, again, this facility is not normally used with a connectionless service. The *buffer allocation* (*BA*) field is inserted in the header by the source to help the CS protocol at the destination allocate an appropriate amount of (buffer) memory for the complete SDU. At the trailer, the *pad* field is used to make the total number of octets in the complete CS-PDU an integral multiple of 4 octets. Similarly, the *alignment* (*AL*) field is a single (dummy) octet to make the trailer 4 octets also. Collectively this leads to easier memory management at the destination. The *length* field indicates the total length of the complete PDU and this is used by the receiving protocol to detect any malformed SDUs.

On receipt of each CS-PDU, the SAR protocol segments this into multiple 48 octet segments – SAR-PDUs – as shown in Figure 10.9(a). In the header, the *segment type* (*ST*) indicates whether the segment is the first, continuation, last, or only segment resulting from the segmentation of the CS-PDU. The



CS = convergence sublayer
SAR = segmentation and reassembly
BE = begin–end
BA = buffer allocation
AL = alignment
ST = segment type
SN = sequence number
MID = message identifier
LI = length indicator
UU = AAL layer user-to-user identifier
CPI = common part identifier

Figure 10.9 CS and SAR PDU formats: (a) AAL3/4; (b) AAL 5.

sequence number (*SN*) is used to detect missing segments. In practice, a device like a network server may have a number of frames – and hence CS-PDUs – being received concurrently from multiple sources. Hence for the receiving SAR protocol to relate each incoming segment to the correct PDU, the source SAR protocol adds the same *message identifier* (*MID*) field to the head of each segment relating to the same CS-PDU. At the tail, the *length indicator* (*LI*) indicates the number of useful octets in the segment since the CS-PDU is not necessarily an integral number of 44 octet segments. Clearly, this field has meaning either in the last segment relating to a multisegment CS-PDU or, if it is the only segment. The *CRC* is used to detect possible transmission errors introduced during the transfer of segments.

AAL 5

Because of the origin of AAL 3/4, the CS-PDU contains a number of fields in the header which were included primarily to support a connection-oriented service. The MID field at the head of each SAR-PDU – which is present to enable the destination to relate segments to a specific frame – performs a similar function to the protocol connection identifier – VPI/VCI – present in the header of each ATM cell. As a result, the alternative AAL 5 service class was defined. This provides a similar service to AAL 3/4 but with a reduced number of control fields in both the CS and SAR PDUs. It is known as the **simple and efficient adaptation layer (SEAL)**. As with AAL 3/4, its operation is best described by considering the format of the PDUs associated with both the CS and SAR protocols. These are shown in Figure 10.9(b).

As we can see, with AAL 5 there is no header associated with the constructed CS-PDU. Also, the pad field in the trailer is longer (0–47 octets) so that the length of each CS-PDU can be an integral number of complete 48-octet segments. This has the advantage that the fields at the trailer of the CS-PDU are always the last 8 octets of the last segment, which leads to faster processing at the destination. The AAL *user-to-user identifier* (*UU*) enables the two correspondent user layers to relate the AAL SDU to a particular SAP. The use of the *CPI* field has yet to be defined. It is included to support future functions and currently to make the tail an even number of octets. The *length* field indicates the number of octets in the user data field and is an integer value in the range 0 to 65 535. The *CRC* field detects the presence of any transmission errors in the reassembled CS-PDU. If errors are detected, then the user layer above is informed when the SDU contained within the PDU is delivered. In this way, it is left to the user layer to decide what action to take: in some instances this will be to discard the SDU – for example, if it contains normal data – while in others, it will be to accept it and take appropriate recovery steps – for example, if the SDU contents relate to a video or voice portion of a multimedia document.

As we can see, there is no head or tail associated with the SAR PDU. Each comprises the full 48 octets and the SAR protocol is said to be null. The lack of a MID field at the head of each segment means that the segments relating to the same CS-PDU are identified using the field in the header of the ATM cell that is used to transport the segment. With AAL 5, the SDU type bit in user data cells – also known as **ATM-layer user-to-user (AUU) cells** – is used to indicate whether the cell contents form the beginning or continuation of the CS-PDU (binary 0) or the end (binary 1). Although this means the operation of the AAL layer is now linked with that of the ATM layer, it is done to improve the efficiency of the segmentation process. AAL 5 is also used as the AAL layer in the C-plane for the segmentation and reassembly of messages associated with the signaling protocol. It is then known as the **signaling AAL** or **SAAL**.

10.4.2 ATM layer

The ATM layer performs all the functions relating to the routing and multiplexing of cells over VCs, which may be semipermanent or set up on demand. Its main function is to assign a header to the segment streams generated by the AAL relating to a particular call. Similarly, on receipt of cell streams, its function is to remove the header from each cell and pass the cell contents – segments – to the appropriate AAL layer protocol.

To perform these functions, the ATM layer maintains a table that contains a list of VCIs. Normally, in ATM LANs for example, all switching is carried out using VPIs only and the VCI field in the cell header is then used to multiplex/demultiplex the cells relating to specific calls/transactions which are transported over the same path. In the case of semipermanent VCs, the VPIs are downloaded by network management – via the network management protocol stack – and in the case of on-demand VCs, the VPIs are obtained using the appropriate signaling protocol set. In both cases the cell streams relating to the management/signaling protocol messages are transferred over permanently assigned VCs.

In a public network, the ATM layer offers various service classes each of which has been specified to meet a particular type of application requirement. These are:

- **constant bit rate (CBR):** this supports isochronous traffic such as uncompressed speech, audio, and video;
- **variable bit rate/real time (VBR/RT):** this supports variable bit rate traffic with real-time requirements. An example is the transmission in real time of a compressed video. There is a declared mean and peak rate and, if the peak rate is exceeded, cells are tagged at the network interface to indicate they can be dropped if necessary;
- **variable bit rate/non-real time (VBR/NRT):** this supports variable bit rate traffic but with no real-time requirements. An example is the transfer of a file containing compressed video from a server to a client workstation;
- **available bit rate (ABR):** this supports bursty traffic with a known minimum, mean, and peak rate. The network guarantees the mean rate but provides the additional rate according to competing demands on the available transmission capacity. If the network is unable to provide the additional rate, however, it informs the user to reduce the input rate;
- **unspecified bit rate (UBR):** this service does not provide any guarantees nor is the user informed if the network is unable to transfer the submitted information. The service offered, therefore, is similar to that provided by a best-effort network such as a LAN or the Internet.

Associated with each service class is a network quality of service (QoS) guarantee which forms the basis of a contract between the network provider and the network user. Essentially, the provider agrees to meet the bandwidth, delay, and delay variation (jitter) requirements of the cell stream associated with a user application providing the cell stream conforms to an agreed set of parameters. A range of QoS parameters have been defined and, for each contract, a set of parameter values is negotiated which the network provider agrees to meet or exceed. Collectively, the parameters form what is called the **traffic descriptor** of the call and, to meet the requirements of the various types of call, a different set of parameters may be defined for the cell stream flowing in each direction of a connection.

The parameters relating to the user include:

- **peak cell rate (PCR):** the maximum rate the source will enter cells;
- **sustained cell rate (SCR):** the average rate the source will enter cells;
- **minimum cell rate (MCR):** the minimum cell rate that is acceptable by the source;
- **cell delay variation tolerance (CDVT):** the maximum level of variation in intercell times.

The parameters relating to the network include:

- **cell loss ratio (CLR):** the maximum ratio of cells that will not be delivered owing to transmission errors or excessive delays within the network;
- **cell transfer delay (CTD):** the average transfer delay of cells across the network;
- **cell delay variation (CDV):** the average variation in cell transfer delay.

In order to enforce the contract, for each cell entered into the network and subsequently delivered, the network determines whether the parameters for this call/VC conform to the agreed contract. The algorithm used is known as the **generic cell rate algorithm (GCRA)** and the principles on which it is based are shown in Figure 10.10.

As we can see, four examples of cell inter-arrival times are given:

(i) Cell 2 arrives according to the agreed PCR and hence is accepted.

(ii) Cell 2 arrives early but within the agreed CDVT. Hence providing the new SCR is not violated, the cell is accepted. Note, however, that the expected time of arrival of cell 3 is t_0 +2T.

(iii) Cell 2 arrives later than the agreed PCR but, since a source can enter cells at a lower rate than the agreed PCR, it also is accepted. Note that the expected time of arrival of cell 3 is the time cell 2 arrived plus T.

(iv) Cell 2 arrives earlier than the agreed CDVT and hence violates the contract. Normally, the network will either set the cell loss priority (CLP) bit in the cell header or discard the cell depending on the current loading of the network.

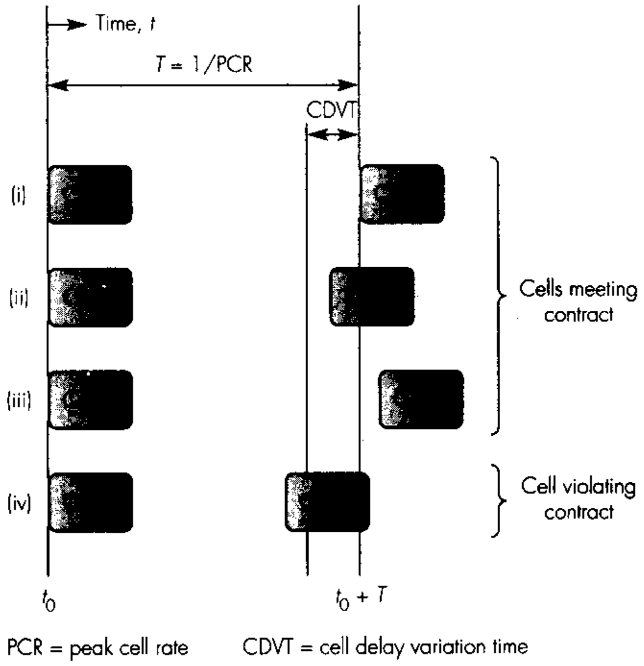


Figure 10.10 Principle of operation of generic cell rate algorithm.

10.5 ATM LANs

ATM LANs have been designed to meet the high bandwidth demands of multimedia applications and hence are an alternative to the LANs based on fast Ethernet which we described in Section 8.7. A typical site LAN is shown in Figure 10.11.

As we explained earlier in Section 10.3, an ATM switch has a defined number of ports and its function is to provide a high bit rate switched communications path between ports. The cost of a switch is a function of the number of ports that it supports. If all the stations are connected directly to the switch then as the deployment of multimedia PCs/workstations increases, switches with a very large number of ports are required. However, not all stations require network services at the same time. To minimize the number of ports, groups of stations – for example, in a building – are connected to the switch through a **remote concentrator unit (RCU)**. There is no switching function in the RCU and its role is simply to multiplex/demultiplex the cell streams from/to those stations that are involved in a network transaction onto/from the link connecting the concentrator to a switch port. In this way, providing the interconnecting link has sufficient capacity to support the

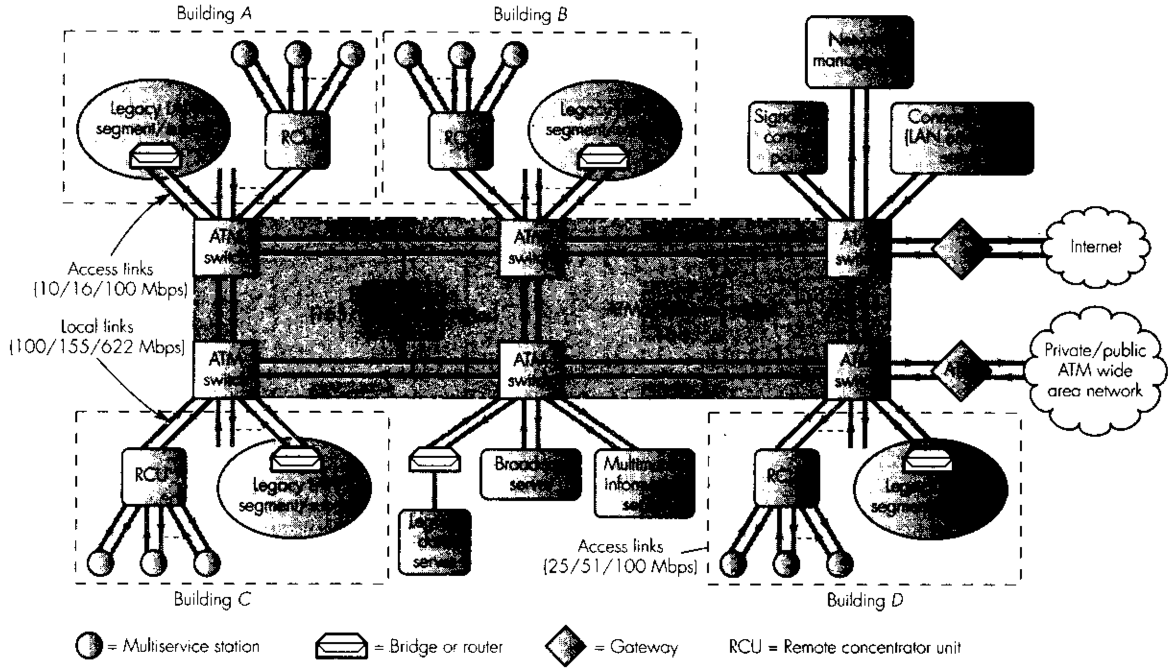


Figure 10.11 ATM LAN schematic.

anticipated number of concurrent transactions, the number of switch ports required is reduced considerably.

Prior to sending any information relating to a call, a communications path through the network is first established. All the cells relating to the call are constrained to follow this same path and are delivered in the same order as they were submitted. Recall that the conversion of all information into streams of fixed-sized cells has the advantage that the cells relating to the different types of call can be switched in a uniform way and independent of the type of media which they contain. In addition, the use of cells has advantages in terms of the utilization of transmission bandwidth.

The applications supported by a multimedia PC/workstation include telephony, videophony, conventional data networking, and access to a range of related servers. These include, in addition to those present with existing data networks – electronic mail, printing services, and so on – a broadcast server, a database server supporting multimedia Web pages, and so on. For example, the **broadcast server** – which performs the same functions as an MCU – enables a user at a multimedia workstation to set up on demand a videoconferencing session between three or more similar workstations by sending the integrated media streams output from all the workstations involved directly to the server in real time. The server relays the appropriate media streams to the other workstations as the conference proceeds.

Similarly, the server holding multimedia Web pages enables a user at a workstation to access a particular page and then interactively browse through it.

In practice, ATM LANs are being introduced in an incremental way and currently a majority of LAN installations are still of the Ethernet type. Within the context of ATM LANs, these are often referred to as **legacy LANs**. The major bottleneck associated with such networks is access to servers, since these require significant bandwidth to support multiple concurrent transactions. Hence in addition to providing direct access to the newer multimedia servers, the backbone switches also provide connections both to the existing data-only LANs in each building – for example through bridges – and to the servers associated with them. Typically, these are all interconnected by a set of point-to-point virtual connections (VCs). In the context of high-speed LANs, the interconnected set of backbone switches can be viewed as providing the same function as a high-speed backbone subnetwork.

