CHAPTER 7
Congestion Control in ATM Networks

Congestion control, otherwise known in the ATM Forum standards as *traffic management*, is a very important component of ATM networks. It permits an ATM network operator to carry as much traffic as possible so that revenues can be maximized without affecting the quality of service offered to the users.

As we will see in this Chapter, an ATM network can support several quality-of-service categories. A new connection, at call set-up time signals to the network the type of quality-of-service category that it requires. If the new connection is accepted, the ATM network will provide the requested quality of service to this connection without affecting the quality of service of all the other existing connections. This is achieved using congestion control in conjunction with a scheduling algorithm that is used to decide in what order cells are transmitted out of an ATM switch (see section 6.7).

Two different classes of congestion control schemes have been developed, namely, *preventive* congestion control and *reactive* congestion control. In preventive congestion control, as its name implies, we attempt to prevent congestion from occurring. This is done using the following two procedures: *call (or connection) admission control* (CAC), and *bandwidth enforcement*. Call admission control is exercised at the connection level and it is used to decide whether to accept or reject a new connection. Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the source transmitting on this connection is within its negotiated traffic parameters.

Reactive congestion control is based on a totally different philosophy than preventive congestion control. In reactive congestion control, the network uses feedback

messages to control the amount of traffic that an end-device transmits so that congestion does not arise.

In this Chapter, we first present the parameters used to characterize ATM traffic, the quality-of-service parameters, and the ATM quality-of-service categories. Then, we describe in detail the preventive and the reactive congestion control schemes.

7.1 Traffic characterization

The traffic submitted by a source to an ATM network can be described by the following traffic parameters: *peak cell rate* (PCR), *sustained cell rate* (SCR), *maximum burst size* (MBS), *burstiness*, and *correlation of inter-arrival times*. Also, various probabilistic and empirical models have been used to describe the arrival process of cells. Below, we examine these traffic parameters in detail and we also briefly introduce some empirical and probabilistic models. Two additional parameters, namely, *cell delay variation tolerance* (CDVT) and *burst tolerance* (BT), will be introduced later on in this Chapter.

Peak cell rate (PCR)

This is the maximum amount of traffic that can be submitted by a source to an ATM network, and it is expressed as ATM cells per second. Due to the fact that transmission speeds are expressed in bits per second, it is more convenient to talk about the peak bit rate of a source, i.e., the maximum number of bits per second submitted to an ATM connection, rather than its peak cell rate. The peak bit rate can be translated to the peak cell rate, and vice versa, if we know which ATM adaptation layer is used. The peak cell rate has been standardized by both the ITU-T and the ATM Forum..

Sustained cell rate (SCR)

Let us assume that an ATM connection is up for a period of time equal to D. During that time, the source associated with this connection transmits at a rate that varies over time. Let S be the total number of cells transmitted by the source during the period D. Then, the average cell rate of the source is S/D. (One would be inclined to use the abbreviation ACR for the average cell rate, but this abbreviation is used to indicate the *allowed cell rate* in the ABR mechanism described in section 7.8.1!)

The average cell rate has not been standardized by ITU-T or by the ATM Forum. Instead, an upper bound of the average cell rate, known as the *sustained cell rate* (SCR) has been standardized by the ATM Forum. This is obtained as follows.

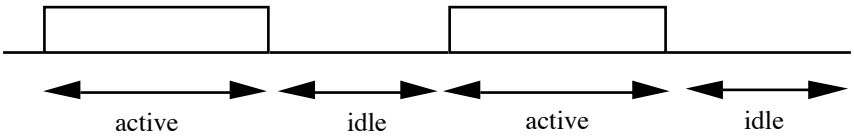


Figure 7.1: A bursty source

Let us first calculate the average number of cells submitted by the source over successive short periods T. For instance, if the source transmits for a period D equal to 30 minutes and T is equal to 1 second, then there are 1800 T periods and we will obtain 1800 averages, one per period. The largest of all these averages is called the sustained cell rate. We observe that the SCR of a source cannot be larger than the source's PCR nor can it be less than the source's average cell rate.

The SCR is not to be confused with the average rate of cells submitted by a source. However, if we set T equal to D, then the SCR simply becomes the average cell rate at which the source submits cells to the ATM network. For instance, in the above example, the SCR will be equal to the average cell rate, if T is equal to 30 minutes. The value of T is not defined in the standards, but in the industry it is often taken to be equal to 1 second.

Maximum burst size (MBS)

Depending upon the type of the source, cells may be submitted to the ATM network in bursts. These bursts may be fixed or variable in size. For instance, in a file transfer, if the records retrieved from the disk are of fixed size, then each record results to a fixed number of ATM cells submitted to the network back-to-back. In an encoded video transfer, however, each coded image has a different size, which results to a variable number of cells submitted back-to-back. The *maximum burst size* (MBS) is defined as the maximum number of cells that can be submitted by a source back-to-back at peak cell rate. The MBS was standardized by the ATM Forum.

Burstiness

This is a notion related as to how the cells transmitted by a source are clumped together. Typically, a source is bursty if it transmits for a period of time and then becomes idle for another period of time, as shown in figure 7.1. The longer the idle period, and the higher the arrival rate during the active period, the more bursty the source is.

The burstiness of a source can significantly affect the cell loss in an ATM switch. Let us consider an output buffer of the output buffering non-blocking ATM switch,

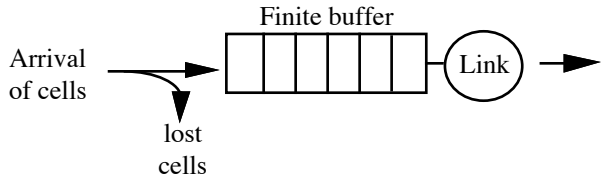


Figure 7.2: A finite capacity buffer.

shown in figure 6.23. This buffer is shown in figure 7.2. It has a finite capacity queue and it is served by a link indicated in the figure by a circle. The arrival stream of ATM cells to the queue can be seen as the superposition of several different arrival streams coming from the input ports of the switch. A cell that arrives at a time when the queue is full is lost.

Now, from queueing theory we know that as the arrival rate increases, the cell loss increases as well. What is interesting to observe is that a similar behaviour can be also seen for the burstiness of a source. The curve in figure 7.3, shows qualitatively how the cell loss rate increases as the burstiness increases while the arrival rate remains constant. (Detailed curves relating the cell loss probability to the burstiness of the arrival process can be obtained by carrying out the simulation project "A simulation model of an ATM multiplexer – Part 2" described at the end of this Chapter.)

Correlation

Let us consider successive inter-arrival times of cells generated by a source, as shown in figure 7.4. In an ATM environment it is highly likely that the inter-arrival times are correlated either positively or negatively. Positive correlation means that, if an inter-

arrival time is large (or small), then it is highly likely that the next inter-arrival time will also be large (or small). Negative correlation implies the opposite. That is, if an inter-arrival time is large (or small), then it is highly likely that the next inter-arrival time will be small (or large). As in the case of burstiness, the correlation of the inter-arrival time of cells can significantly affect the cell loss probability in an ATM switch.

7.1.1 Standardized traffic descriptors

The ATM Forum has standardized the following traffic descriptors: peak cell rate, cell delay variation tolerance, sustained cell rate, and maximum burst size. The ITU-T has only standardized the peak cell rate. The peak cell rate, sustained cell rate, and maximum burst

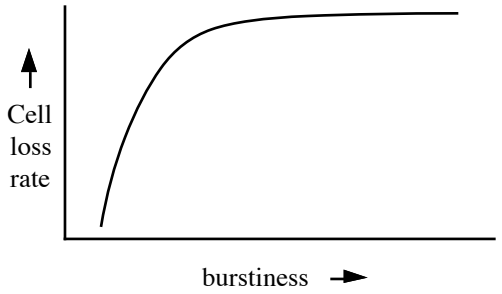


Figure 7.3: Cell loss rate vs burstiness

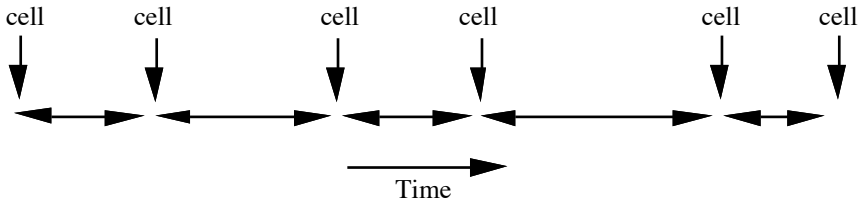


Figure 7.4: Successive inter-arrival times of cells

size depend upon the characteristics of the source. The cell delay variation tolerance is used in the *generic cell rate algorithm* (GCRA), discussed later on in section 7.7.1 of this Chapter, and it is independent of the characteristics of the source. It is specified by the administrator of the network to which the source is directly attached.