All communications across the ATM network are carried out over previously established VCs. These can be set up either on demand by a user or semipermanently by network management. With **on-demand connections**, the user device, before sending any information cells, sends a request for a **switched virtual connection (SVC)** to be set up between itself and the required destination. This can be done either in a distributed way involving all the switches in the connection or using a central control unit known as the **signaling control point (SCP)**. This is responsible for the overall management of both the transmission bandwidth and the setting up and clearing of switched connections through the network. On receipt of the request – known as a **signaling message** – the SCP first determines the availability of both the required destination and the transmission bandwidth appropriate to the call across the network. All the end-user stations are connected to the SCP – normally a powerful workstation – by a separate VC and, assuming the required destination and network resources are available, the SCP sets up routing information in the switch network to link the two user stations (VCs) involved in the call. It then informs the originator of the call that it can commence sending information cells. All the signaling messages associated with the setting up and clearing of calls/connections are transferred across the network to and from the SCP in the form of cells over a separate set of VCs that are permanently set up for this function. The latter are known as **signaling virtual channel connections (SVCCs)**.

To access networked servers such as electronic mail and multimedia information servers, because the ATM network is connection-oriented, before any frames can be transferred, a VC must be in place between each workstation and the set of servers. In practice, the number of servers may be large and this requirement can be met by setting up **permanent VCs (PVCs)** between all user stations and servers and a central data forwarding point known as the **connectionless server (CLS)** or, because it provides a similar service to that provided by a legacy (broadcast) LAN, the **LAN emulation server (LES)**. We shall discuss the operation of this and the SCP in more detail in the next section.

The permanent virtual connections associated with the various services are all set up under the overall control of the network management station. A reserved VC is permanently in place between the **network management station** and the control processor in each RCU and backbone ATM switch. Also between all multimedia stations and servers and their network point of attachments. These are used by the network manager to set up the VCs associated with the various services. The network manager uses these connections to download routing information for entry into the routing tables held by these devices.

As with existing data-only networks, users of multimedia stations, as well as needing to communicate with other users at the same site, also want to communicate with users connected to an ATM LAN at a different site. Hence an ATM LAN has a gateway to the Internet and also to newer private/public wide area ATM networks. To meet these needs, new generations of private networks based on the ATM are being introduced. Also, a number of telecom providers are introducing a new generation of public network based on the same technology. As we shall see in Sections 10.6 and 10.7, these consist of ATM MANs which, in turn, are interconnected together to form an ATM wide area network.

Example 10.1

A segment of an ATM network is [illegible] alongside each RCU/switch are the [illegible] manent VCs are to be set up by [illegible] B, C and D, firstly to the SCP for [illegible] information transfer – and secondly [illegible] relating to connectionless calls. Also [illegible] to connect the server to the CLS/[illegible] for RCU 1 and RCU 2 and SW 1 [illegible] assuming VP-only switching is used [illegible] the RCUs, the VPI/VCI field is used to [illegible]

Answer:

A suitable set of routing table entries is given in Figure 10.12(b). Note the following points when interpreting the entries:

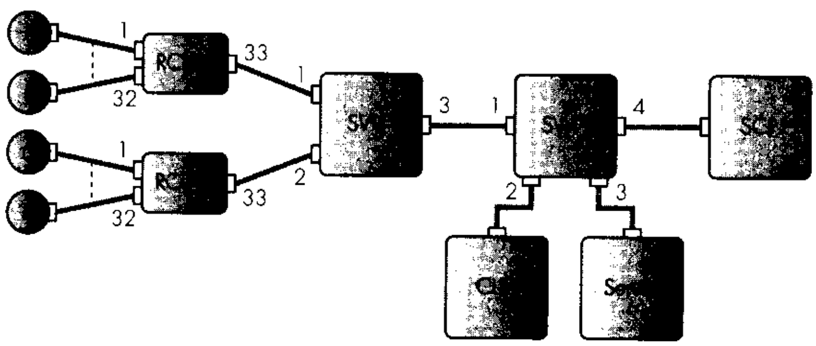
- The SCP, CLS, and server are all connected to their switches by separate transmission lines.
- For on-demand calls, two separate VCs are shown, one between each workstation and the SCP for the signaling messages associated with a call – the signaling channel (SC) – and the other for the cell streams associated with the call – the call channel (CC). Since the latter are to be semipermanent, they are shown set up between each workstation and SW 2. Alternatively, they could be set up on demand by the SCP between each pair of workstations involved in a call.
- For connectionless traffic, a separate VC is required between each workstation and the CLS and also between the CLS and the server.
- The SCP, CLS, and server all use the combined VPI/VCI in the cell header to identify [illegible] specific calls/server transactions.

10.1 Continued

[illegible] each RCU, the VCI field identifies the port [illegible] within each virtual path. Also, in this [illegible] virtual paths are required per RCU rather than [illegible] allows the approach to be scaled to large installations.

- [illegible] the network, all switching is carried out using VPIs only.
- [illegible] on-demand call, the SCP creates entries in the routing [illegible] to link the port/VCI of the calling party to that of the [illegible]
- For connectionless traffic, when relaying the cell streams received from each station to the server, the CLS assigns a new VPI/VCI. Also, in order to relay the cell streams in the reverse direction, it maintains a table that maps the incoming VPI/VCI from the stations to those used for communicating with the server.
- When responding to a request, the server uses the same VPI/VCI values for the cells making up the response as were used in the request.

(a)



(b)

RCU 1:

	In Port	In VPI	In VCI	Out Port	Out VPI	Out VCI
SC	1	0	1	33	1	1
CC	1	0	2	33	2	1
CLS	1	0	3	33	3	1
⋮						
SC	32	0	1	33	1	32
CC	32	0	2	33	2	32
CLS	32	0	3	33	3	32

In Port	In VPI	In VCI	Out Port	Out VPI	Out VCI
33	1	1	1	0	1
33	2	1	1	0	2
33	3	1	1	0	3
⋮					
33	1	32	32	0	1
33	2	32	32	0	2
33	3	32	32	0	3

Figure 10.12 ATM LAN routing example: (a) network segment; (b) example routing table entries.

(b cont.)

RCU 2:

	In			Out			In			Out		
	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI
SC	1	0	1	33	1	1	33	1	1	1	0	1
CC	1	0	2	33	2	1	33	2	1	1	0	2
CLS	1	0	3	33	3	1	33	3	1	1	0	3
				⋮						⋮		
SC	32	0	1	33	1	32	33	1	32	32	0	1
CC	32	0	2	33	2	32	33	2	32	32	0	2
CLS	32	0	3	33	3	32	33	3	32	32	0	3

SW 1:

	In			Out			In			Out		
	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI
SC	1	1	X	3	1	X	3	1	X	1	1	X
CC	1	2	X	3	2	X	3	2	X	1	2	X
CLS	1	3	X	3	3	X	3	3	X	1	3	X
SC	2	1	X	3	4	X	3	1	X	2	4	X
CC	2	2	X	3	5	X	3	2	X	2	5	X
CLS	2	3	X	3	6	X	3	3	X	2	6	X

X = 1–32

SW 2:

	In			Out			In			Out		
	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI	Port	VPI	VCI
SC	1	1	X	4	1	X	4	1	X	1	1	X
CC	1	2	X	1	Y	X	1	Y	X	1	2	X
CLS	1	3	X	2	3	X	2	3	X	1	3	X
SC	1	4	X	4	4	X	4	4	X	1	4	X
CC	1	5	X	1	Y	X	1	Y	X	1	5	X
CLS	1	6	X	2	6	X	2	6	X	1	6	X
CLS'	2	10	Z	3	10	Z	3	10	Z	2	10	Z

X = 1–32
Y = 2/5
Z = 1–64

CLS:

In		Out		In		Out	
VPI	VCI	VPI	VCI	VPI	VCI	VPI	VCI
3	X	10	Y	10	Y	3	X
6	X	10	Y	10	Y	6	X

X = 1–32
Y = 2/5

SCP: Calls in progress

Signaling channel		Call channels						Call type details
		In			Out			
VPI	VCI	Port	VPI	VCI	Port	VPI	VCI	
1	X	1	2	X	1	Y	X	–
4	X	1	5	X	1	Y	X	–

SC = signaling channel
CC = call channel
CLS = workstation/CLS channel
CLS' = CLS/server channel

Figure 10.12 Continued

10.5.1 Call processing

As we saw in Figure 10.11 and its associated text, there are two types of traffic associated with an ATM LAN: that relating to the message flows exchanged between the distributed community of ATM-enabled stations and their associated servers, and that relating to the message flows exchanged between the bridges/routers that form the interface with the various legacy LAN segments/subnets and their associated servers. Also, in the case of multimedia stations, there are two types of call: one connection-oriented and the other connectionless. The first relates to network services such as telephony and videophony, while the second relates to more conventional data services similar to those used with legacy LANs. We shall consider the three types separately.

Connection-oriented calls

Services such as telephony and videophony require a separate VC to be set up between any pair of workstations for the duration of the call. The standards relating to this type of service are based on those used to set up calls in an ISDN since, in principle, the operation of setting up and clearing calls is the same as that used to perform the same function within an ISDN. All the (signaling) messages relating to the setting up and clearing of calls are carried over a separate channel – called the **signaling virtual channel connection (SVCC)** – which is independent of that used for carrying the message flows associated with the call. When the network is first configured or a new outlet added, a semipermanent VC is set up by network management between each network outlet and the central signaling control point. This is analogous to installing a physical wire connection between each telephone outlet and a local (telephone) PBX.

When an SVCC is set up by network management, an entry is also made in the routing table of the SCP. This consists of the ATM network address of the workstation together with the allocated routing address; that is, the VPI/VCI addresses used in the headers of cells that are received/output by the SCP from/to this workstation. The network address of each station is analogous to that of a telephone outlet and hence is a hierarchical address that uniquely identifies the station in the context of the total network. These have a standard format and, in ATM LANs, are 20-byte addresses.

To initiate the setting up of a call, the user of the station follows a dialog which involves specifying the address of the required recipient of the call and also the call type – telephony, videophone, and so on. This, in turn, results in the exchange of signaling messages using the signaling protocol set in the station and a similar protocol set in the SCP computer. The protocol architecture to support this was shown earlier in Figure 10.6 and the signaling protocol set is defined in **Recommendation Q.2931**. This is based on the ISDN signaling protocol set to ascertain whether the user is prepared to receive a call. If the response is positive, this is communicated back to the SCP which proceeds to set up a (duplex) VC across the network by adding entries into each switch routing table that links the port/VPI/VCI of the calling station with that of the called station. The SCP then informs the calling station that a VC has been set up and the call commences. In a similar way, either user can invoke the closing down of the connection at any time by initiating the sending of appropriate disconnection messages over the separate signaling channel.

To set up a call involving multiple stations – for example, for a teleconferencing or videoconferencing session – all the stations must be fully interconnected. For small numbers of stations this can be done by the SCP directly using a similar procedure to that used for a two-party call. Since in some instances the number of stations involved may be large, an alternative approach is to route all the information flows relating to such calls via a central routing point called the broadcast server.

Using this approach, to set up a conferencing session, the initiator of the session communicates with the SCP as before – using the signaling channel – but this time indicates a conferencing call – and its type – and the addresses of the stations involved. The SCP, in turn, communicates with the user of each of these stations – using their signaling channels and associated protocol set – to ascertain their availability and willingness to participate in a conferencing session. On receipt of their responses, the SCP initiates the setting up of a switched VC between each station that returns a positive response and the broadcast server. It then informs, firstly, the broadcast server of the type of call and the set of switched VCs involved, and secondly, the users of the stations that the session can commence. The session then proceeds with the broadcast server relaying the information flows received from each station out to all the other stations. These play out the received information under the control of the user of the station, for example, by using a separate window for each member of a videoconferencing session.

Connectionless calls

As we mentioned earlier, ATM-enabled stations are often introduced in an incremental way as existing frame-based LAN segments are replaced by cell-based (ATM) segments. A major issue when considering connectionless working is interworking, not only between ATM (cell-based) stations, but also between such stations and stations that are connected to legacy-LAN segments.

As we explained in Chapter 8, most large legacy LANs consists of a number of LAN segments of the same type interconnected by bridges. Alternatively, if the segments are of different types, as we saw in Chapter 9, they are interconnected by subnet routers. Recall that when using bridges all routing is carried out at the MAC sublayer. When using routers, however, routing is carried out at the IP/IPX layer. Both the MAC and IP layers provide a best-effort connectionless service for the transfer of preconfigured frames or datagrams, respectively. To support interworking between a station with a cell interface and a station connected to a legacy LAN, two alternatives must be supported: one for use with a bridged LAN and the other for use with a router-based LAN. With the first, the interface between the ATM LAN and the legacy LAN is a bridge and with the second the interface is a subnet router. Two different protocol architectures are used, each of which provides seamless interworking in the appropriate environment. We shall consider each separately.

LAN emulation

The method used when bridges form the interface to the legacy LAN has been developed by a group of companies that are all involved in the manufacture of LAN networking equipment. This group is called the **ATM Forum**. The various networking components used with this method are shown in Figure 10.13(a).

The aim of the various networking components is to emulate the broadcast mode of operation of a legacy LAN over a connection-oriented ATM LAN. The term used to describe this method is known as **LAN emulation (LE)**. The three components used are an **LE configuration server (LECS)**, an **LE server (LES)**, and a **broadcast and unknown-address server (BUS)**. Although each is shown as a separate entity in Figure 10.13, for a small LAN, they may all be implemented in a single computer. In this case, the message/frames relating to each component are identified by the type of virtual channel connection (VCC) on which they arrive.

(a)

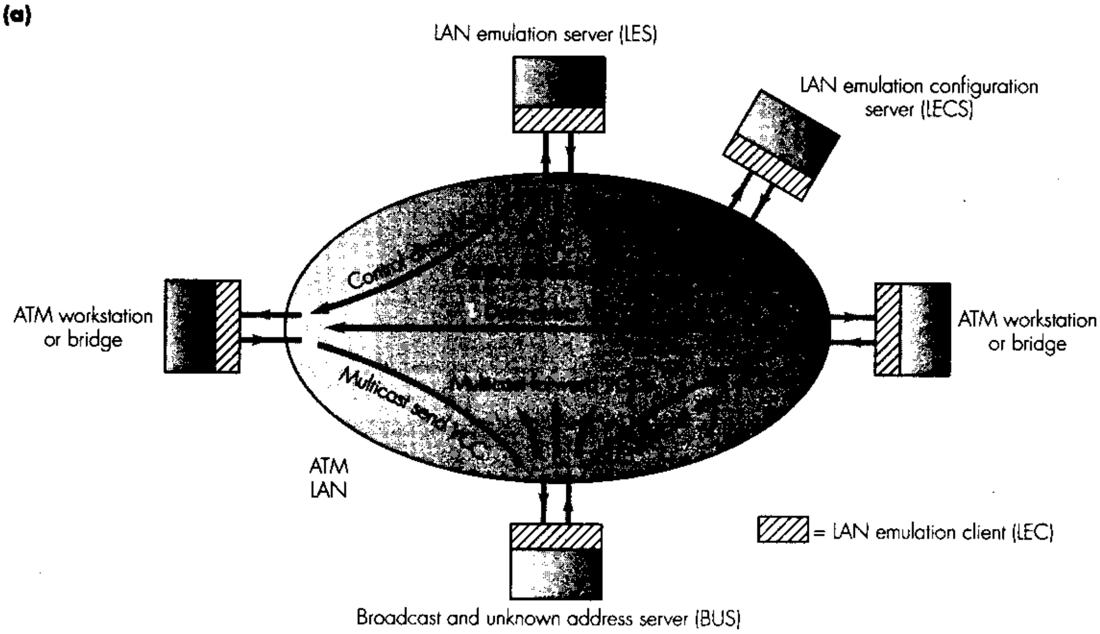


Figure 10.13 LAN emulation: (a) terminology and networking components; (b) unicast protocol architecture; (c) multicast protocol architecture.

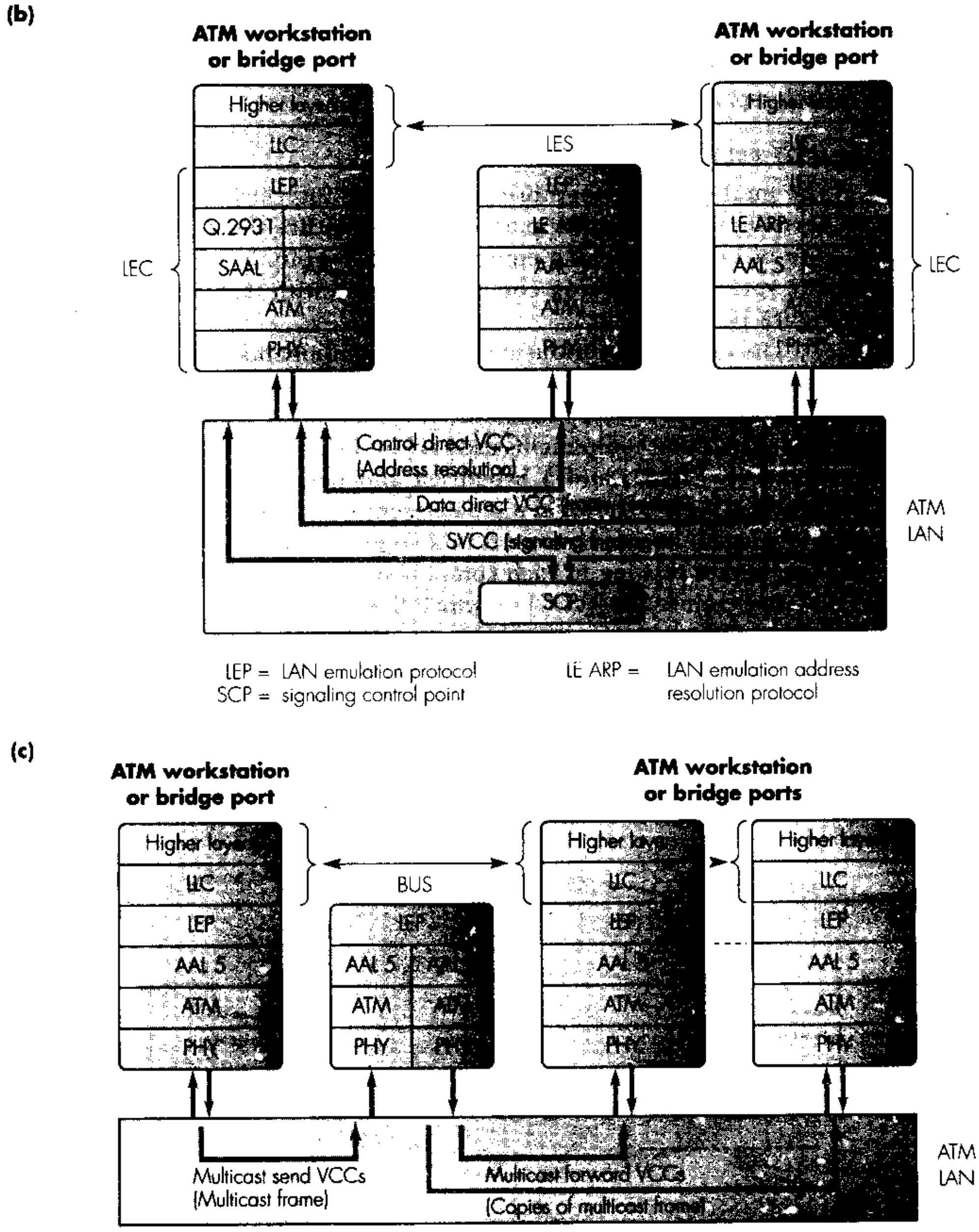


Figure 10.13 Continued

In a large installation there may be multiple emulated LANs each of which has its own LES and BUS. The LECS is used by all stations that are connected to the total ATM LAN to determine the ATM addresses of the LES and BUS. There is a reserved VCC to connect all stations to the LECS for this purpose. The LES provides an address resolution service to convert from 48-bit MAC addresses into 20-byte ATM addresses. The BUS is responsible, firstly, for supporting multicasting/broadcasting, and secondly, for relaying frames to stations whose MAC address is unknown by the LES.

All stations connected to the ATM LAN – ATM-enabled stations, bridges, and so on – have an **LE client (LEC)** subsystem associated with them. This comprises hardware and software that provides a similar service to the MAC chipset (and associated software) in a station that is connected to a legacy LAN. When a station is powered up, it goes through an initialization procedure at the end of which it has initialized certain operational parameters and its own set of addresses – as described below, a station has more than one address. As we can see in Figure 10.13(a), a separate pair of VCCs is used to connect each LEC to the LECS, the LES, and the BUS. Once initialized, the LEC first obtains the pair of VCCs of the LES and BUS from the LECS and then proceeds to register itself with both the LES and BUS. At the end of the initialization phase, both the LES and BUS have the set of addresses of all active stations that are connected to the ATM LAN. In the case of a bridge, the LEC can either register the MAC addresses of all stations that are connected to its legacy LAN port or, more usually, just the address of the port through which it is connected to the ATM LAN. In the first case the bridge operates in the **nonproxy mode** and in the second, the **proxy mode**.

We can see in Figure 10.13(b) that the protocol stack within each LEC contains a **LAN emulation protocol (LEP)** layer – immediately below the LLC sublayer – which communicates with a similar layer in the LES. In order to make the underlying network transparent to the LLC sublayer, the service provided by the LEP layer is a connectionless service similar to that offered by the MAC sublayer. The two user service primitives are LE_UNITDATA.request and LE_UNITDATA.indication.

To emulate a broadcast LAN, both primitives have a source and destination MAC address as parameters. This means that each workstation and bridge connected to the ATM LAN, in addition to a 20-byte ATM address, has a 48-bit MAC address associated with it. It also has a 2-byte **LEC identifier (LECID)** which is used to identify uniquely the ATM workstation or bridge port among those currently attached to the LES. We shall look at the function of the LECID later.

The VCC that (logically) connects an LEC to the LES is called the **control direct VCC**. On receipt of an LE service request primitive, the LEP in the source LEC reads the source and destination MAC addresses and passes these to the **LE address resolution protocol (LE ARP)**. The latter forms an address resolution request message – containing the two MAC addresses and the ATM address of the LEC – and sends this to the LE ARP in the LES over the control direct VCC. Assuming the LE ARP in the LES knows the ATM address of the destination LEC, it returns this in a reply message to the requesting LE ARP – the latter initiates the setting up of a **data direct VCC** between itself and the LE ARP in the destination LEC. This is done using either the SCP and the signaling protocol set described earlier for connection-oriented calls or, with some ATM switch architectures, directly by the two LE ARPs involved. The connection is cleared by the LE ARP when no further frames are received for the same destination LEC within a defined timeout interval.

If the LE ARP in the LES does not have the ATM address of the destination LEC – for example, if the destination MAC address relates to a station connected to the legacy LAN port of a bridge – the LES sends a copy of the LE ARP request message to all registered LECs using the set of **control distribute VCCs**. The LE ARP which has knowledge of the destination MAC address then replies with the corresponding ATM address – or its own ATM address if it is a bridge – using the control direct VCC. Once the source LE ARP has obtained the ATM address of the destination, it sets up a data direct VCC as before.

The flow of data frames, each of which has the source and destination MAC addresses at its head, can then start. Since bridges are involved, these can be either Ethernet, 802.3 or 802.5 frames with the frame type defined in the header of the frame. No FCS field is required since, in an ATM LAN, this is performed by the AAL 5 sublayer. Because of the use of the intermediate address resolution phase, the LE connectionless service is said to be provided *indirectly*.

The foregoing relates to unicasting, that is, there is a single destination station for each submitted MAC frame. To implement multicasting, a copy of the created MAC frame must be forwarded to all stations that belong to the same multicast group. Clearly, using the method just described, a multiplicity of switched VCs would be required between each member of the group and all other members. To avoid this requirement, the BUS is used and the associated protocol architecture is as shown in Figure 10.13(c).

An additional pair of VCCs is used between the LEC in each station and the LEC in the BUS. In the station-to-BUS direction, the VCC is known as the **multicast send VCC** and in the reverse direction, the **multicast forward VCC**. On receipt of a service primitive with a multicast destination address, the LEP creates a frame as before but with its own 2-byte LECID at the head. It sends this directly to the LEP in the BUS over the multicast send VCC. On receipt of the frame, the LEP broadcasts a copy of the frame to all stations using the set of multicast forward VCCs. The LEP in each station first determines from the LECID at the head of the frame whether it originated the frame. If it did, then the LEP simply discards the frame. This is known as **echo suppression**. If the LECID does not match that of the station, the receiving LEP determines from the multicast address at the head of the frame whether this station is part of the multicast group. If it is, the frame is passed up to the LLC layer, if not, the frame is discarded.

The unknown address service associated with the BUS enables an LEP to send a limited number of frames to their intended destination during the period a data direct VCC is being set up. The procedure followed is the same as that for a multicast frame and, since multiple copies of each frame are sent – each over a separate multicast forward VCC – a limit is set on the number of such frames the LEP can send. Any further frames received must then be retained – cached – until the data direct VCC is in place.

Classical IP over ATM

The method used when subnet routers form the interface to the legacy LAN has been developed by the Internet Engineering Task Force (IETF). The basic method is known as **classical IP over ATM (IPOA)**.

Recall from Chapter 9 that, using the IP, before a station/host can exchange datagrams with another station, it must first know the IP/MAC address-pair of its local subnet router. Also, the local subnet router must know the address-pair of all the stations that are connected to the LAN segments to which it is attached. These are acquired using the ARP protocol and exploiting the broadcast nature of legacy LANs. Once these have been acquired, all datagrams/packets are exchanged between two stations via the router and it is this that relays the packets to their intended destination using the MAC address corresponding to the IP destination address in the datagram header.

To emulate the same operation with a non-broadcast ATM LAN, a permanent VCC is set up between all stations and router ports that are connected to the ATM LAN and a central node known as the connectionless server (CLS). The CLS provides a dual function of address resolution and a datagram/packet relaying service. Since with this method all packets are relayed directly by the CLS, the connectionless service is said to be provided *directly*. The protocol architecture used with a CLS is shown in Figure 10.14.

The ARP in each station/router port first registers its own address-pair – IP and ATM addresses – with the ARP in the CLP using the corresponding permanent VCC. In a similar way, the ARP in the CLS can acquire the address-pair of a station or router port that is connected to it by a VCC. In this way, the ARP in the CLS builds up a routing table containing the address-pair of all stations and router ports that are connected to it. Whenever a station IP has a datagram to send, it simply sends the datagram to the IP layer in the CLS over the VCC that connects the station to the CLS. On receipt of the datagram, the IP in the CLS first reads the destination IP address from the head of the datagram and uses this to obtain the corresponding ATM address from its routing table. The IP then forwards the datagram/packet to the destination over the corresponding VCC.

Multicasting with a CLS is carried out in a similar way to that described with an LES except with a CLS a separate broadcast server is not required. Instead, on receipt of a datagram with a multicast address, the IP layer in the

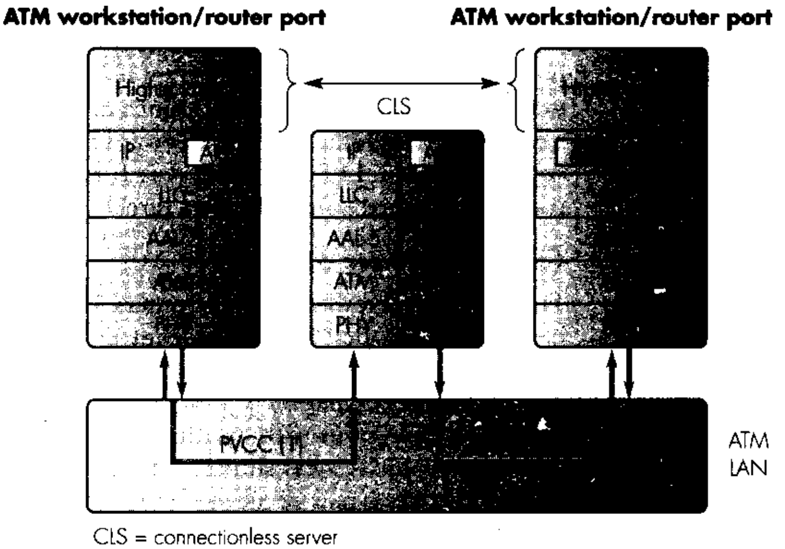


Figure 10.14 Protocol architecture to support classical IP over an ATM LAN.

CLS simply sends a copy of the datagram to all other stations using their corresponding VCC. Each recipient IP then determines from the multicast address at the head of the datagram whether it is a member of the multicast group. If it is, the datagram is passed to the higher protocol layers, if not, the datagram is discarded.

When a CLS is being used, there are two alternative relaying modes. With the first, the complete datagram is reassembled by the AAL before it is processed by the IP layer. Clearly, this introduces delays and requires significant buffer storage for its implementation. In order to reduce these overheads, we may use a second mode of working known as **streaming** or **pipelining**. In practice, the destination address required for routing is carried in the first ATM segment/cell and hence in this mode, the first (beginning of message) segment is passed directly to the IP layer for processing as soon as it is received by the AAL protocol. The new VC is then determined and this and the remaining segments making up the datagram are relayed directly without reassembly. The AAL with this method is AAL 3/4 since the MID field in the header of each cell is required for identification purposes during the reassembly process.

10.6 ATM MANs

Metropolitan area networks (MANs) have been installed by a number of network providers to meet the demand from customers for a switched high-speed LAN interconnection facility. Prior to the introduction of MANs, the only way of interconnecting a set of LANs that belong to different enterprises was by means of high bit rate leased lines. The disadvantage of this approach is the large number of leased lines that are required and, for each additional site, a new set of lines must be put in place. Initially, the LANs were all distributed around a large town or city but later the service was extended to cover multiple towns and cities.

All MANs operate using the asynchronous transfer mode and hence one approach to implementing a MAN is to use a set of interconnected ATM switches similar to the ATM backbone network we showed earlier in Figure 10.11. An alternative (and often less costly) solution is to use a high bit rate shared transmission medium. This is the approach taken in **distributed queue dual bus (DQDB)** MANs and, since this has been adopted by a number of providers, we shall limit our discussion of MANs to DQDB.

DQDB is an international standard that is defined in **IEEE 802.6**. The standard relates to a single subnetwork in the context of a larger network; typically, the larger network comprises a number of interconnected DQDB subnetworks. Each subnetwork is made up of **dual contradirectional buses** – that is, two unidirectional buses running in opposite directions – to which a distributed community of **access nodes** – also known as **customer network interface units** – are attached. The buses may be in an **open bus topology** or a **looped bus topology**, in which case the two ends are separate but physically colocated. As we shall see later, this can be used to provide better fault tolerance. Three example applications are shown in Figure 10.15: part (a) shows a single subnetwork MAN; part (b) a typical split-site private network; and part (c) a large wide area public network comprising multiple interconnected MANs.