7.1.2 Empirical models

Several empirical models have been developed to predict the amount of traffic generated by an variable bit rate MPEG video coding algorithm. These empirical models are statistical models and they are based on regression techniques.

MPEG is a standards group in ISO that is concerned with the issue of compression and synchronization of video signals. In MPEG, successive video frames are compressed following a format like: I B B B P B B B P B B B I, where I stands for *I-frame*, B for *B-frame*, and P for *P-frame*. An intra-coded frame, or I-frame, is an encoding of a picture based entirely on the information in that frame. A predictive-coded frame, or P-frame, is based on motion compensated prediction between that frame and the previous I- or P-frame. A bidirectional-coded frame, or B-frame, is based on motion compensated prediction between that frame and the previous I- or P-frame or the next I- or P-frame.

The encoder also can select the sequence of I, P, and B frames, which form a group of frames known as a *group of pictures* (GOP). The group of frames repeats for the entire duration of the video transmission.

The size of the resulting frame varies significantly between frame types. I-frames are the largest while B-frames are the smallest. The size of an I-frame varies based on picture content. P- and B-frames vary depending on the motion present in the scene as well as picture content.

The number of bits produced by each frame in such a sequence is correlated and it can be predicted using an *autoregressive-moving average* (ARMA) model. Such a model can be used in a performance evaluation study to generate video traffic. (See the simulation project “Estimating the ATM traffic parameters of a video source”, given at the end of this Chapter.)

7.1.3 Probabilistic models

Probabilistic models of arrival processes are abstractions of real-life arrival processes. They do not represent real-life arrival processes exactly, but they capture some of the

traffic parameters described above, and in view of this, they are extremely useful in performance evaluation studies.

When we talk about a probabilistic model of an ATM arrival process, we assume that the arrival process is generated by a source which transmits cells over an ATM link. The link is assumed to be used exclusively by this source, and it is slotted with a slot being equal to the time it takes for the link to transmit a cell. Now, if we place ourselves in front of the link and observe the slots go by, then we will see that some of the slots carry a cell while others are empty. A model of an ATM arrival process describes which slots carry a cell and which slots are idle.

ATM sources are classified into *constant bit rate* (CBR) and *variable bit rate* (VBR). A CBR source generates the same number of bits every unit time whereas a VBR source generates traffic at a rate that varies over time. Examples of CBR sources are circuit emulation services such as T1 and E1, unencoded voice, and high quality audio. Examples of VBR sources are encoded video, encoded voice with suppressed silence periods, IP over ATM, and frame relay over ATM. The arrival process of a CBR source is easy to characterize. The arrival process of a VBR source is more difficult to characterize and it has been the object of many studies.

CBR sources

As mentioned above, a CBR source generates the same number of bits every unit time. For instance, a 64 Kbps unencoded voice produces 8 bits every 125 msec. Since the generated traffic stream is constant, the PCR, SCR, and average cell rate of a CBR source are all the same, and a CBR source can be completely characterized by its PCR.

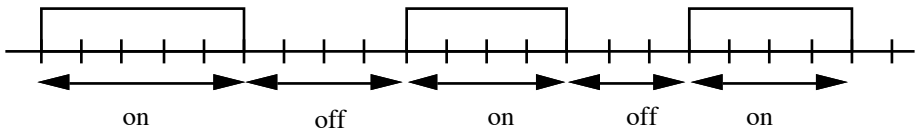


Figure 7.5: The on/off process

Let us assume that a CBR source has a PCR equal to 150 cells per second, and the ATM link over which it transmits has a speed, expressed in cells per second, of 300.

Then, if we observe the ATM link, we will see that every other slot carries a cell. If the speed of the link is 450 cells per second, then every third slot carries a cell, and so on.

VBR sources

A commonly used traffic model for data transfers is the *on/off process* shown in figure 7.5. In this model, a source is assumed to transmit only during an active period, known as the *on period*. This period is followed by a silent period, known as the *off period*, during which the source does not transmit. This cycle of an on period followed by an off period repeats continuously until the source terminates its connection. During the on period there may be a cell transmitted every slot, or every fixed number of slots, depending upon the source's PCR and the speed of the link.

The PCR of an on/off source, is the rate at which it transmits cells during the on period. For example, if it transmits every other slot, then its PCR is equal to half the speed of the link, where the link's speed is expressed in cells per second. Alternatively, we can say that the source's peak bit rate is half the link's capacity, expressed in bits per second. The average cell rate is:

$$\frac{\text{PCR x mean length of on period}}{\text{mean length of on and off period}}$$

The on/off model captures the notion of burstiness, which is an important traffic characteristic in ATM networks. The burstiness of a source is indicative of how cells are clumped together. There are several different ways of measuring burstiness. The simplest one is to express it as the ratio of the mean length of the on period divided by the sum of the mean on and off periods, that is

$$r = \frac{\text{mean on period}}{\text{sum of mean on and off periods}}$$

This quantity can be also seen as the fraction of time that the source is active transmitting. When r is close to 0 or to 1, the source is not bursty. The burstiness of the source increases as r approaches 0.5. Another commonly used measure of burstiness, but more complicated to calculate, is the squared coefficient of variation of the inter-arrival

times defined by $Var(X)/(E(X))^2$, where X is a random variable indicating the inter-arrival times.

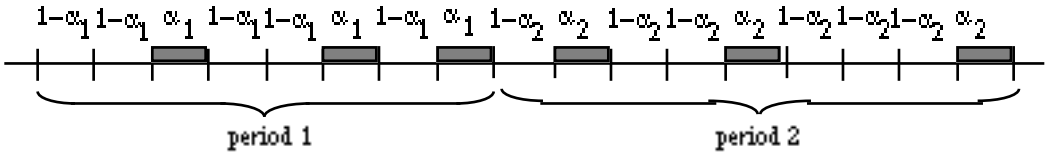


Figure 7.6: The two-state MMBP

The length of the on and off periods of the on/off process follow an arbitrary distribution. A special case of the on/off process is the well-known *interrupted Bernoulli process* (IBP) which has been used extensively in performance studies of ATM networks. In an IBP the on and off periods are geometrically distributed and cells arrive during the on period in a Bernoulli fashion. That is, during the on period, each slot contains a cell with probability α or it is empty with probability 1- α.

The IBP process can be generalized to the two-state *Markov modulated Bernoulli process* (MMBP). A two-state MMBP consists of two alternating periods, period 1 and 2. Each period is geometrically distributed. During period i, we have Bernoulli arrivals with rate α_i, i = 1,2. That is, each slot during period i has α_i probability of containing a cell, as shown in figure 7.6.Transitions between the two periods are as follows:

	period 1	period 2
period 1	p	1-p
period 2	1-q	q

That is, if the process is in period 1 (period 2), then in the next slot it will be in the same period with probability p (q) or it will change to period 2 (period 1)with probability 1-p (1-q). A two-state MMBP model captures both the notion of burstiness and the correlation of inter-arrival times. More complicated MMBPs can be obtained using n different periods.

The above arrival processes were defined in discrete time. That is, we assumed that the link is slotted, and the length of the slot is equal to the time it takes to transmit a cell. Similar arrival processes have been defined in continuous time. In this case, the

underlying assumption is that the link is not slotted, and the arrival of an ATM cell can occur at any time. The continuous-time equivalent of the IBP is the *interrupted Poisson process* (IPP) which is a well known process used in teletraffic studies. In an IPP the on and off periods are exponentially distributed and cells arrive in a Poisson fashion during the on period. An alternative model can be obtained using the fluid approach. In this case, the on and off periods are exponentially distributed as in the IPP model, but the arrivals occur during the on period at a continuous rate, like fluid flowing in. This model has been used extensively in performance studies, and it is referred to as the *interrupted fluid process* (IFP).

The IPP can be generalized to a two-state *Markov modulated Poisson process* (MMPP), which consists of two alternating periods, period 1 and 2. Each period i, i = 1,2, is exponentially distributed with a mean $1/\mu_i$ and during the ith period arrivals occur in a Poisson fashion at the rate of λ_i. More complicated MMPPs can be obtained using n different periods.

7.2 Quality of service (QoS) parameters

A number of different parameters can be used to express the quality of service of a connection, such as, *cell loss rate* (CLR), *jitter*, *cell transfer delay* (CTD), *peak-to-peak cell delay variation*, and *maximum cell transfer delay* (max CTD).