As part (a) shows, a single DQDB MAN subnetwork can cover an area in excess of 30 miles (50 km). In this example, the access nodes may be located at the telephony switching offices in that area. The use of 34/45/140/155 Mbps circuits provides **seamless interconnection** for many LANs distributed around the city; that is, two users at different sites communicating with one another will be unaware of the fact that an intermediate network is involved.

Part (b) illustrates the use of two DQDB subnetworks in a private network application. In this instance, all the networking equipment located at both sites are privately owned and run and the high bit rate circuit interconnecting both sites leased from a public carrier. One site is assumed to have a looped bus topology and the other an open bus. This example also shows how a DQDB network can be used for the interconnection of two private telephone exchanges (PBXs) which requires a constant bit rate channel.

Part (c) illustrates how several multisubnetwork MANs can be interconnected together to form a larger network that spans several cities. As we can

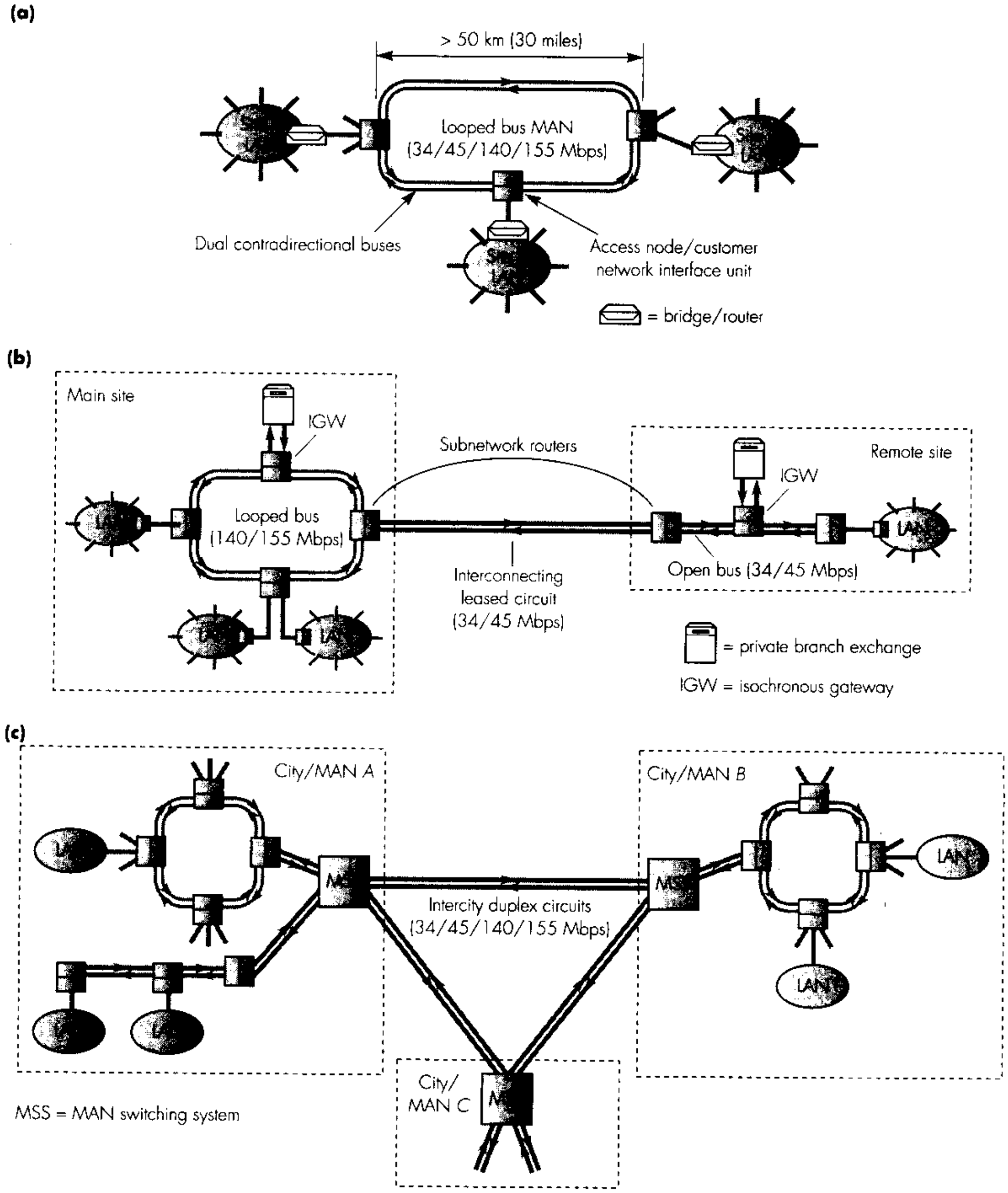


Figure 10.15 DQDB/MAN network architectures: (a) single-site MAN; (b) dual-site private network; (c) wide area multiple MAN network.

see, in this case each DQDB subnetwork in a city is connected to a **MAN switching system** (**MSS**). The MSS, on receipt of each LAN frame, either relays it directly to the required subnetwork – if it is connected to it – or it forwards the frame to the required MSS over the appropriate intercity duplex circuit. Such a network provides a **switched multimegabit data service** (**SMDS**) similar in principle to that provided by a frame relay network.

Normally, a LAN is connected to its nearest access node by either a remote bridge or a router. All frames are transported across each DQDB subnetwork in the form of fixed-sized units known as segments. At the interface to each subnetwork, the frames are first fragmented into segments and then reassembled back into their original form at the destination. Because the format of LAN frames differs from one type of LAN to another, prior to the transfer of a frame, a standard header and trailer are added to it. With public networks, the header contains two new source and destination addresses which are the networkwide addresses of the corresponding customer access/interface units. As we shall see in Section 10.6.7, these are hierarchical addresses which identify the MSS, the subnetwork, and the customer access unit. All segments are transferred over each subnetwork using both buses which, because these pass data in opposite directions, ensures that a copy of each segment transmitted is received by all nodes on the subnetwork. Between subnetworks, routing is carried out on reassembled frames using the networkwide destination address at the head of each frame.

10.6.1 Subnetwork architectures

A schematic diagram of an open dual-bus subnetwork is shown in Figure 10.16(a). At the head of each bus is a **slot generator** which generates a contiguous stream of **slots** each capable of transferring a standard 53-octet cell. Access nodes are attached to both buses through read and write connections, with the read operation performed ahead of the write operation. The physical layer in each access node reads the contents of each slot without modifying its contents, and only if new data is to be sent by an access node does it overwrite the existing contents. We can conclude, therefore, that an access node only copies (reads) data from a bus, it does not remove it. Also, it only changes the contents of a slot when it is permitted to do so by the access protocol. Hence failure of the access unit within a node does not have any effect on the contents of the slots that pass by on the two buses providing the failure does not cause the node continuously to write.

A looped bus architecture is shown in Figure 10.16(b) and, as we can see, in this configuration both slot generators are colocated. Alternatively, the same slot generator can be used for both buses. In both cases, the two buses are still independent and are not connected together as in a ring network. Also, in addition to generating slots, the **head of bus** is responsible for passing management information to the access nodes. Typically, management information is generated by a separate network management station and

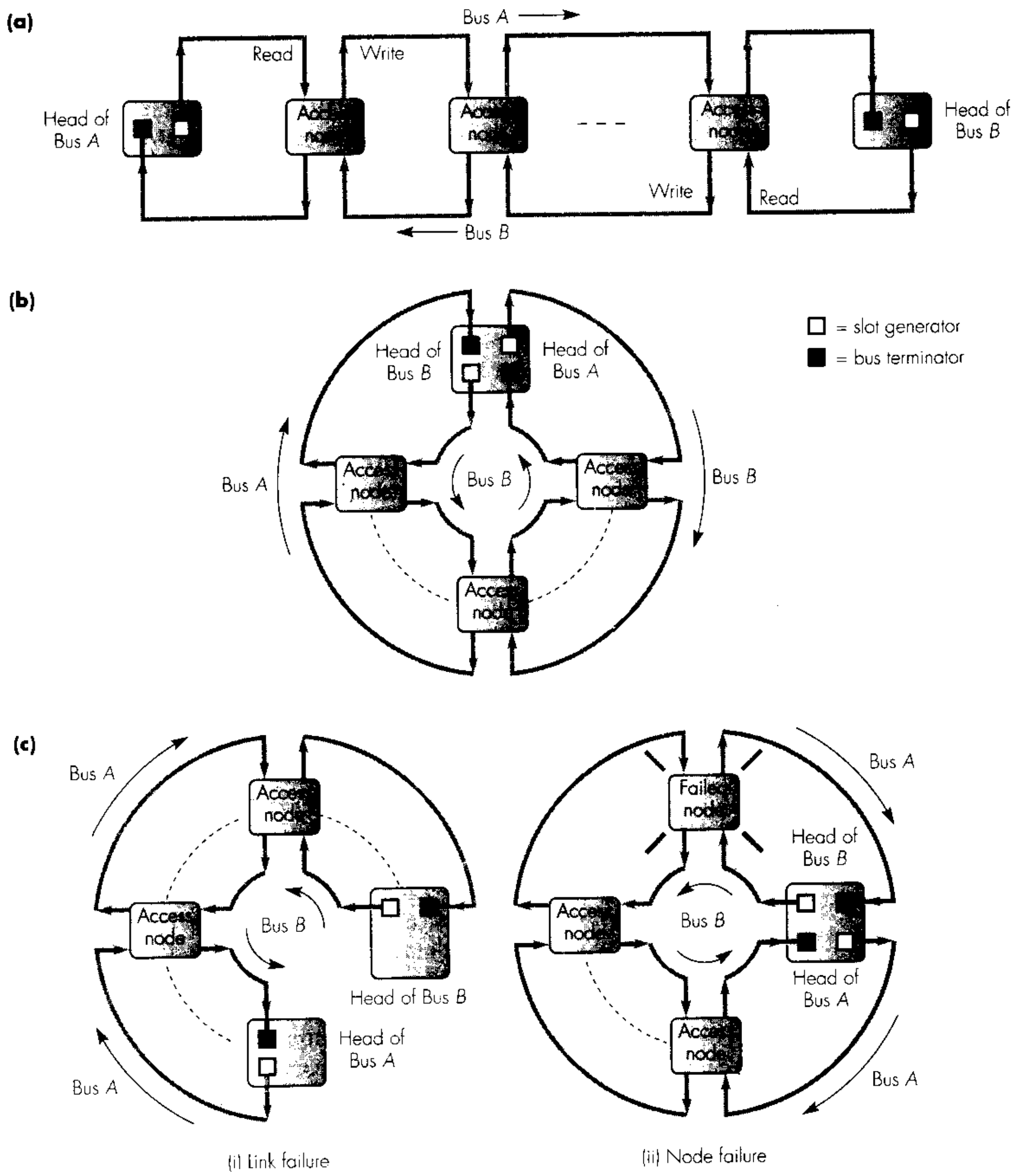


Figure 10.16 DQDB architectures: (a) open bus; (b) looped bus; (c) example reconfigured looped bus networks.

relates to either the allocation of isochronous bandwidth or the operational integrity of the subnetwork.

With a looped bus topology reconfiguration can be supported in the event of a link or a node failure by replicating the head of bus functions so that more than one node can assume the role of head of bus. Two examples of reconfigured loops are shown in Figure 10.16(c); the first illustrates the reconfigured loop after a link failure and the second the reconfigured loop after the failure of an access node. The reconfiguration operation is carried out under the control of a remote network management station.

10.6.2 Protocol architecture

Figure 10.17(a) identifies the components of the protocol architecture defined in IEEE 802.6. As with the other IEEE 802 standards, this defines the operation of the MAC sublayer – known as the DQDB layer in the standard – and the physical layer.

In addition to the normal connectionless (best-effort) data service provided for the interconnection of the frame-based LAN types, the DQDB MAC sublayer provides two additional services: a connection-oriented (reliable) data service and an isochronous service. Typically, the latter is used for the interconnection of two private (telephone) branch exchanges. For such applications, the multiplexed voice samples within the duplex link connecting the exchanges to their access nodes must be transferred across the bus at the same rate as the access link, hence the term "isochronous".

MAC sublayer

As Figure 10.17(a) shows, the MAC sublayer comprises four main functions: convergence, bus arbitration, common, and layer management. As we indicated earlier, all information is transferred across the dual buses in the form of fixed-sized segments. The convergence functions are required to convert all incoming source information into segments before its transfer and back into its original form before its delivery. For example, the MAC convergence function fragments the data frames submitted by a bridge or router into multiple segments ready for their transfer and, on receipt, reassembles them back into frames prior to their delivery.

There are two alternative control modes for gaining access to the slots on the two buses: **queued arbitrated** (**QA**) and **prearbitrated** (**PA**). When isochronous services are supported, in order to provide a constant bit rate service, the required number of slots are preallocated and marked by the head-of-bus node. The prearbitrated function in each node providing the service is responsible for identifying the reserved slots relating to it – from an identifier at the head of each slot – and then either initiating the transmission of the isochronous data within these or receiving the data contained within them. All the remaining bus slots are used for the transfer of LAN information frames. Access to these slots is controlled by a distributed queuing algorithm in the queued arbitrated function block.

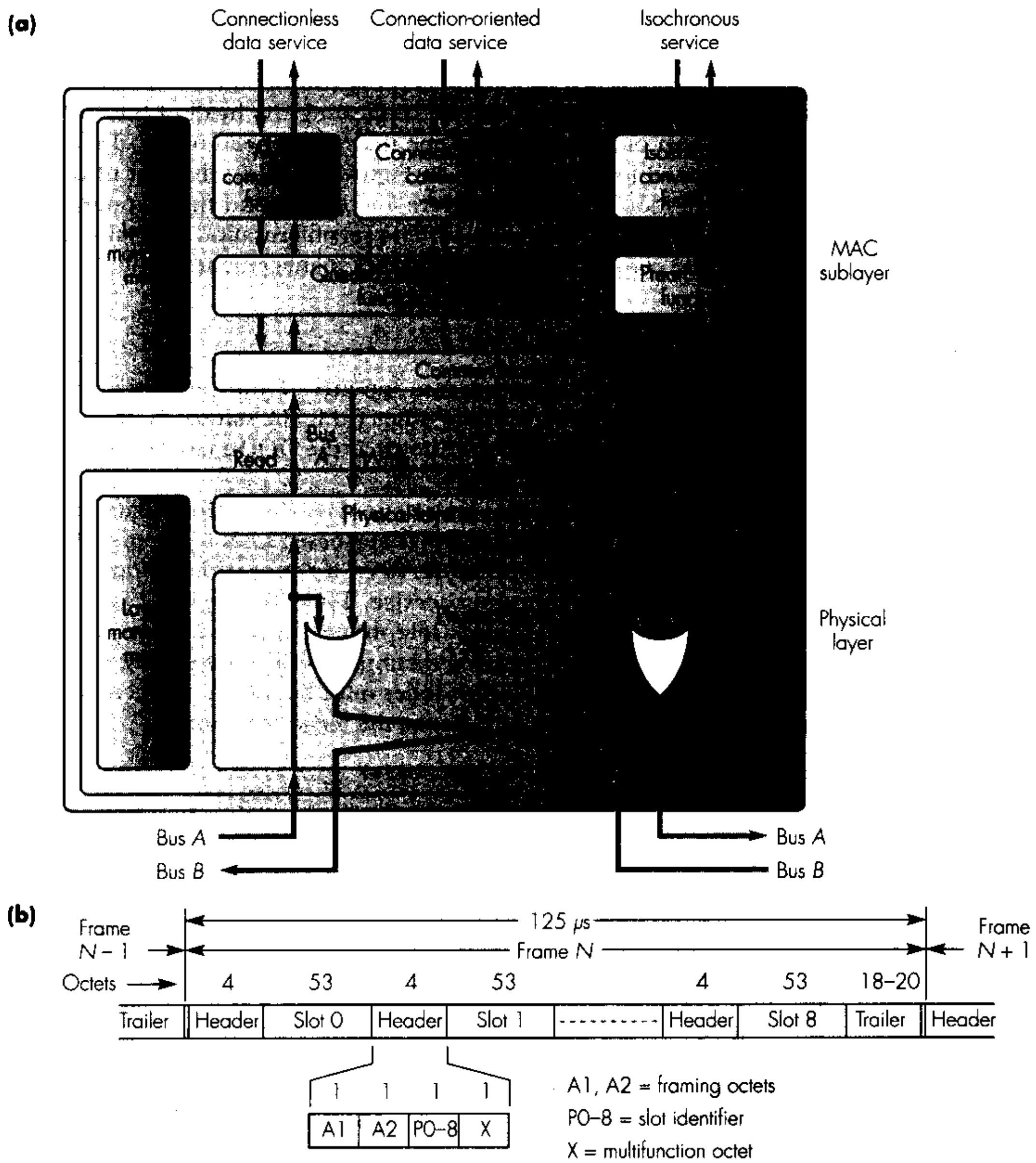


Figure 10.17 DQDB protocol architecture: (a) layer functions; (b) example physical layer convergence function.