The cell loss rate (CLR) is a very popular quality-of-service parameter and it was the first one to be used in ATM networks. This is not surprising, since there is no flow control between two adjacent ATM switches or between an end-device and the switch to which it is attached. Also, cell loss is easy to quantify, as opposed to other quality-of-service parameters such as jitter and cell transfer delay. Minimizing the cell loss rate in an ATM switch has been used as a guidance to dimensioning ATM switches, and also a large number of call admission control algorithms were developed based on the cell loss rate.

The jitter is an important quality-of-service parameter for real-time applications, such as voice and video. In these applications, the inter-arrival gap between successive cells at the destination end-device cannot be greater than a certain value, as this may cause the receiving play-out process to pause. In general, the inter-departure gaps

between successive cells transmitted by the sender are not the same as the inter-arrival gaps at the receiver. Let us consider figure 7.7. The gap between the end of the transmission of the ith

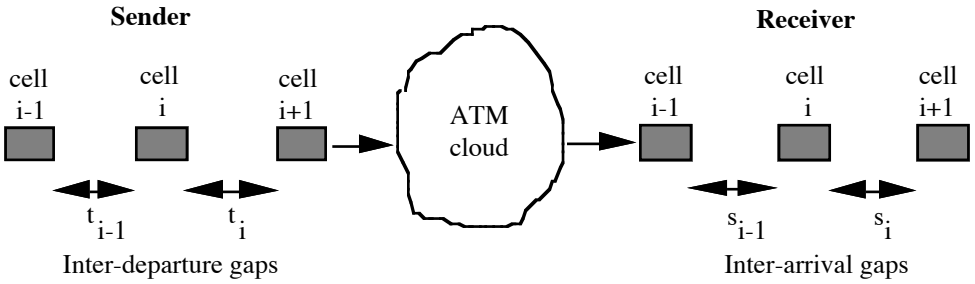


Figure 7.7: Inter-departure and inter-arrival gaps

cell and the beginning of the transmission of the (i+1)st cells is t_i. The gap between the end of the arrival of the ith cell and the beginning of the arrival of the (i+1)st cell is s_i. The inter-departure gap t_i may be less, equal, or greater than s_i. This is due to buffering and congestion delays in the ATM network. This variability of the inter- arrival times of cells at the destination is known as jitter.

It is important that the service provided by an ATM network for a voice or a video connection is such that the jitter is bounded. If the inter-arrival gaps s_i are less than the inter-departure gaps t_i, then the play-out process will not run out of cells. (If this persists for a long period of time, however, it may cause over-flow problems). If the inter-arrival gaps are consistently greater than the inter-departure gaps, then the play-out process will run out of cells and will pause. This is not desirable, since the quality of the voice or video delivered to the user will be affected. Bounding jitter is not easy to accomplish.

The cell transfer delay (CTD) is the time it takes to transfer a cell end-to-end, that is, from the UNI of the transmitting end-device to the UNI of the receiving end-device. It is made up of a fixed component and a variable component. The fixed cell transfer delay is the sum of all fixed delays that a cell encounters from the transmitting end-device to the receiving end-device, such as, propagation delay, fixed delays induced by transmission systems, and fixed switch processing times. The variable cell transfer delay,

known as the *peak-to-peak cell delay variation*, is the sum of all variable delays that a cell encounters from the transmitting end-device to the receiving end-device. These delays are primarily due to queueing delays in the switches along the cell's path. The peak-to-peak cell delay variation should not to be confused with the cell delay variation tolerance (CDVT) which is used in the generic cell rate algorithm (GCRA) described in section 7.7.1.

The *maximum cell transfer delay* (max CTD) is another quality-of-service parameter that defines an upper bound on the end-to-end cell transfer delay. This upper bound is not an absolute bound. Rather, it is a statistical upper bound, which means that the actual end-to-end cell transfer delay may occasionally exceed max CTD. That is, the sum of the fixed cell transfer delay and the peak-to-peak cell delay variation may exceed max CTD, as shown in figure 7.8. For example, let us assume that the max CTD is set to 20 msec and the fixed CTD is equal to 12 msec. Then, there is no guarantee that the peak-to-peak cell delay variation will always be less than 8 msec. The max CTD can be obtained as a percentile of the end-to-end cell transfer delay, so that the end-to-end cell transfer delay exceeds it only a small percent of the time. For instance, if it is set to the 99th percentile, then 99% of the time the end-to-end cell transfer delay will be less than max CTD and 1% of the time it will be greater.

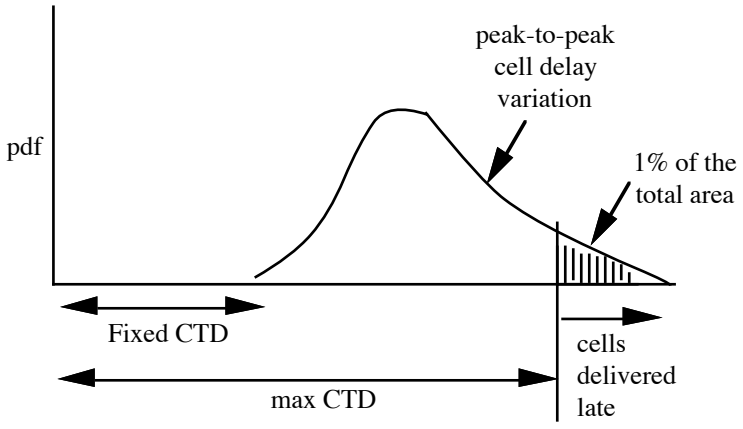


Figure 7.8: Cell transfer delay

Of the quality-of-service parameters described above, the CLR, the peak-to-peak cell delay variation, and the max CTD have been standardized by the ATM Forum and they can be signalled at call set-up time. That is, at call set-up time, the calling party can

specify values for these parameters. These values are upper bounds, and they represent the highest acceptable (and consequently the least desired) values. The values for the peak-to-peak cell delay variation and for the max CTD are expressed in msec. As an example, the calling party can request that the CLR is less or equal than 10^{-6}, the peak-to-peak cell delay variation is less or equal than 3 msec, and the max CTD is less or equal than 20 msec.

The network will accept the connection, if it can guarantee the requested quality-of-service values. If it cannot guarantee these values then it will reject the connection. Also, it is possible that the network and the calling party may negotiate new values for the quality-of-service parameters. As will be seen in the following section, the number of quality-of-service parameters signalled at call set-up time depends on the type of ATM service requested by the calling party.

Three additional quality-of-service parameters are used, namely the *cell error rate* (CER), the *severely errored cell block ratio* (SECBR) and the *cell misinsertion rate* (CMR). These three parameters are not used by the calling party at call set-up. They are only monitored by the network.

The cell error rate (CER) of a connection is the ratio of the number of *errored* cells, that is, cells delivered to the destination with erroneous payload, to the total number of cells transmitted by the source.

The severely errored cell block ratio (SECBR) is the ratio of the total number of severely errored *cell blocks* divided by the total number of transmitted cell blocks. A cell block is a sequence of cells transmitted consecutively on a given connection. A severely errored cell block occurs when more than a pre-defined number of errored cells, lost cells, or misinserted cells are observed in a received cell block.

The cell misinsertion rate (CMR) is the number of cells delivered to a wrong destination divided by a fixed time interval. A misinserted cell is a cell transmitted on a different connection due to an undetected error in its header.

7.3 ATM service categories

An ATM service category is in simple terms a quality-of-service class. Each service category is associated with a set of traffic parameters and a set of quality-of-

service parameters. Functions such as call admission control and bandwidth allocation (see section 7.6) are applied differently for each service category. Also, as described in section 6.7, the scheduling algorithm that determines in what order the cells in an output buffer of an ATM switch are transmitted out, provides different priorities to cells belonging to different service categories. In addition, a service category may be associated with a specific mechanism that is in place inside the network. The service category of a connection is signalled to the network at call set-up time, along with its traffic and quality-of-service parameters.

The ATM Forum has defined the following six service categories: *constant bit rate* (CBR), *real-time variable bit rate* (RT-VBR), *non-real-time variable bit rate* (NRT-VBR), *available bit rate* (ABR), *unspecified bit rate* (UBR), and *guaranteed frame rate* (GFR). The first two service categories, namely CBR and RT-VBR are for real-time applications, whereas the remaining service categories are for non-real-time applications.

The CBR service

This service is intended for real-time applications which transmit at constant bit rate, such as circuit emulation services and constant-bit rate video.

Since the rate of transmission of a constant-bit rate application does not change over time, the peak cell rate is sufficient to describe the amount of traffic that the application transmits over the connection. The cell delay variation tolerance (CDVT) is also specified, and its use will be explained in section 7.7.1. A CBR service is for real-time applications, and therefore, the end-to-end delay is an important quality-of-service parameter. In view of this, in addition to the CLR, the two delay-related parameters, namely the peak-to-peak cell delay variation and the max CTD, are also specified.

In summary, the following traffic parameters are specified: PCR and CDVT. Also, the following quality-of-service parameters are specified: CLR, peak-to-peak cell delay variation, and max CTD.

The RT-VBR service

This service is intended for real-time applications which transmit at a variable bit rate, such as encoded video and encoded voice.

Since the rate of transmission of a variable-bit rate application varies over time, the peak cell rate is not sufficient to describe the amount of traffic that the application will transmit over the connection. In addition to the PCR and the cell delay variation tolerance, the sustained cell rate (SCR) and the maximum burst size (MBS) are specified. As in the CBR service, the RT-VBR service is also intended for real-time applications. Therefore, in addition to the CLR, the two delay-related parameters, namely the peak-to-peak cell delay variation and the max CTD, are also specified.

In summary, the following traffic parameters are specified: PCR, CDVT, SCR, and MBS. Also, the following quality-of-service parameters are specified: CLR, peak-to-peak cell delay variation, and max CTD.

The NRT-VBR service

This service is intended for non-real-time applications which transmit at a variable bit rate. As in the RT-VBR service, the traffic parameters PCR, the cell delay variation tolerance (CDVT), the sustained cell rate (SCR) and the maximum burst size (MBS) are specified. Since this service is not intended for real-time applications only the CLR is specified.

In summary, the following traffic parameters are specified: PCR, CDVT, SCR, MBS. Also, the CLR quality-of-service parameter is specified.

The UBR service

This is a best-effort type of service for non-real-time applications with variable bit rate. It is intended for applications that involve the transfer of data, such as file transfer, web browsing, and email. No traffic or quality-of-service parameters are specified.

The PCR and the CDVT can be specified but a network can ignore it. Also, a UBR user may indicate a *desirable minimum cell rate* (DMCR), but a network is not required to guarantee such as a minimum bandwidth.

The ABR service

This service is intended for non-real-time applications which can vary their transmission rate according to the congestion level in the network.

A user requesting the ABR service specifies a *minimum cell rate* (MCR) and a maximum cell rate, which is its PCR. The minimum cell rate could be zero. The user varies its transmission rate between its MCR and its PCR in response to feedback messages that it receives from the network. These feedback messages are conveyed to the user through a mechanism implemented in the network. During the time that the network has a slack capacity, the user is permitted to increase its transmission rate by an increment. When congestion begins to build up in the network, the user is requested to decrease its transmission rate by a decrement. A detailed description of the ABR service is given in section 7.8.1.

The following traffic parameters are specified: PCR, CDVT, and MCR. The CLR for ABR sources is expected to be low. Depending upon the network, a value for the CLR may be specified.

The GFR service

This service is for non-real-time applications that require a minimum cell rate (MCR) guarantee, but they can transmit in excess of their requested MCR. The application transmits data organized into frames, and the frames are carried in AAL 5 CPS-PDUs. The network does not guarantee delivery of the excess traffic. When congestion occurs, the network attempts to discard complete AAL 5 CPS-PDUs rather than individual cells. The GFR service does not provide explicit feedback to the user regarding the current level of congestion in the network. Rather, the user is supposed to determine network congestion through a mechanism such as TCP, and adapt its transmission rate.

The following traffic parameters are specified: PCR, minimum cell rate (MCR), maximum burst size (MBS), and *maximum frame siz*e (MFS). The CLR for the frames that are eligible for the service guarantee is expected to be low. Depending upon the network, a value for the CLR may be specified.

ATM transfer capabilities

In the ITU-T standard, the ATM service categories are referred to as *ATM transfer capabilities*. Some of the ATM transfer capabilities are equivalent to ATM Forum's service categories, but they have a different name. The CBR service is called the *deterministic bit rate* (DBR) service, the RT-VBR service is called the *real-time statistical bit rate* (RT-SBR) service and the NRT-VBR service is called the *non-real-time statistical bit rate* (NRT-SBR) service. The UBR service category has no equivalent ATM transfer capability. Both the ABR and GFR services have been standardized by ITU-T. Finally, the ITU-T ATM transfer capability *ATM block transfer* (ABT), described in section 7.6.2, has no equivalent service category in the ATM Forum standards.

7.4 Congestion control

Congestion control procedures can be grouped into the following two categories: *preventive* control and *reactive* control.

In preventive congestion control, as its name implies, we attempt to prevent congestion from occurring. This is achieved using the following two procedures: *call (or connection) admission control* (CAC), and *bandwidth enforcement*. Call admission control is exercised at the connection level and it is used to decide whether to accept or reject a new connection. Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the source transmitting on this connection is within its negotiated traffic parameters.

Reactive congestion control is based on a different philosophy than preventive congestion control. In reactive congestion control, the network uses feedback messages to control the amount of traffic that an end-device transmits so that congestion does not arise.

In the remaining sections of the Chapter, we examine in detail various preventive and reactive congestion control schemes.

7.5 Preventive congestion control

As mentioned above, preventive congestion control involves the following two procedures: call admission control (CAC), and bandwidth enforcement. Call admission control is used by the network to decide whether to accept a new connection or not.

As we have seen so far, ATM connections maybe either permanent virtual connections (PVC) or switched virtual connections (SVC). A PVC is established manually by a network administrator using network management procedures, whereas an SVC is established in real-time by the network using the signalling procedures described in Chapters 10 and 11.

In our discussion below we will consider a point-to-point SVC. We recall that point-to-point connections are bi-directional. The traffic and quality of service parameters can be different for each direction of the connection.

Let us assume that an end-device, referred to as end-device 1, wants to set-up a connection to a destination end-device, referred to as end-device 2. A point-to-point SVC is established between the two end-devices, when end-device 1 sends a SETUP message to its ingress switch, referred to as switch A, requesting that a connection is established to end-device 2. The ingress switch calculates a path through the network to the switch to which the destination end-device is attached using a routing algorithm. We shall refer to this switch as B. Then, it forwards the set-up request to its next-hop switch, which in turn forwards it to its next-hop switch, and so on until it reaches switch B. Switch B sends the set-up request to end-device 2, and if it is accepted, a confirmation message is sent back to end-device 1.

The set-up message, as will be seen in Chapter 10, contains a variety of different types of information, including values for the traffic and quality-of-service parameters. This information is used by each switch in the path to decide whether it should accept or reject the new connection. This decision is based on the following two questions:

1. Will the new connection affect the quality-of-service of the existing connections already carried by the switch?
2. Can the switch provide the quality-of-service requested by the new connection?

As an example, let us consider a non-blocking ATM switch with output buffering, shown in figure 6.23, and let us assume that the quality of service is measured by the cell loss

rate. Typically, the traffic that a specific output port sees is a mixture of different connections that enter the switch from different input ports. We assume that so far the switch provides a cell loss probability of 10^{-6} for each existing connection routed through this output port. Now, let us assume that the new connection requests a cell loss rate of 10^{-6}. What the switch has to decide is whether the new cell loss rate for both the existing connections and the new connection will be 10^{-6}. If the answer is yes, then the switch can accept the new connection. Otherwise it will reject it.

Each switch on the path of a new connection has to decide independently of the other switches whether it has enough bandwidth to provide the quality-of-service requested for this connection. This is done using a call admission control algorithm. Various call admission control algorithms are discussed in the following section.

If a switch along the path is unable to accept the new connection, then the switch refuses the set-up request and it sends it back to a switch in the path that can calculate an alternative path.

Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the transmitting source is within its negotiated traffic parameters. Bandwidth enforcement procedures are discussed in section 7.7.

7.6 Call admission control (CAC)

Call admission algorithms (CAC) may be classified into *non-statistical bandwidth allocation (or peak bit rate allocation)* and *statistical bandwidth allocation*. Below, we examine these two classes.

Non-statistical bandwidth allocation

Non-statistical bandwidth allocation, otherwise known as peak bit rate allocation, is used for connections requesting a CBR service. In this case the CAC algorithm is very simple, as the decision to accept or reject a new connection is based purely on whether its peak bit rate is less than the available bandwidth on the link. Let us consider, for example, a non-blocking switch with output buffering, such as the one shown in figure 6.23, and suppose that a new connection with a peak bit rate of 1 Mbps has to be established

through output link 1. Then, the new connection is accepted if the link's available capacity is more or equal to 1 Mbps.