We can see from Figure 10.17(a) that the common function block forms the interface between the physical layer and the two bus arbitration function blocks. The physical layer interface takes the form of single octet transfers – read and write – to and from both buses. The main role of the common function block is to relay octets between the physical layer and the appropriate bus arbitration function. This involves detecting the start and end of each slot and examining selected octets within the slot/cell header according to its defined format. We shall consider this in more detail in Section 10.6.6. In addition, since more than one node can act as the head of bus (to enhance reliability) the nodes which can perform this function take part in the reconfiguration operation of the bus/loop when faults are detected. Also, when selected as the head of bus, the node generates contiguous streams of slots. Both these operations form part of the common functions.

Physical layer

The physical layer provides a standard interface to the MAC sublayer. However, as Figure 10.15 showed, the physical layer can be implemented using a range of different transmission media. For example, with public networks this can be 34, 45, 140, or 155 Mbps. As we saw in Sections 7.2.3 and 7.2.4, such circuits utilize different framing formats and so a **physical layer conversion function** – which establishes a contiguous sequence of slots over the selected physical transmission medium – must be provided. An example of how this is achieved is shown in Figure 10.17(b).

The example relates to the use of a 34.368 Mbps E3 digital circuit. As the numbering indicates, this is at the third level in the PHD multiplexing hierarchy and is used to transmit four 8.448 Mbps streams. As we showed in Figure 7.17, a number of bits/octets are used for framing and other purposes and hence a lower bit rate is available for the transmission of DQDB slots. When using such circuits for each bus, the head of bus establishes the frame structure shown in Figure 10.17(b) using the available transmission bandwidth. This has a duration of 125 μs which ensures that the frame can be used to support various constant bit rate voice services as well as data transfer. As we can see, preceding each 53-octet slot is a 4-octet header. The first two octets of the header enable the physical layer convergence function in each node to synchronize to the start of each new frame. The third octet identifies the slots within each frame. The fourth octet is used for a number of functions including the transfer of management information from the head of bus to the two layer management entities in the attached nodes. The physical layer convergence function reads and interprets the octets in the header field but only passes the 53-octet cell in each slot to the MAC sublayer. Because of the fixed 125 μs interval, each frame does not contain an exact multiple of 57 octets. The unused octets in this field arise due to the plesiochronous (nearly synchronous) mode of operation of such circuits as described in Section 7.2.4. With higher bit rate circuits, the 125 μs interval is maintained and hence each frame contains proportionately more (DQDB) slots.

Finally, as with all the protocol layers, the layer management entities associated with the physical layer and MAC sublayer are provided for a remote management entity to control and monitor the operation of both these layers. For the MAC sublayer this involves receiving and acting upon information relating to the slots, for example, the list of identifiers that are carried at the head of each prearbitrated slot and the message identifiers that are at the head of queued arbitrated slots, the use of which we shall describe in Section 10.6.6. It also involves the setting of operational parameters such as timers and the activation of the configuration control procedure. All information relating to such operations is carried in the header that precedes each slot.

10.6.3 Queued arbitrated access protocol

Access to the slots that are available for the transfer of asynchronous data is based on a distributed queuing algorithm. This is known as **queued-packet distributed-switch** (**QPSX**) and the principles of the method are shown in Figure 10.18.

When interpreting the figure, it should be remembered that for an access control unit (ACU) to broadcast a segment/cell to all other ACUs, it must send a copy of the segment on both buses. Also, each bus operates independently and the QPSX algorithm ensures each ACU gains access to both buses in a fair way.

As part (a) shows, each slot contains two bits at its head that are used in the distributed queuing algorithm: the **busy** or **B-bit** and the **request** or **R-bit**. Also, as part(b) shows, access to the slots on each bus is controlled by a separate **request counter**. To ensure access to empty slots on each bus is fair, requests for slots on one bus are made using the R-bit in slots on the other bus. Then, for each counter, whenever a slot passes with the R-bit set, the contents of the corresponding counter are incremented by one. Similarly, whenever an empty slot is repeated at the interface of the opposite bus, the counter is decremented by one. At any point in time, therefore, the request counter for the corresponding bus contains the number of outstanding requests for slots on that bus from the access nodes that are downstream of the access node. Note that since there can be a different number of ACUs on each side of the access mode, each bus operates independently of the other.

Figure 10.18(c) shows that, to transmit a cell (containing a segment of data) on bus A, the sending access node transfers the current contents of the request counter for bus A to a second counter known as the **countdown counter**. The node then resets the contents of the request counter to zero and sets the R-bit in the first slot received on the opposite bus with the R-bit reset. This has the effect of placing the segment into the distributed queue for bus A. The same procedure is followed concurrently with the set of counters associated with bus B to place a copy of the segment in the distributed queue for bus B.

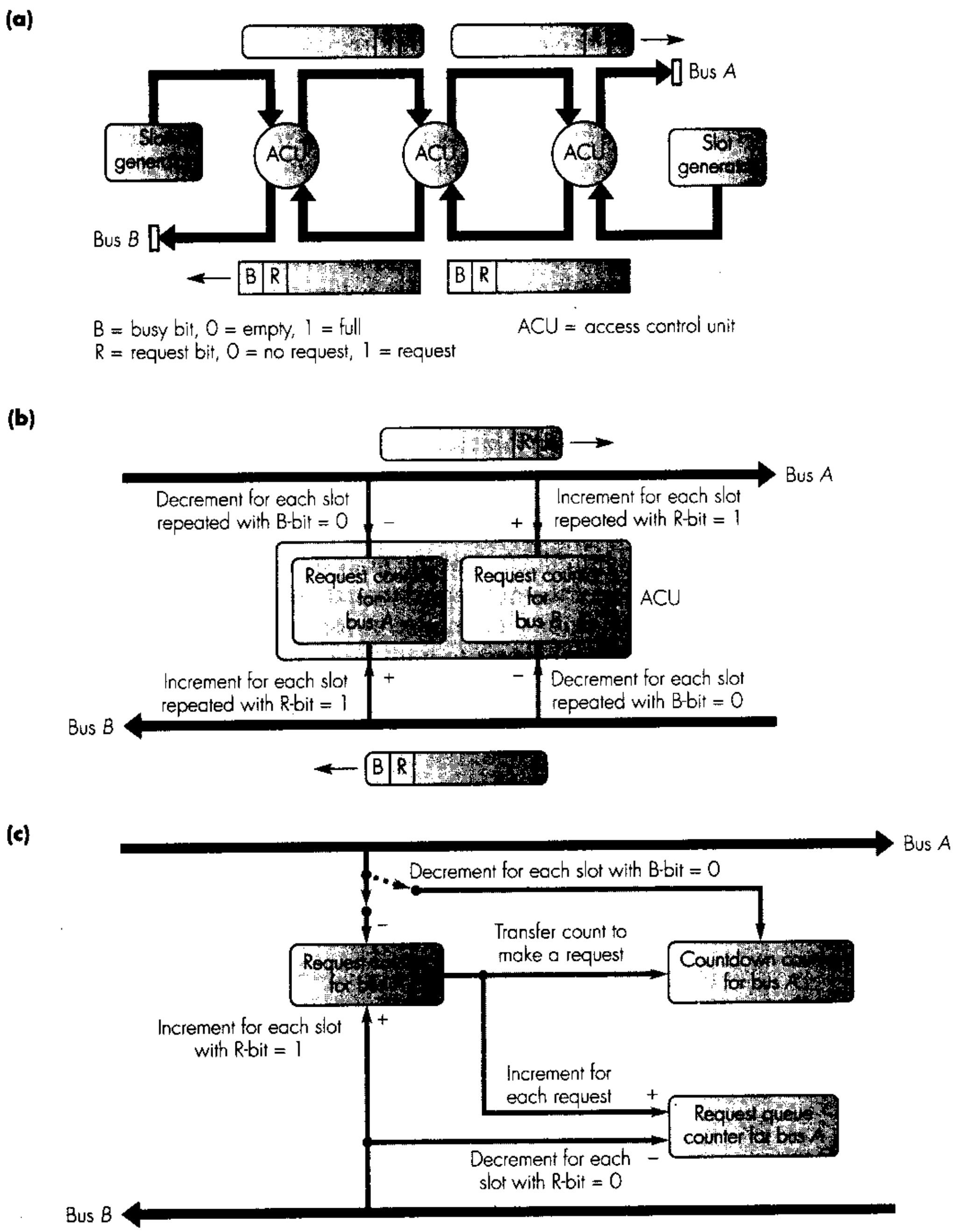


Figure 10.18 DQDB access control principles: (a) request/busy bits; (b) request counters; (c) queuing mechanism.

While a segment is queued for a bus, any slots that are repeated with their R-bit set cause the corresponding request counter to be incremented as before. However, slots which are repeated at the opposite bus interface with their B-bit reset (zero), cause only the countdown counter for that bus to be decremented. The queued segment is then transmitted on the bus when the corresponding countdown counter is zero and an empty slot is received.

In practice, since both buses operate independently, it is possible for a string of slots to be received on the opposite bus all with their R-bits set. This means that it may not be possible to set the R-bit in a slot – in response to a new request – until more than one segment has been queued and transmitted on the opposite bus. To allow for this, a third counter known as the (local) **request-queue counter** keeps a record of outstanding requests. Whenever a new request is made, the counter is incremented and, all the time the counter contents are greater than zero, whenever a slot is received with its R-bit reset, the R-bit is set and the counter is decremented.

10.6.4 Bandwidth balancing

Following the introduction of the draft standard of DQDB, detailed performance analyses of the queued-arbitrated protocol demonstrated that, under heavy load conditions, the nodes nearest to the head of each bus start to obtain preferential access to both buses relative to the nodes nearer the

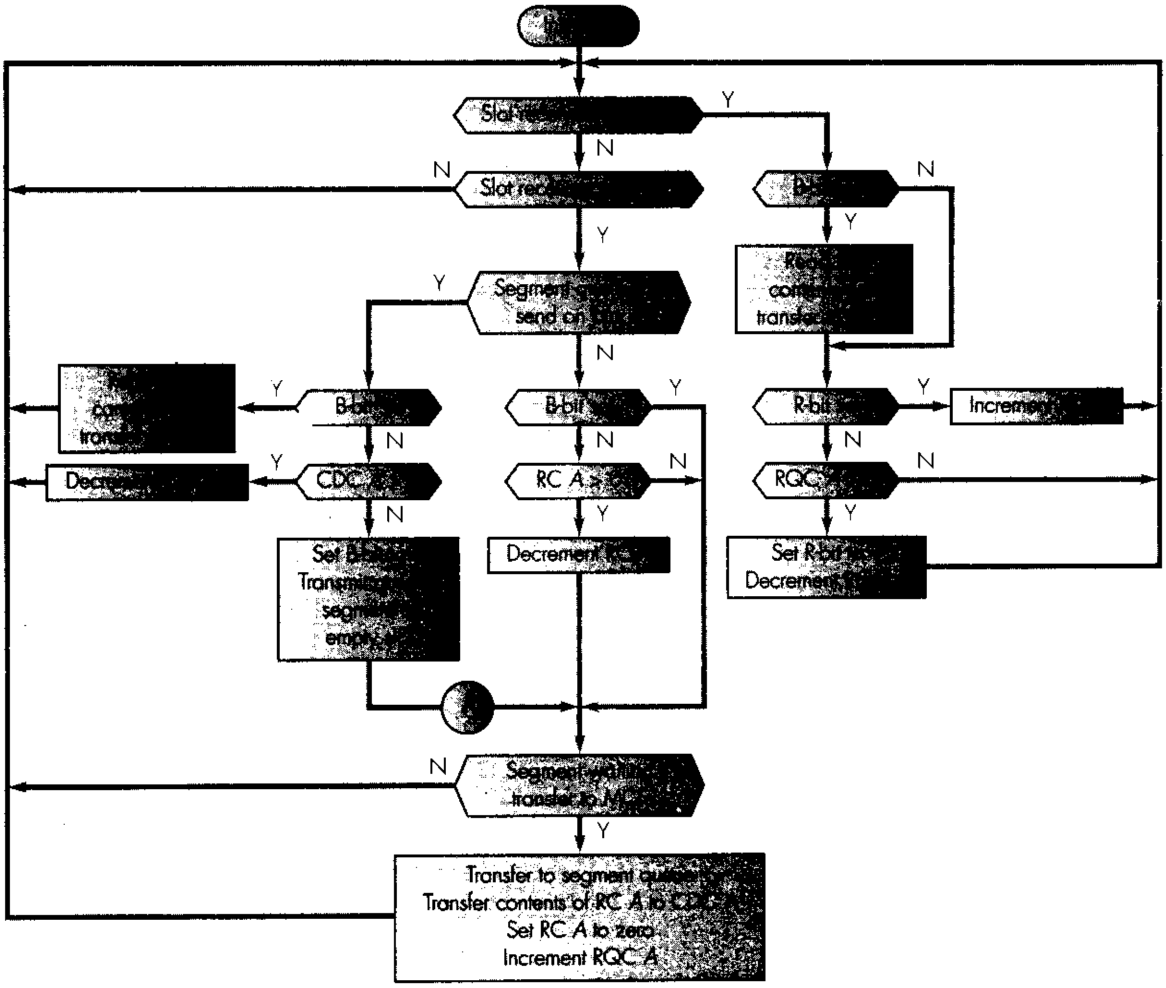
Example 10.2

Derive a flowchart showing the steps taken by the queued arbitrated function to effect the transmission of a set of queued segments [illegible] by the MAC convergence function on a single bus of a [illegible] DQDB subnetwork.

Answer:

A flowchart showing the steps [illegible] the transmission of [illegible] on bus A is given in Figure [illegible]. Consider the following points [illegible] interpreting the figure:

- On receipt of a full slot on either bus, the queued arbitrated function simply passes the contents of the slot payload directly to the MAC convergence function. This function determines whether the segment is intended for this node.
- Only a single segment can be queued for transmission by the queued arbitrated function at one time. Hence only after this function has transmitted a segment does it return to the output queue of the MAC convergence function to determine whether another segment is awaiting transmission.