In the case where non-statistical allocation is used for all the connections routed through a link, the sum of the peak bit rates of all the existing connections is less than the link's capacity. Peak bit rate allocation may lead to a grossly underutilized link, unless the connections transmit continuously at peak bit rate.

Statistical bandwidth allocation

In statistical bandwidth allocation, the allocated bandwidth on the output link is less than the peak bit rate of the source. In the case where statistical allocation is used for all the connections on the link, the sum of the peak bit rates of all the connections may exceed the link's capacity. Statistical allocation makes economic sense when dealing with bursty sources, but it is difficult to implement effectively. This is due to the fact that it is not always possible to characterize accurately the traffic generated by a source and how it is modified deep in an ATM network. For instance, let us assume that a source has a maximum burst size of 100 cells. As the cells belonging to the same burst travel through the network, they get buffered in each switch and due to multiplexing with cells from other connections and scheduling priorities, the maximum burst of 100 cells may become much

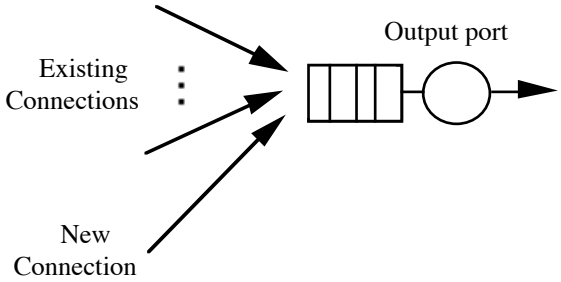


Figure 7.9: An ATM multiplexer

larger deep in the network. Other traffic descriptors, such as the PCR and the SCR, can be similarly modified deep in the network. For instance, let us consider a source with a peak bit rate of 128 Kbps. Due to multiplexing and scheduling priorities, it is possible

that several cells from this source can get batched together in the buffer of an output port of a switch. Now, let us assume that this output port has a speed of, say 1.544 Mbps. Then, these cells will be transmitted out back-to-back at 1.544 Mbps, which will cause the peak bit rate of the source to increase temporarily!

Another difficulty in designing a CAC algorithm for statistical allocation is due to the fact that an SVC has to be set-up in real-time. Therefore, the CAC algorithm cannot be CPU intensive. This problem may not be as important when setting up PVCs. The problem of whether to accept or reject a new connection may be formulated as a queueing problem. For instance, let us consider again our non-blocking switch with output buffering. The CAC algorithm has to be applied to each output port. If we isolate an output port and its buffer from the switch, we will obtain the queueing model shown in figure 7.9. This type of queueing structure is known as the *ATM multiplexer*. It represents a number of ATM sources feeding a finite-capacity queue which is served by a server, i.e., the output port. The service time is constant and it is equal to the time it takes to transmit an ATM cell. (The simulation project described at the end of this Chapter deals with an ATM multiplexer.)

Now let us assume that the quality of service, expressed in cell loss rate, of the existing connections is satisfied. The question that arises is whether the cell loss rate will still be maintained if the new connection is accepted. This can be answered by solving the ATM multiplexer queueing model with the existing connections and the new connection. However, the solution to this problem is CPU intensive and it cannot be done in real-time. In view of this, a variety of different call admission control algorithms have been proposed which do not require the solution of such a queueing model.

Most of the call admission control algorithms that have been proposed are based solely on the cell loss rate QoS parameter. That is, the decision to accept or reject a new connection is based on whether the switch can provide the new connection with the requested cell loss rate without affecting the cell loss rate of the existing connections. No other QoS parameters, such as peak-to-peak cell delay variation and the max CTD, are considered by these algorithms. A very popular example of this type of algorithm is the *equivalent bandwidth*, described below.

Call admission control algorithms based on the cell transfer delay have also been proposed. In these algorithms, the decision to accept or reject a new connection is based on a calculated absolute upper bound of the end-to-end delay of a cell. These algorithms are closely associated with specific scheduling mechanisms, such as static priorities, early-deadline-first, and weighted fair queueing. Given that the same scheduling algorithm runs on all the switches in the path of a connection, it is possible to construct an upper bound of the end-to-end delay. If this is less than the requested end-to-end delay, then the new connection is accepted.

Below, we examine the equivalent bandwidth scheme and then we present the ATM block transfer (ABT) scheme used for bursty sources. In this scheme, bandwidth is allocated on demand and only for the duration of a burst. Finally, we present a scheme for controlling the amount of traffic in an ATM network based on *virtual path connections* (VPC).

7.6.1 Equivalent bandwidth

Let us consider a finite capacity queue served by a server at the rate of μ. This queue can be seen as representing an output port and its buffer in a non-blocking switch with output buffering. We assume that this queue is fed by a single source, whose equivalent bandwidth we wish to calculate. Now, if we set μ equal to the source's peak bit rate, then we will observe no accumulation of cells in the buffer. This is because the cells arrive as fast as they are transmitted out. Now, if we slightly reduce the service rate μ, then we will see that cells are beginning to accumulate in the buffer. If we reduce the service rate a little bit more, then the buffer occupancy will increase. If we keep repeating this experiment and each time we lower slightly the service rate, then we will see that the cell loss rate begins to increase. The equivalent bandwidth of the source is defined as the service rate e at which the queue is served that corresponds to a cell loss rate of ε. The equivalent bandwidth of a source falls somewhere between its average bit rate and its peak bit rate. If the source is very bursty, it is closer to its peak bit rate, otherwise, it is closer to its average bit rate. We note that the equivalent bandwidth of a source is not related the source's SCR.

There are various approximations that can be used to compute quickly the equivalent bandwidth of a source. A commonly used approximation is based on the assumption that the source is an interrupted fluid process (IFP) characterized by the triplet (R,r,b), where R is its peak bit rate, r the fraction of time the source is active, defined as the ratio of the mean length of the on period divided by the sum of the mean on and off periods, and b the mean duration of the on period. Let us now assume that the source feeds a finite-capacity queue with a constant service time, and let K be the size of the queue expressed in bits. The service time is equal to the time it takes to transmit out a cell. Then, the equivalent bandwidth e is given by the expression:

$$e = \frac{a - K + \sqrt{(a - K)^2 + 4Kar}}{2a} R \qquad (7.1)$$

where $a = b(1-r)R \ln(1/\varepsilon)$.

The equivalent bandwidth of a source is used in statistical bandwidth allocation in the same way that the peak bit rate is used in non-statistical bandwidth allocation. For instance, let us consider an output link of a non-blocking switch with output buffering, and let us assume that it has a transmission speed of 25 Mbps and its associated buffer has a capacity of 200 cells. We assume that no connections are currently routed through the link. The first set-up request that arrives is for a connection that requires an equivalent bandwidth of 5 Mbps. The connection is accepted and the link has now 20 Mbps available. The second set-up request arrives during the time that the first connection is still up and it is for a connection that requires 10 Mbps. The connection is accepted and 10 Mbps are reserved, leaving 10 Mbps free. If the next set-up request arrives during the time that the first two connection are still up and it is for a connection that requires more than 10 Mbps, then the new connection is rejected.

This method of simply adding up the equivalent bandwidth requested by each connection may lead to under-utilization of the link. That is, more bandwidth may be allocated for all the connections than it is necessary. The following approximation for the equivalent bandwidth of N sources corrects the over-allocation problem:

$$c = \min \{\rho + \sigma\sqrt{-2\ln(\varepsilon) - \ln 2\pi}\ ,\ \sum_{i=1}^{N} e_i\ \} \qquad (7.2)$$

where ρ is the average bit rate of all the sources, e_i is the equivalent bandwidth of the ith source, calculated using expression (7.1), and σ is the sum of the standard deviation of the bit rate of all the sources and it is equal to

$$\sigma = \sum_{i=1}^{N} \sqrt{r_i (R_i - r_i)}\ .$$

When a new set-up request arrives, the equivalent bandwidth for all the existing connections and the new one is calculated using expression (7.2). The new connection is accepted if the resulting bandwidth c is less than the link's capacity.

Below we demonstrate through a numerical example how the maximum number of connections admitted using the above expressions for the equivalent bandwidth varies with the buffer size K, the cell loss rate ε, and the fraction of time the source is active r. We consider a link with a transmission speed C equal to 150 Mbps and a buffer capacity of K cells. Each connection is characterized by the parameters R, its peak bit rate, ρ the average bit rate, and b the mean duration of the on period. We note that the quantity r defined above is related to ρ through the expression $rR = \rho$. In the numerical examples presented below, we assume that all connections are identical with traffic parameters (R,ρ,b) = (10 Mbps, 1 Mbps, 310 cells).

We note that if we admit connections using their peak bit rate, then a maximum of 150 Mbps/10 Mbps = 15 connections will be admitted. On the other hand, if we admit connections using the average bit rate, a maximum of 150 Mbps/1 Mbps = 150 connections will be admitted. These two values can be seen as an upper and lower bound on the number of connections that can be admitted using the equivalent bandwidth method.

In figure 7.10, the maximum number of connections that can be admitted is plotted as a function of the buffer size K. The buffer size was increased from 31 cells to 31,000,

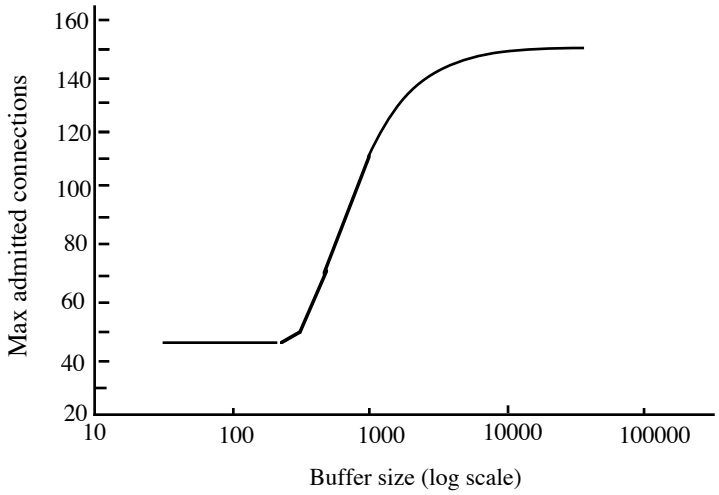


Figure 7.10: Varying the buiffer size K

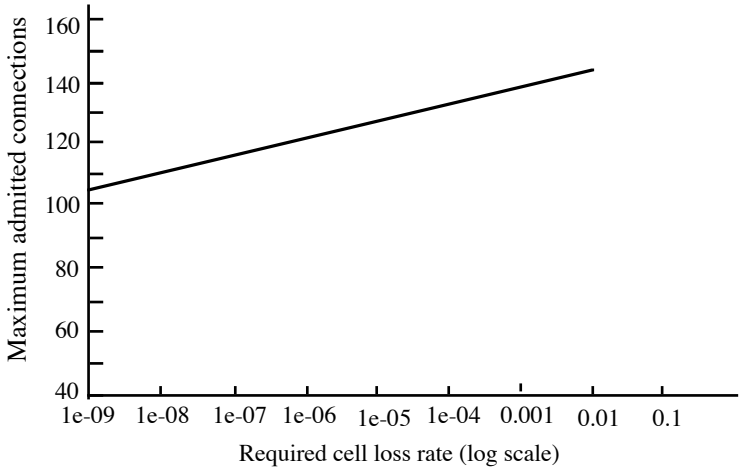


Figure 7.11: Varying the required cell loss rate ε

cells. For each value of K, the maximum number of admitted connections was obtained using expressions (7.1) and (7.2) with the cell loss rate fixed to 10^{-6}. We observe that for small values of K, the maximum number of connections admitted by the equivalent bandwidth algorithm is constant. As K increases, the maximum number of admitted connections increases as well, and eventually flattens out.

In figure 7.11, the maximum number of admitted connections is plotted against the cell loss rate ε. The buffer size was fixed to 1236 cells, i.e.,64 Kbytes. We observe that the maximum number of admitted connections is not very sensitive to the cell loss

rate ε. In this particular example, the buffer size is large enough so that the equivalent algorithm admits a large number of connections. In general, the equivalent bandwidth algorithm becomes more sensitive to ε when the buffer size is smaller.

Finally, in figure 7.12, the maximum number of admitted connections is plotted against r, the fraction of time that a source is active, where r = ρ/R. We recall from section 7.1.2, that r can be used to express the burstiness of a source. The buffer size was fixed to 1236 cells and the cell loss rate ε to 10^{-6}. We observe that the maximum number of admitted connections dependents on r. As r increases, the source becomes more bursty and requires more buffer space in order to maintain the same cell loss rate. As a result the maximum number of admitted connections falls sharply as r tends to 0.5.

7.6.2 The ATM block transfer (ABT) scheme

A number of congestion control schemes were devised for bursty sources whereby each switch allocates bandwidth on demand and only for the duration of a burst. The main idea

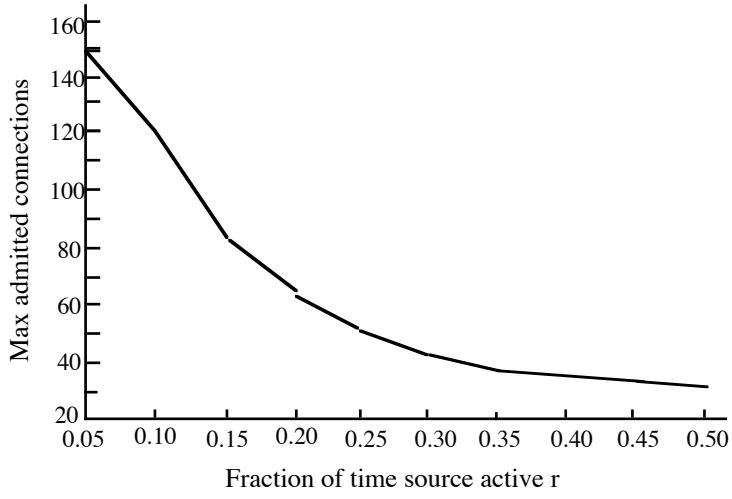


Figure 7.12: Varying r

behind these schemes is the following. At connection set-up time, the path through the ATM network is selected, and each switch in the path allocates the necessary VPI/VCI labels and updates the switching table used for label swapping in the usual way. However, it does not allocate any bandwidth to this connection. When the source is ready

to transmit a burst, it notifies the switches along the path, and it is at that moment that each switch will allocate the necessary bandwidth for the duration of the burst.

These congestion control schemes are known as *fast bandwidth allocation* schemes. The ATM block transfer (ABT) scheme is a fast bandwidth allocation scheme and it is a standardized ATM transfer capability. ABT makes use only of the peak bit rate and it is intended for VBR sources whose peak bit rate is less than 2% of the link's capacity.

In ABT, a source requests bandwidth in incremental and decremental steps. The total requested bandwidth for each connection may vary between zero and its peak bit rate. For a step increase, a source uses a special reservation request cell. If the requested increase is accepted by all the switches in the path, then the source can transmit at the higher bit rate. If the step increase is denied by a switch in the path, then the step increase request is rejected. Step decreases are announced through a management cell. A step decrease is always accepted. At the cell level, the incoming cell stream of a source is shaped, so that the enforced peak bit rate corresponds to the currently accepted peak bit rate.

A fast reservation protocol (FRP) unit was implemented to handle the relevant management cells. This unit is located at the UNI. The protocol utilizes different timers to ensure its reliable operation. The end-device utilizes a timer to ensure that its management cells, such as step increase requests, sent to its local FRP unit are not lost. When the FRP unit receives a step increase request, it forwards the request to the first switch in the path, which in turn forwards it to the next-hop switch, and so on. If the request can be satisfied by all the switches on the path, then the last switch will send an ACK to the FRP unit. The FRP unit then informs the end-device that the request has been accepted, updates the policing function, and sends a validation cell to the switches in the path to confirm the reservation. If the request cannot be satisfied by a switch, the switch simply discards the request. The upstream switches, which have already reserved bandwidth, will discard the reservation if they do not receive the validation cell by the time a timer expires. This timer is set equal to the maximum round trip delay between the FRP unit and the furthermost switch. If the request is blocked, the FRP unit will re-try to

request the step increase after a period set by another timer. The number of attempts is limited.

This mechanism can be used by an end-device to transmit bursts. When the end-device is ready to transmit a burst, it issues a step increase request with a requested bandwidth equal to its peak bit rate. If the request is granted, the end-device transmits its burst, and at the end it announces a step decrease with bandwidth equal to its peak bit rate.

In a slightly different version of the ABT protocol, the end-device starts transmitting its burst immediately after it issues a reservation request. The advantage of this scheme is that the end-device does not have to wait until the request is granted. The burst will get lost if a switch in the path is unable to accommodate the request.

7.6.3 Virtual path connections

A virtual path connection can be used in an ATM network to create a dedicated connection between two switches. Within this connection, individual virtual circuit connections can be set-up, without the knowledge of the network.