RC A = request counter for bus A RQC A = request queue counter for bus A

CDC A = countdown counter for bus A MCF = MAC convergence function

= point of introduction of bandwidth balancing procedure (see Section 10.6.4)

Figure 10.19 Flowchart of the algorithm used to control the transmission and reception of segments on busA of a dual bus DQDB subnetwork.

centre of the bus. We can best understand the reason for the unfairness by remembering that the node at the head of each bus has first call on the use of the request bit in the slots that pass on one of the buses. Also, although the same node is the last to make requests on the other bus, the related empty slots pass this node first. Under heavy load conditions when the demand for slots starts to exceed supply, this has the effect shown in graphical form in Figure 10.20(a). The access delay variation relates to a heavily loaded subnetwork and is the same for both buses. As we can see, the unfairness increases as the network size increases and/or the bit rate increases.

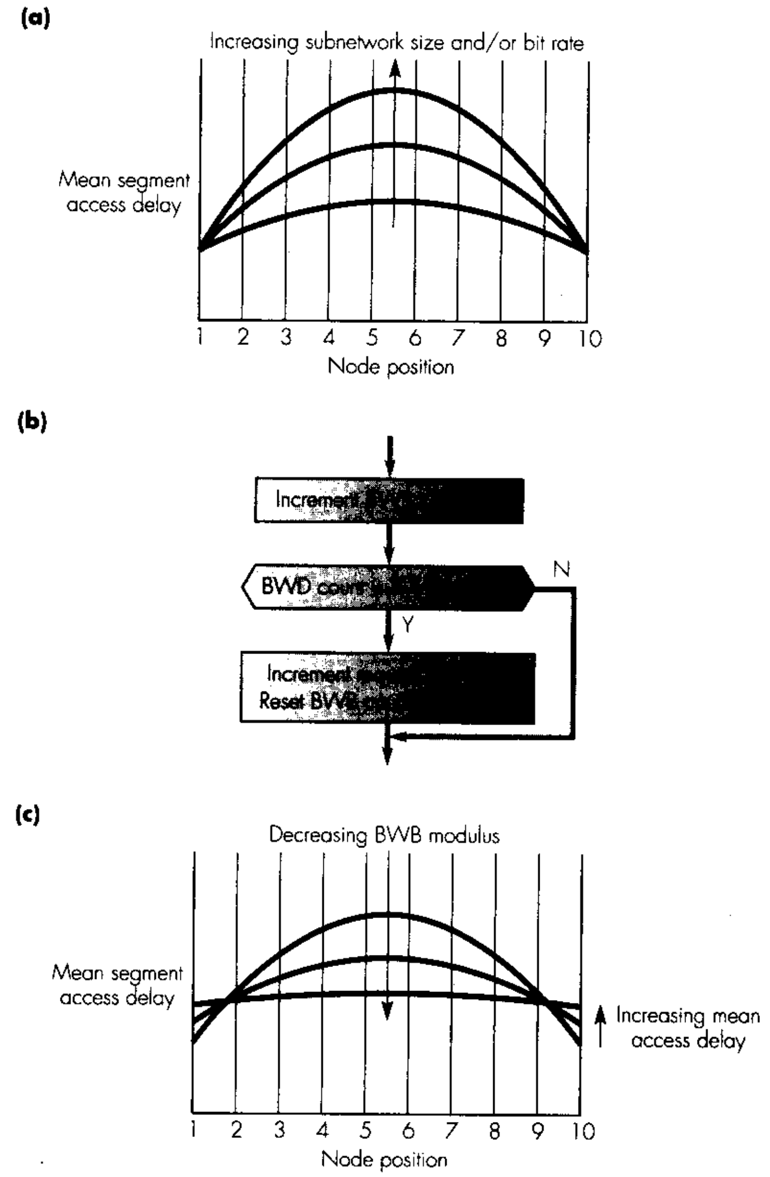


Figure 10.20 Bandwidth balancing: (a) unfairness effect; (b) remedial actions; (c) effect on mean access delay.

To overcome this effect, a modification to the basic access control algorithm known as the **bandwidth balancing mechanism** was introduced. To implement the scheme, a fourth counter called the **bandwidth balancing (BWB) counter** is introduced for each bus. Whenever a segment is transmitted on a bus, the BWB counter for that bus is incremented. Then, whenever the counter reaches a preset limit, the node allows an additional free slot to pass by on this bus by incrementing the corresponding request counter. The BWB counter is then reset to zero and the process repeats. The preset limit is called the **bandwidth balancing modulus**. This operation means that each node, after transmitting a block of segments equal to the BWB modulus, must allow an additional slot to pass by on the related bus. This can be used by the first node lower down the bus which has a queued segment to transmit and a zero countdown counter. The additional processing steps required are shown in Figure 10.20(b) and these are introduced at point *A* in the flowchart shown in Figure 10.19.

The bandwidth balancing mechanism obtains the necessary effect by reducing the utilization of each bus; the smaller the BWB modulus, the larger the bandwidth loss. This is shown in graphical form in Figure 10.20(c), which illustrates the impact of reducing the modulus for a single network type – size and bit rate. As we can see, as the modulus decreases, the unfairness decreases but at the expense of an increase in the mean access delay. In practice, a compromise is made and a value of 8 is normally used. In the worst case, this results in a loss of utilization of $1/(8 + 1)$ or 11.1%.

10.6.5 Prioritized distributed queuing

Although not always implemented, the basic queued arbitrated access control scheme just described can be extended to support the transmission of prioritized cells/segments. There can be three priority classes and each has a separate set of counters – request counter, countdown counter, and request-queue counter – for each bus. The three priority classes are 0, 1, and 2, with 2 having the highest priority. Cells/segments relating to data-only frames are always allocated class 0, and class 2 is reserved for the transfer of cells relating to network management messages. Class 1 is intended for the transfer of cells containing information that is sensitive to delay or delay variation. Note that bandwidth balancing is not used in this mode.

An example application that may use class 1 cells is for the transfer of compressed video information. As we saw in Section 4.3.1, although the information is generated at a constant rate – determined by the video frame refresh rate – the amount of information associated with each compressed frame varies and depends on the level of movement that has taken place relative to the previous frame. We can deduce that if the isochronous service is used then the (preallocated) bandwidth needs to be that required to transfer a completely new frame at the refresh rate. However, by using the queued arbitrated access method, the amount of information transferred can vary from one frame to the next. By assigning a higher priority to such segments, the priority control method endeavors to ensure that these are transferred before slots containing other types of LAN traffic.

The general arrangement for controlling access to bus *A* is shown in Figure 10.21. For clarity only the request and countdown counters are shown. A similar arrangement is used for access to bus *B*. As we can see, there is a separate R-bit for each priority class at the head of each slot/cell. These are shown as R0, R1, and R2, where R2 is the highest priority. Assume initially there are no segments waiting transmission on bus *A* from this node – part (a). The operation is as follows:

- When an empty slot (B = 0) is repeated at the interface with bus *A*, the access control unit for this bus decrements all three request counters by one.
- When a slot is repeated at the interface with bus *B* – with a priority of 1 for example – the access unit for bus *B* increments only request counters 1 and 0 – RC 1 and RC 0 – and leaves the higher priority counter – RC 2 – unchanged. This means that lower-priority requests do not delay the transmission of higher-priority segments.

Now assume that a segment becomes ready to transmit on bus *A* of priority 1 – part (b). The steps are as follows:

- The current contents of RC 1 are transferred to CDC 1 (and RC 1 is reset to zero) when a slot is repeated at the interface to bus *B* with the corresponding request bit reset to zero.
- When an empty slot is repeated at the interface with bus *A*, request counters RC 0 and RC 2 and countdown counter CDC 1 are decremented.
- If a slot with an R-bit of, say, priority 2 passes, then request counters RC 0 and 2 and countdown counter CDC 1 are incremented.
- The segment is transmitted when CDC 1 becomes zero and an empty slot is received.

Hence incrementing the lower priority countdown counter when a request for a higher priority slot is received, effectively delays the transmission of the lower-priority segment. This means that segments with a higher priority are always transmitted ahead of lower-priority segments.

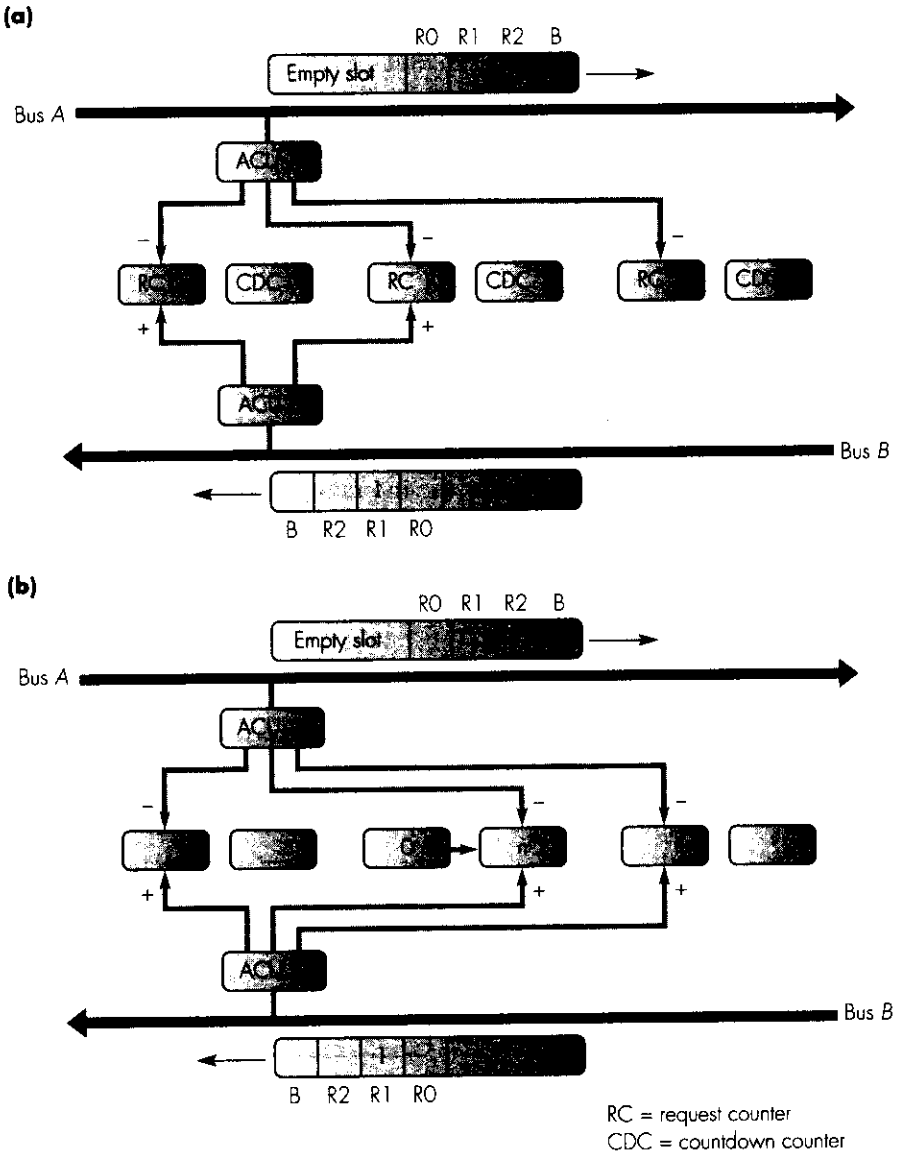


Figure 10.21 Priority access control: (a) no segments waiting; (b) segments queued at priority 1.

10.6.6 Slot and segment formats

As we saw in Section 10.6.2, the 53-octet slot/cell on each bus comprises a 5-octet header and a 48-octet payload (contents) field. The structure of the header is shown in Figure 10.22(a) and, as we can see, is similar to that used with an ATM network.

The *access control* field contains, in addition to the busy bit and three request bits, a *slot type* bit which indicates whether the slot is to be used for queued arbitrated or prearbitrated (isochronous) data. For slots containing either connection-oriented or isochronous data, the 20-bit *virtual channel identifier* (*VCI*) identifies the logical connection to which the cell contents relate. For LAN (also known as connectionless) data, the VCI is set to all 1s. The *payload type* indicates the type of data being carried. It is 00 for all user data – both queued arbitrated and prearbitrated – and the other bit combinations are used for management information. The *priority* has a default value of 00 and, for the moment, no other values have been defined. Finally, the *header check sequence* is an 8-bit CRC for error detection purposes.

To transfer connectionless data across the subnetwork – for example, a MAC frame between two remote bridges – the submitted frame is first divided

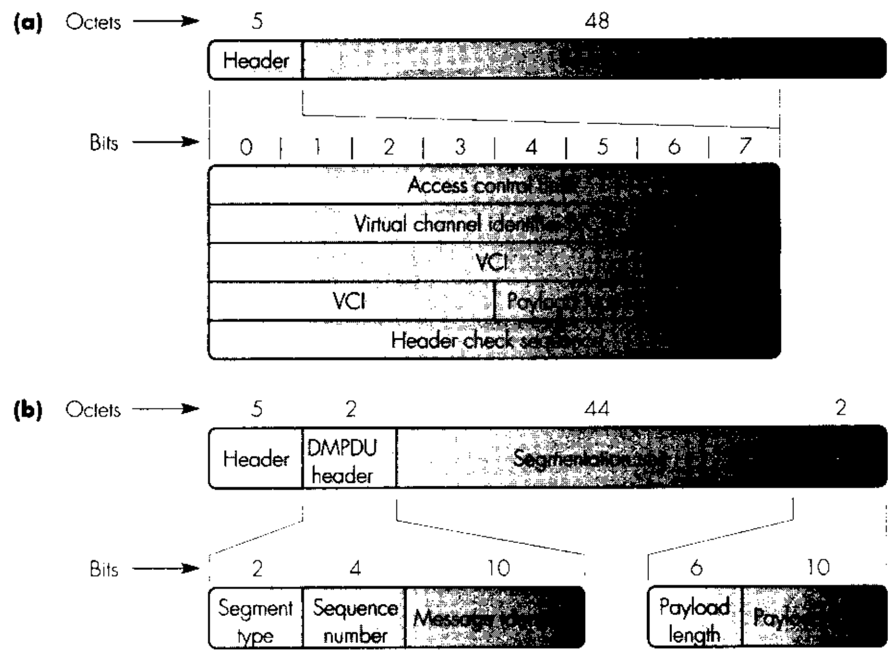


Figure 10.22 Slot and segment formats: (a) slot header; (b) connectionless data segment format.

(segmented) into multiple segments by the MAC convergence protocol in the source access control unit. On receipt, the same protocol at the destination reassembles the received segments back into the original frame. For connection-oriented and isochronous data, the VCI in the cell header is used by the destination to identify those cells which are intended for it. However, for connectionless data, an additional 2-octet header at the start of the 48-octet payload field is used instead. In addition, as Figure 10.22(b) shows, a 2-octet trailer is added at the end of the payload field.

Since segments containing connectionless data are part of a larger (MAC) data frame, they are called **derived MAC protocol data units (DMPDUs)**. The segments relating to a frame – referred to as a message in the standard – are transferred in one of four segment types:

- **single segment message (SSM)**: if the frame/message can be carried in a single segment;
- **beginning of message (BOM)**: indicates this is the first segment of a multiple segment frame/message;
- **continuation of message (COM)**: indicates the contents are between the start and end of a multiple segment frame/message;
- **end of message (EOM)**: indicates the last segment.