Let us assume, for instance, that a permanent virtual path connection is established between two switches, namely 1 and 2. These two switches may not be adjacent and they may communicate through several other switches. A fixed amount of bandwidth is allocated to the virtual path connection. This bandwidth is reserved for this particular connection and it cannot be shared with other connections, even when it is not used entirely. An end-device attached to switch 1 and wishing to communicate to an end-device attached to switch 2, is allocated part of the bandwidth of the virtual path connection using non-statistical or statistical bandwidth allocation. The connection is rejected if there is not enough bandwidth available within the virtual path connection, since the total amount of traffic carried by this virtual path connection cannot exceed its allocated bandwidth.

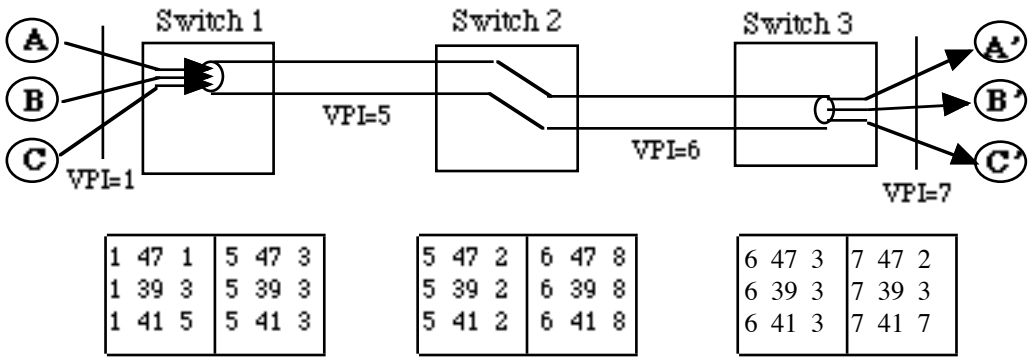


Figure 7.13: Label swapping in a virtual path connection

A virtual channel connection maintains the same VCI value through out an entire virtual path connection. That is, its VCI value has global significance. The virtual path, however, is identified by a series of VPI values, each having local significance.

An example of label swapping in a virtual path connection is given in figure 7.13. A virtual path connection has been established between switches 1 and 3. Users A, B, and C are attached to switch 1, and via the virtual path connection, they are connected to their respective destinations A', B', and C', which are attached to switch 3. Each switch is represented by a square, and its switching table is given immediately below the square. We assume that the switching table is centralized and it contains information for all input ports. The first three columns in the switching table give the VPI/VCI of each incoming connection and its input port. The second three columns give the new label and the destination output port of the connection. Also, the first row of each switching table is associated with the connection from A to A', the second row is associated with the connection from B to B', and the third row is associated with the connection from C to C'. We observe, that the virtual path connection has a VPI=1 on the UNI between users A, B, C and switch 1, a VPI=5 on the hop from switch 1 to switch 2, a VPI=6 on the hop from switch 2 to switch 3, and a VPI=7 on the UNI between switch 3 and users A', B', and C'. The virtual channel connections from A to A', B to B', and C to C' are identified by the VCIs 47, 39, and 41 respectively.

Virtual path connections provide a network operator with a useful mechanism. For instance, it can be used to provide a customer with a dedicated connection between two locations. Within this connection, the customer can set-up any number of virtual

circuit connections, as long as the total bandwidth allocated to the virtual path connection is not exceeded.

Virtual path connections can be combined to form a virtual network overlaid on an ATM network. Such a virtual network can be set-up by a network operator in order to control the amount of traffic in the network. In addition, the network operator can set-up different virtual networks for different ATM service categories.

7.7 Bandwidth enforcement

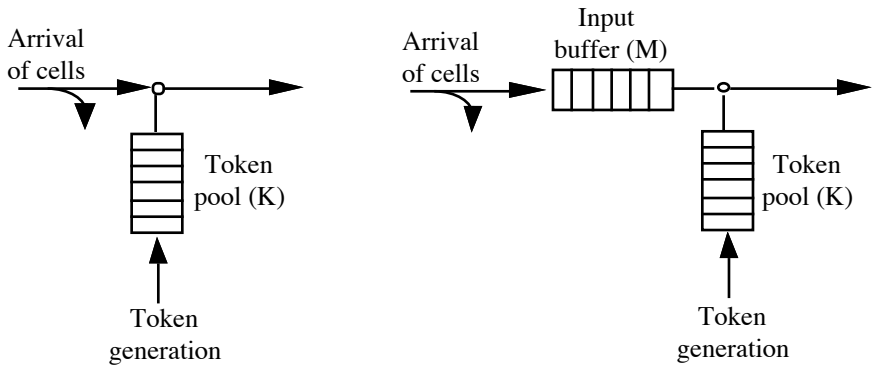
The function of bandwidth enforcement is to ensure that the traffic generated by a source conforms with the *traffic contract* that was agreed upon between the user and the network at call set-up time. According to the ITU-T and the ATM Forum, the traffic contract consists of (1) the traffic parameters, (2) the requested quality of service parameters, and (3) a definition of conformance. The traffic and the quality-of-service parameters, as we have seen, depend upon the requested service category.

Testing the conformance of a source, otherwise known as policing the source, is carried out at the user-network interface (UNI). It involves policing the peak cell rate and the sustained cell rate using the *generic cell rate algorithm* (GCRA). ITU-T first standardized this algorithm for the peak cell rate. The ATM Forum adapted the same algorithm, and it also extended it for testing the conformance of the sustained cell rate. It is possible that multiple GCRAs can be used in series, such as one for the peak cell rate and another one for the sustained cell rate.

Policing each source is an important function from the point of view of a network operator, since a source exceeding its contract may affect the quality-of-service of other existing connections. Also, depending upon the pricing scheme used by the network operator, there may be loss of revenue. A source may exceed its contract due to various reasons, such as, intentional or unintentional underestimation by the user of the required bandwidth, and malfunctioning of user's equipment.

The generic cell rate algorithm is based on a popular policing mechanism known as the *leaky bucket*. The leaky bucket may be *unbuffered* or *buffered*. The unbuffered leaky bucket consists of a token pool of size K, as shown in figure 7.14 (a). Tokens are

generated at a fixed rate. A token is lost if it is generated at a time when the token pool is full. An arriving cell takes a token from the token pool, and then enters the network. The number of tokens in the token pool is then reduced by one. A cell is considered to be a *violating* cell (or, a *non-compliant* cell), if it arrives at a time when the token pool is empty. The buffered leaky bucket is shown in figure 7.14 (b). It is the same as the unbuffered leaky bucket with the addition of an input buffer of size M, where a cell can



(a) Unbuferred leaky bucket **(b) Buferred leaky bucket**

Figure 7.14: The leaky bucket

wait if it arrives at a time when the token pool is empty. A cell is considered to be a violating cell, if it arrives at a time when the input buffer is full. Violating cells are either dropped or tagged (see section 7.7.2).

A leaky bucket is completely defined by its parameters: K, token generation rate and M, if it is a buffered leaky bucket. The difficulty with the leaky bucket is in fixing its parameters, so that it is transparent when the source adheres to its contract, and it catches all the violating cells when the source exceeds its contract. Given a probabilistic model of an arrival process of cells to the UNI, it is possible to fix the parameters of the leaky bucket using queueing-based models. However, it has been shown that the leaky bucket can be very ineffective in catching violating cells. Dual leaky buckets have been suggested for more efficient policing, where the first leaky bucket polices violations of the peak cell rate, and the second one polices violations of the source's burstiness. As will be seen below, GCRA does catch all violating cells, but to do that it needs an additional traffic parameter.

In addition to GCRA, a source can shape its traffic using a *traffic shaper* in order to attain desired characteristics for the stream of cells it transmits to the network. Traffic shaping involves peak cell rate reduction, burst size reduction, and reduction of cell clumping by suitably spacing out the cells in time.

7.7.1 The generic cell rate algorithm (GCRA)

Unlike the leaky bucket mechanism, GCRA is a deterministic algorithm and it does catch all the violating cells. However, for this it requires an additional new traffic parameter known as the *cell delay variation tolerance* (CDVT). This parameter is not to be confused with the peak-to-peak cell delay variation parameter described in section 7.2.