The *sequence number* and *message identifier* are used together to enable the destination to reassemble the segments relating to a multiple segment message back into its original form.

The *sequence number* is used to detect any missing segments. It is set to zero in the first segment (BOM) and increments for each successive continuation segment (COM) and last segment (EOM). If a missing segment is detected, the remaining segments of the frame/message are discarded.

All segments relating to the same frame are allocated the same *message identifier (MID)* by the source access unit. This enables the remaining access units on the bus to identify the segments that relate to the same frame. To ensure these are unique, each access unit is allocated a separate block of identifiers when it is initialized. Clearly, a message identifier is not required in single segment messages and hence it is set to zero.

The trailer comprises two fields: the *payload length* and the *payload CRC*. As we shall see in Section 10.6.7, all submitted frames are padded out to be a multiple of 4 octets. This means that a segment may contain from 4 to 44 octets in multiples of 4 octets. Clearly, not all submitted frames will comprise multiples of 44 octets and hence the payload length indicates the actual number of octets in a single segment message or end of message segment. The payload CRC is a 10-bit CRC and is used to detect transmission errors in the entire 48-octet segment.

10.6.7 SMDS

In a public network the connectionless data service offered by the MAC convergence function is known as the switched multimegabit data service (SMDS) or the **connectionless broadband data service (CBDS)** by the ITU-T. In such networks, the various sublayer functions/protocols shown in Figure 10.17 are known as the **SMDS interface protocols (SIP)**: the MAC convergence protocol is known as SIP level 3; the queued arbitrated protocol SIP level 2; and the physical convergence protocol SIP level 1. A typical interconnection schematic and associated internetworking protocol architecture are shown in Figure 10.23. In part(a) the two LANs are interconnected through MAC bridges while in part (b) IP routers are used.

Recall from Chapter 8 that the different types of LAN utilize different header formats, address types, and maximum frame sizes. To accommodate all types of MAC frame, the size of the **SMDS service data unit** can be up to 9188 octets. Also, because of the different addressing formats, prior to segmenting a submitted frame (SDU), the level 3 SIP first encapsulates the frame between a standard header and trailer. In addition, in order to simplify the buffering operation at the destinations, if necessary, it adds additional padding octets at the tail of the submitted frame so that its length is a multiple of 4-octets. The resulting message unit – referred to as a packet – is known as an initial MAC PDU (IMPDU) – or SIP level 3 PDU – and its format is shown in Figure 10.24(a).

Thus the SMDS network provides a connectionless service that is transparent to the customer's internetworking method. To achieve this, on receipt of a frame the access gateway – known as an SMDS edge gateway – simply broadcasts the frame over its local DQDB subnetwork using the queued-arbitrated access control protocol. In this way, a copy of all submitted frames is received by all the other gateways – access nodes – attached to the same DQDB subnetwork and, through them, by all the other bridges/routers. A decision is made by the latter whether to forward the frame on its LAN or simply discard it.

As we can see, the header comprises two or possibly three fields. The common header contains an 8-bit sequence number – known as the *begin-end tag* and used to enable the level 3 SIP to detect missing frames – and a specification of the amount of buffer memory required to store the complete IMPDU. The MAC convergence protocol (MCP) header contains a number of subfields relating to the protocol. These include the addresses of the source and destination gateways which, in a public network, are 60-bit E.164 addresses as defined for use with an ISDN. However, to cater for other address types these are both 64-bit address fields with the most significant 4 bits identifying the address types used in the remaining 60 bits – for example, 16/48-bit MAC addresses. Other subfields include the number of PAD octets present and an indication of whether a CRC is present or not. The header extension has been included to allow additional subfields to be added in the future.

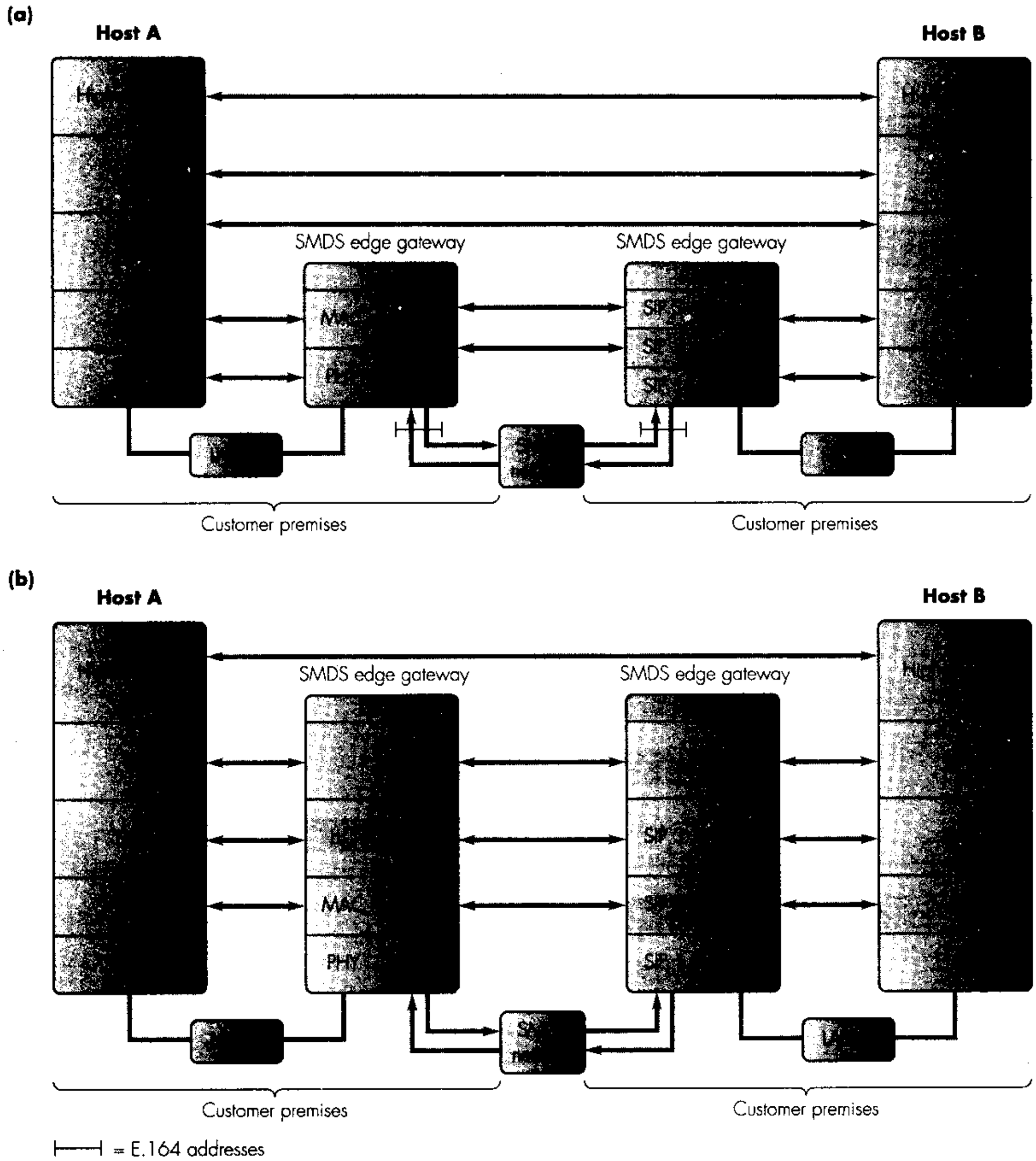


Figure 10.23 SMDS internetworking protocol architectures: (a) bridges; (b) routers.

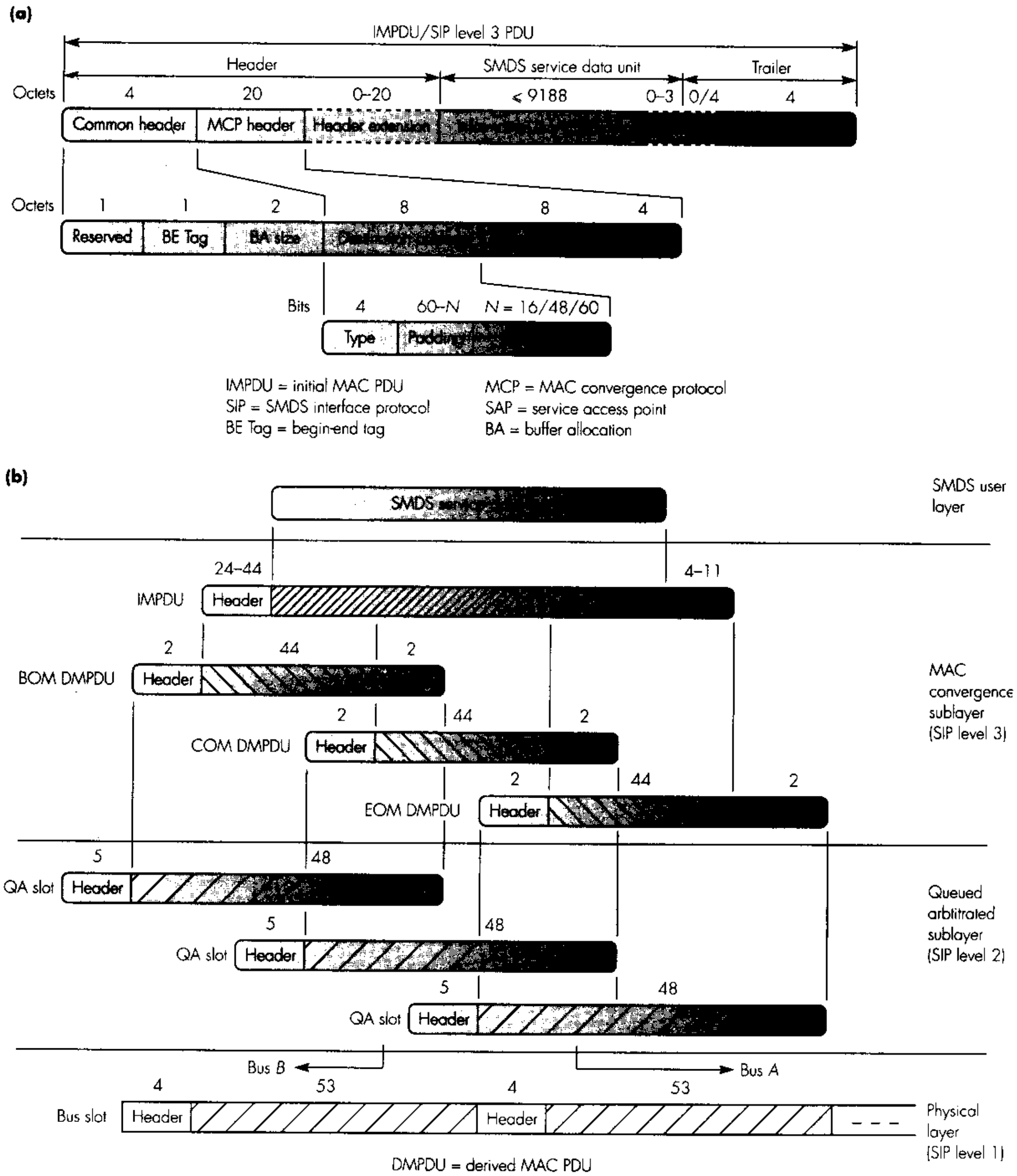


Figure 10.24 Frame transmission overheads: (a) initial MAC PDU format; (b) frame segmentation.

The trailer may include an optional 32-bit CRC which is used for error detection on the complete IMPDU. The common trailer contains the same information as the common header. Hence if the maximum header extension and CRC fields are present and the information field is the maximum 9188 octets, there is an integral number of 210 segments after segmentation.

The steps taken to transfer a submitted MAC frame across the SMDS network, together with the overheads associated with each sublayer function, are summarized in Figure 10.24(b). The submitted SMDS service data unit is first encapsulated by the MAC convergence protocol to form an IMPDU. It then segments this into a number of DMPDUs, each with the corresponding header and trailer. The resulting 48-octet segments are passed to the queued arbitrated function which adds the appropriate 5-octet header to them. Finally these are passed to the common function which initiates their transmission via the physical layer convergence sublayer. We can best quantify the overheads associated with each of these functions using an example.

Example 10.3

A 510-octet MAC frame is to be transferred across an SMDS/DQDB subnetwork. Stating clearly any assumptions you make, derive the number of queued arbitrated slots that are required to carry out the transfer and hence the total number of overhead octets involved.

[illegible]

With reference to Figure 10.24(b):

MAC convergence protocol:

- [illegible] to make the frame 512 octets which is an integral [illegible];
- [illegible] extension and CRC are not used, [illegible] trailer are added [illegible] IMPDU;
- total overheads = 2 + 24 + 4 = 30 octets;
- after segmentation, the IMPDU requires 13 DMPDUs, 12 containing a full complement of 44 octets and one with 12 octets;
- total overheads are 4 for each of the 13 DMPDUs [illegible] plus 32 for the part-full EOM DMPDU.

Queued arbitrated (QA) sublayer:

- a further 5 octets are added to each 48-octet [illegible] 13 QA slots;
- [illegible] 65 octets.

10.3 Continued

- Physical layer [illegible]
  - a further [illegible]
  - total overhead [illegible]
- In summary [illegible]
  - Overall required [illegible]
  - [illegible]

10.7 Wide area ATM networks

As we showed earlier in Figure 10.15(c), a number of network providers have created wide area ATM networks by interconnecting multiple MAN subnetworks together. As we indicated, each MAN subnetwork can be implemented using either DQDB or conventional ATM switches. Each MAN subnetwork is connected to a MAN switching system (MSS) and these in turn are interconnected by means of high bit rate lines. For the MSSs to perform their routing/switching function, the E.164 addresses in the header of each IMPDU are networkwide addresses which identify the MSS and subnetwork to which each access gateway is attached. In addition, group addressing is supported and, when the access node receives an IMPDU with a preassigned group address in its destination address field, a copy of the IMPDU is sent to all members of the group.

The interconnected LANs in the two protocol architectures we showed in Figure 10.23 were assumed to be frame-based LANs. However, a similar approach can be used to interconnect ATM LANs. Figure 10.25(a) shows typical arrangement, together with the associated protocol architecture. In this example we assume the connectionless service associated with the ATM LAN is provided at the IP layer. Hence, as we showed in Figure 10.14 and described in the accompanying text, a connectionless server (CLS) is used. However a similar approach is used when the connectionless service is being provided at the MAC sublayer with a LAN emulation server (LES).

As we can see in Figure 10.25(a), an ATM gateway is located at each customer site, one port of which is connected to the site ATM LAN and the other port to an ATM MAN. Typically, there are a number of gateways (sites) connected to the MAN in a particular area and this provides a local gathering and distribution function. As before, an MSS is connected to each MAN and the MSSs are interconnected together to form a wide area switched network. The service provided is the SMDS/CBDS and the ATM LAN gateway is the SMDS edge gateway.

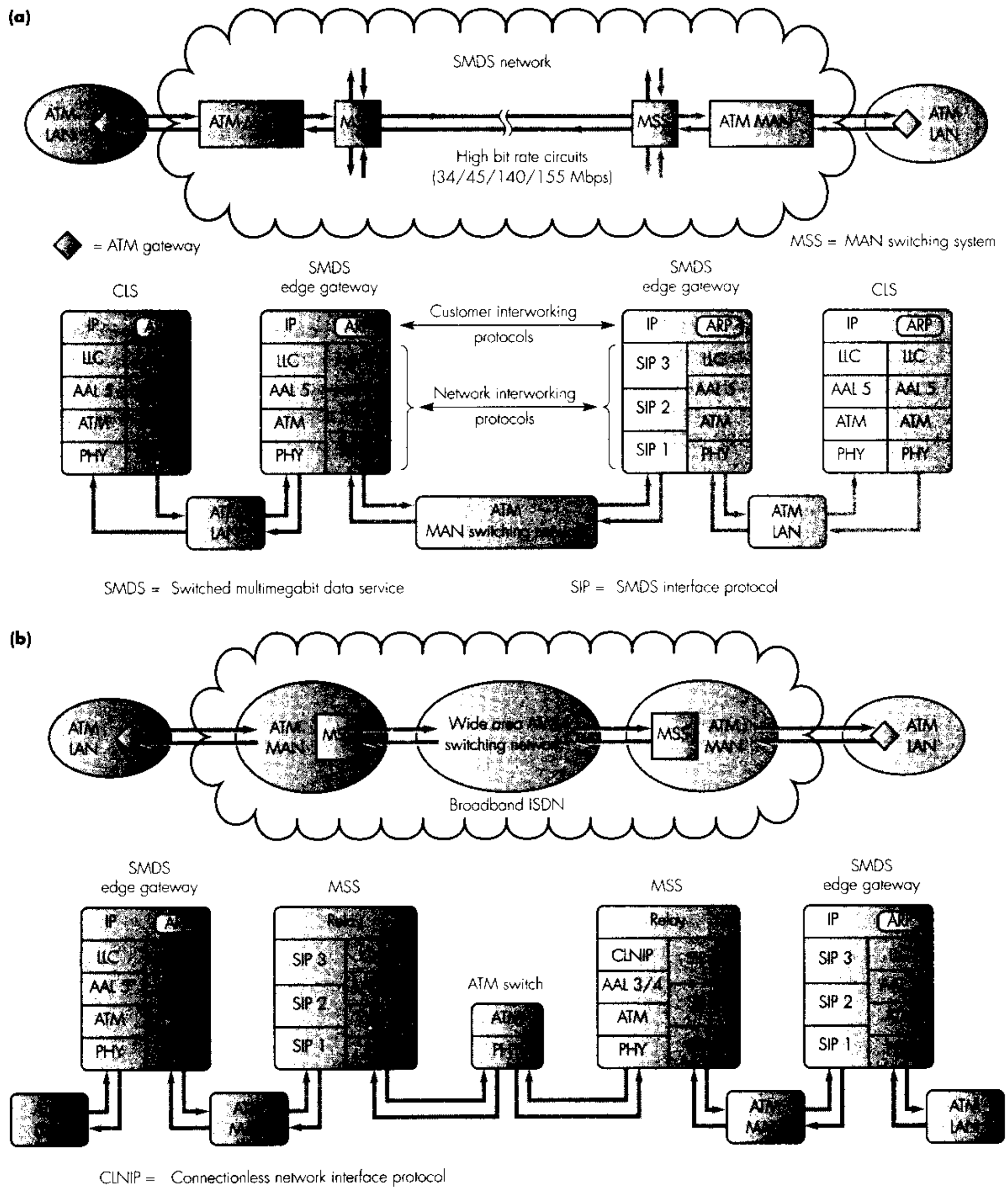


Figure 10.25 Connectionless working over wide area ATM networks: (a) ATM MAN switching network; (b) broadband ISDN.

On receipt of an IP datagram/packet with a destination address that indicates a different site network (netid), the CLS within the ATM LAN relays the packet to the ATM LAN port of the SMDS edge gateway. The packet – or MAC frame if LAN emulation is being used – is first encapsulated into a standardized frame format by the SIP level 3 protocol and then relayed across the SMDS network to the appropriate destination gateway. From there it is relayed first to the site CLS and from there to the destination station.

In some networks, the MSSs are interconnected by an ATM switching network rather than point-to-point high bit rate lines. This type of network, together with the protocol architecture associated with it, is shown in Figure 10.25(b). As we can see, this is similar to part (a) except in this case an intermediate ATM switching network is present. The **connectionless network interface protocol (CLNIP)** provides a similar service to the SIP level 3 protocol and the format of each CLNIP PDU is similar to that given earlier in Figure 10.24(a). The main difference is that there is no common header or trailer with the CLNIP PDU since this is provided by the AAL in an ATM network.

Summary

A summary of the topics discussed in this chapter is given in Figure 10.26.

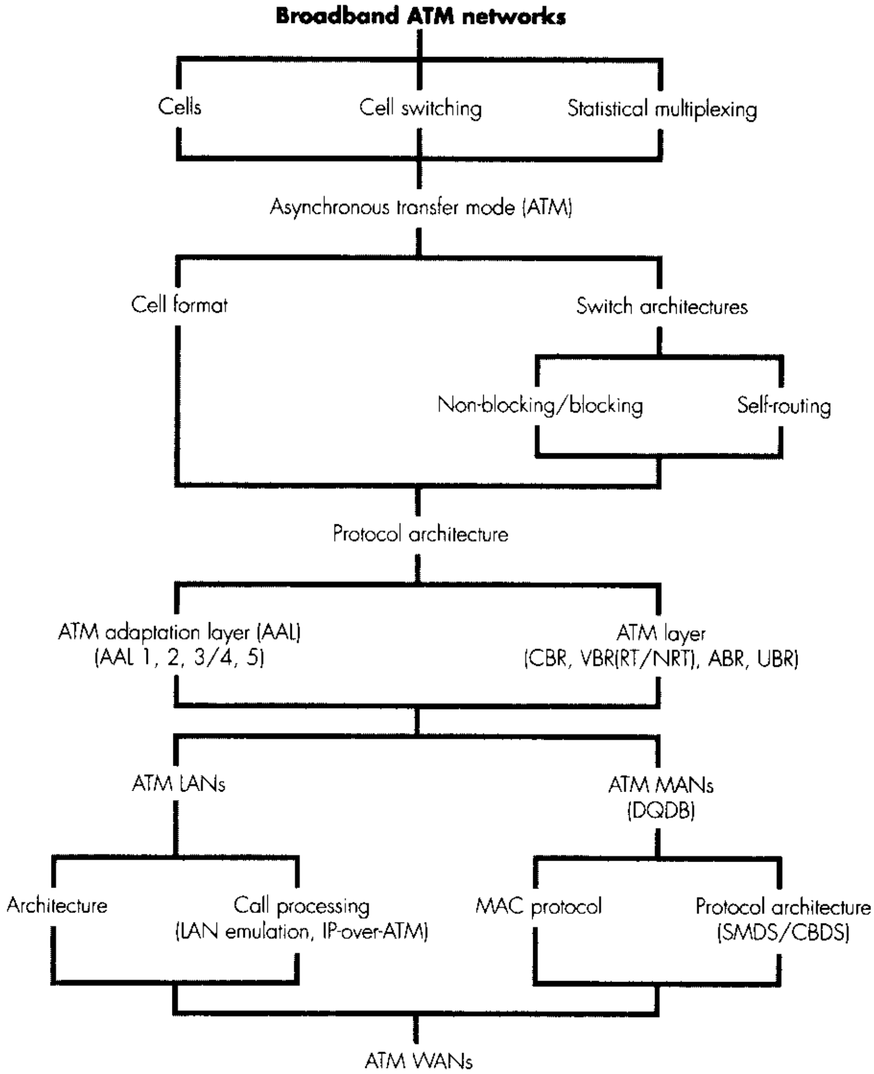


Figure 10.26 Broadband ATM networks, chapter summary.

Exercises

Section 10.1

10.1 Explain the origin/meaning of the following terms relating to B-ISDN networks:
- (i) fixed-size cells,
- (ii) statistical multiplexing,
- (iii) cell switching,
- (iv) asynchronous transfer mode.

10.2 Discuss the reasoning behind the choice of a connection-oriented mode of operation for B-ISDN networks and a 53-byte cell size.

10.3 Explain why the development of B-ISDN has been postponed. Identify those networks and networking equipment of that use ATM.

Section 10.2

10.4 With the aid of the cell switching schematic shown in Figure 10.1, explain
- (i) how the header of each cell can be relatively short,
- (ii) how cells are routed through an ATM switch,
- (iii) the difference between VP routing and VC routing. Give an example of each routing type.

10.5 With the aid of the ATM format shown in Figure 10.2, explain the use of
- (i) the payload type field,
- (ii) the cell loss priority bit,
- (iii) the header checksum.

Section 10.3

10.6 With the aid of the switch architecture shown in Figure 10.3(a), explain the principle of operation of an ATM switch. Include in your explanation the role of the input and output controllers, the switching matrix and the switch control processor. Also explain why cell buffers are required in the input and output controllers.

10.7 With the aid of the diagrams shown in Figure 10.3(b) and (c), explain the principle of operation of the following ATM switch matrix types:
- (i) time division bus,
- (ii) fully-connected.

Assuming they are to operate in a nonblocking mode, estimate the maximum number of I/O ports associated with each matrix type and what determines this.

10.8 Explain why large fabrics consist of multiple switching stages each made up of a number of smaller switching elements interconnected in a regular matrix.

10.9 In the context of a multistage delta switching matrix comprising a number of switching elements, if M is the number of input ports and N is the number of inputs per switching element, derive expressions for the number of switching elements per stage, X, and the number of stages, Y.

10.10 With the aid of the 8-port delta switching matrix shown in Figure 10.4, explain how a cell is routed through the matrix and how blocking can occur.

10.11 An example of a switching matrix that avoids blocking is the Batcher–Banyan switch matrix shown in Figure 10.5. Explain the principle of operation of such switches and how blocking is avoided.

10.12 Using the Batcher–Banyan switch matrix shown in Figure 10.5, identify the paths through the matrix followed by the following set of cells. Assume they arrive simultaneously at the eight input ports and the routing tags given start at port 1:

111, 100, 011, 000, 101, 001, 110, 010

Assume now that the routing tag of the call arriving at port 4 is 111 instead of 000. Determine the effect of this and how it is overcome in practice.

Section 10.4

10.13 With the aid of the protocol architecture shown in Figure 10.6, explain
(i) the role of the signaling control point and the network management stations,
(ii) the meaning of the C, U, and M planes,
(iii) the role of the AAL and ATM layer protocols.

10.14 With the aid of the diagram shown in Figure10.7, state the application domain of each of the four AAL service classes. Also explain the role of the CS and SAR sublayers in implementing these services. Include the use of the service access point.

10.15 With the aid of the two SAR PDU types shown in Figure 10.8, explain the use of the following fields:
(i) SN,
(ii) SNP,
(iii) IT.

10.16 Explain the difference between the AAL3/4 and AAL5 layers and why the latter was developed. Hence, using the PDU formats of the CS and SAR sublayers shown in Figure 10.9, derive the number of protocol overhead octets/bytes required to send an AAL service data unit of 1024 bytes using
(i) AAL3/4,
(ii) AAL5.

10.17 Explain the role of the ATM layer including the meaning of the following service classes:
(i) CBR,
(ii) VBR/RT,
(iii) VBR/NRT,
(iv) ABR,
(v) UBR.

10.18 Explain the meaning of the terms "traffic descriptor" as applied to a call and the use of the following parameters associated with it:
(i) PCR,
(ii) SCR,
(iii) MCR,
(iv) CDVT,
(v) CLR,
(vi) CTD,
(vii) CDV.

10.19 In order to ensure the agreed operational parameters for a call are being adhered to, the network uses the generic cell rate algorithm. With the aid of the four example cell inter-arrival times shown in Figure 10.10, explain the principle of operation of the algorithm.

Section 10.5

10.20 In relation to the schematic diagram shown in Figure 10.11 of an ATM LAN explain the role of the following:
(i) an RCU,
(ii) the broadcast server,
(iii) the signaling control point and how stations communicate with it,
(iv) the LAN emulation server.

10.21 With the aid of the connectionless protocol architecture shown in Figure 10.14, outline the steps followed to enable:
(i) a cell-based client to access a cell-based server
(ii) a PC/workstation attached to a legacy LAN to access a server that is attached to a different legacy LAN (of the same type) both of which are connected to the ATM backbone by bridges
(iii) a PC/workstation that is attached to a legacy LAN to access a server that is attached directly to the ATM backbone. Assume the legacy LAN is connected to the ATM backbone by a router.

Section 10.6

10.22 With the aid of the three DQDB MAN architectures shown in figure 10.15, explain the meaning/use of
(i) dual contradirectional buses,
(ii) access node/customer network interface unit,
(iii) open bus and dual bus topologies,
(iv) isochronous gateway,
(v) MAN switching system,
(vii) SMDS.

10.23 With the aid of the DQDB architecture shown in Figure 10.16(a) and (b), state the differences between an open bus and a loop bus architecture. Explain the principle of operation of each type.

10.24 With the aid of the DQDB architecture shown in Figure 10.16(c). describe how a looped bus architecture can continue working in the presence of
(i) a communications link failure and
(ii) a node failure.

10.25 With the aid of the DQDB protocol architecture shown in Figure 10.17, explain the meaning/use of
(i) queued arbitrated and MAC convergence functions,
(ii) prearbitrated and isochronous conversion function,
(iii) common functions,
(iv) physical layer convergence function.

10.26 With the of a diagrams in Figure 10.18 and the flowchart in Figure 10.19, explain the principle of operation of the queued-packet distributed-switch (QPSX) access control method. Include the meaning/use of:
(i) the busy and request bits in the header of each slot,
(ii) the request and countdown counters,
(iii) the request-queue counter.

10.27 With the aid of Figure 10.20(a), explain why bandwidth balancing is necessary with the QPSX access control method. Hence explain how the remedial actions shown in Figure 10.20(b) overcome the unfairness arising with the basic scheme.

10.28 With the aid of the access control schematic shown in Figure 10.21, explain the principle of operation of the distributed priority queuing mechanism used with multimedia applications. Include the role of the various counters and the effect on their contents of the request and busy bits when
(i) no segments are queued for transmission, and
(ii) a segment is queued at a specified priority level.

10.29 Define the meaning of the terms "DQDB slot" and "DQDB segment". Hence with the aid of the formats shown in Figure 10.22, explain the role of the slot and segment headers.

10.30 Assuming a frame of data is to be transferred over a DQDB subnetwork and the frame requires multiple slots, with the aid of the segment format shown in Figure 10.22(b), explain the function of each field in the DMPDU header and trailer. Describe also the steps followed by the receiver when reassembling the frame.

10.31 Define the service offered by an SMDS network. Hence with the aid of the alternative protocol architectures shown in Figure 10.23, explain the function of the SIP level 3 and SIP level 2 protocols when the SMDS edge gateway is
(i) a bridge and
(ii) a router.

10.32 Define the format of the source and destination addresses used in the header of an initial MAC PDU and why the size of an SMDS service data unit can be up to octets/bytes.

10.33 A 100-byte MAC frame is to be transferred access a DQDB SMDS network. Using the transmission overheads identified in Figure 10.24, derive:
(i) the number of queued arbitrated slots that are required to carry out the transfer of the frame,
(ii) the total number of overhead bytes. State any assumptions you make.

Section 10.7

10.34 With the aid of the ATM MAN switching network and associated protocol architecture shown in Figure 10.25(a), explain
(i) the role of the MAN switching system including the addressing mechanism associated with it,
(ii) how an IP datagram/packet is routed across the total network.

10.35 Identify and explain the differences when an ATM switch is used within the SMDS network. Include the role of the CLNIP protocol in the MSS.