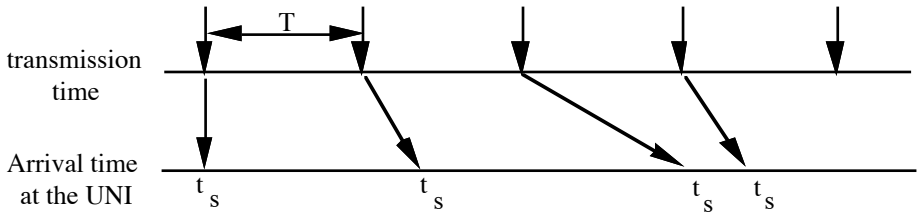


Figure 7.15: Arrival times at the UNI

Let us assume that a source is transmitting at peak cell rate and it produces a cell every T units of time, where T = 1/PCR. As shown in figure 7.15, due to multiplexing with cells from other sources and with signalling and network management cells, it is possible that the inter-arrival time of successive cells belonging to the same source at the UNI may vary around T. That is, for some cells it may be greater than T, and for others it may be less than T. In the former case, there is no penalty in arriving late! However, in the latter case, the cells will appear to the UNI that they were transmitted at a higher rate, even though they were transmitted conformally to the peak cell rate. In this case, these cells should not be penalized by the network. The cell delay variation tolerance is a parameter that permits the network to tolerate a number of cells arriving at a rate which is faster than the agreed upon peak cell rate. This parameter does not depend upon a particular source. Rather, it depends on the number of sources that use the same UNI and the access to the UNI, and it is specified by a network administrator.

GCRA can be used to monitor the peak cell rate and the sustained cell rate. There are two implementations of GCRA, namely, the *virtual scheduling algorithm* and the *continuous-state leaky bucket* algorithm. These two algorithms are equivalent to each other.

Policing the peak cell rate

In order to monitor the peak cell rate, the following two parameters are required: peak emission interval T and cell delay variation tolerance τ. T = 1/PCR, and it is obtained from the user's declared peak cell rate, and as mentioned above, τ is provided by a network administrator.

A flow-chart of the virtual scheduling algorithm is shown in figure 7.16. Variable TAT is the theoretical arrival time of a cell and t_s is the actual arrival time of a cell. At the time of arrival of the first cell, TAT=t_s. Each time a cell arrives, the algorithm calculates the theoretical time TAT of the next arrival. If the next cell arrives late, that is if TAT<t_s, then

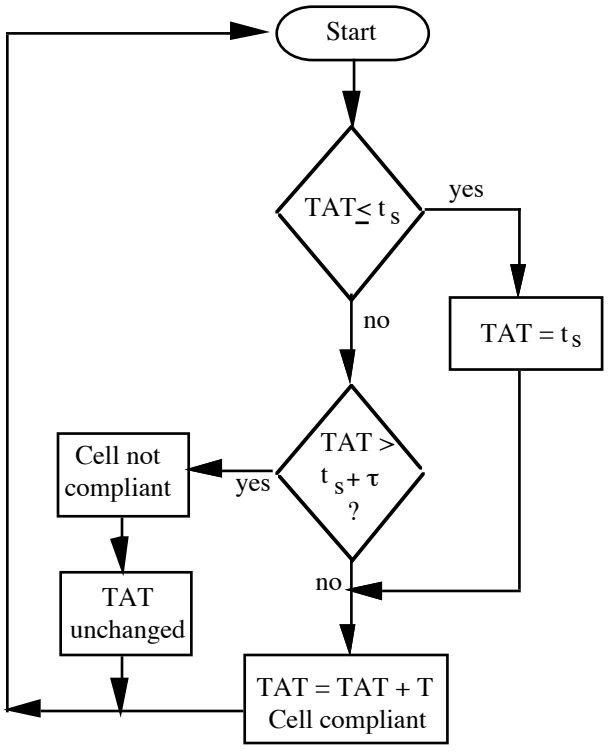


Figure 7.16: The virtual scheduling algorithm

the next theoretical arrival time is set to TAT = t_s+T. If the next arrival is early, that is TAT>t_s, then the cell may be accepted or it may be classified as non-compliant. The decision is based on the cell delay variation tolerance τ and also on previously arrived cells that were late but they were accepted as compliant. Specifically, if TAT<t_s+τ, then the cell is considered as compliant. Notice however, that the next theoretical arrival time TAT is set equal to the theoretical arrival time of the current cell plus T, that is TAT = TAT+T. If the next arrival occurs before the theoretical arrival time TAT, it may still be accepted if TAT<t_s+ τ. However, if cells continue to arrive early, the cell delay variation will be used up and eventually a cell will be classified as non-conformant.

As an example, let us consider the case where T=10, τ=15, and the actual arrival times of the first five cells are: 0, 12, 18, 20 and 25. For cell 1 we have that t_s=TAT=0. The cell is accepted and TAT is set to TAT+10=10. For cell 2, t_s=12 and since TAT≤t_s, the cell is accepted and TAT is set equal to t_s+T=22. Cell 3 arrives at time t_s = 18 and in view of this TAT>t_s. Since TAT≤t_s+τ the cell is accepted and TAT is set equal to TAT+T=32. Cell 4 arrives at time t_s=20, and TAT>t_s. Since TAT≤t_s+τ the cell is accepted and TAT is set equal to TAT+T=42. Cell 5 is not as lucky as cells 3 and 4. Its arrival time is t_s=25 which makes TAT>t_s. Since TAT> t_s+τ the cell is considered as non-compliant.

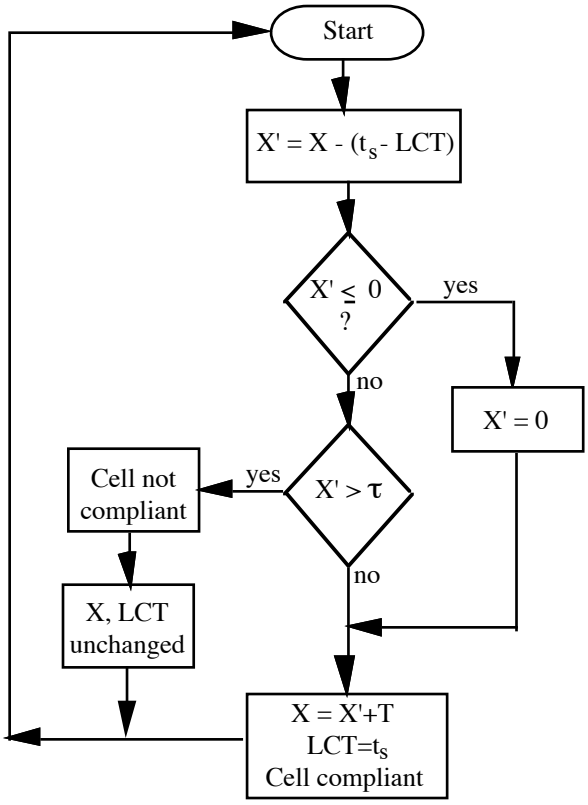


Figure 7.17: Continuous state leaky bucket algorithm

A flow-chart of the continuous state leaky bucket algorithm is shown in figure 7.17. In this algorithm, a finite-capacity leaky bucket is implemented whose real-value content is drained out at a continuous rate of 1 unit of content per unit-time. Its content is increased by a fixed increment T each time a conforming cell arrives. The algorithm makes use of the variables X, X′, and LCT. X indicates the current value of the leaky bucket, X′ is an auxiliary variable, and LCT is the last compliance time, that is the last time a compliant cell arrived. At the arrival time of the first cell, X=0 and LCT=t_s.

When a cell arrives, the quantity X-(t_s-LCT) is calculated and saved in the auxiliary variable X′. If X′≤0, then the cell has arrived late and the cell is accepted, X is increased by T, and LCT=t_s. If X′>0, then depending upon whether X′ is less or greater than τ, the cell is accepted or it is considered as non-compliant. If X>τ, the cell is classified as non-compliant and the values of X and LCT remain unchanged. If X≤τ, then the cell is accepted, X is set to X′+T, and LCT=t_s.

Let us now consider the same example as in the virtual scheduling algorithm, that is T=10, τ=15, and the actual arrival times of the first five cells are: 0, 12, 18, 20 and 25. For cell 1 we have that X=0 and LCT=0. The cell is accepted and X is set to 10 and LCT to 0. For cell 2, we have X′=-2. The cell is accepted and X is set to 10 and LCT to 12. Cell 3 arrives at time 18, which gives a value for X′ equal to 4. Since X′<τ, the cell is accepted and X is set to X′+T=14 and LCT to 18. Cell 4 arrives at time 20, and we have X′=12. Since X′<τ, the cell is accepted and X is set to X′+T=22 and LCT to 20. Cell 5 arrives at time 25, and we have that X′=17. Since X′>τ, the cell is classified as non-complaint and X and LCT remain unchanged.

Policing the sustained cell rate

The sustained cell rate of a source is policed by either GCRA algorithm. As we saw above, GCRA uses the parameters T and τ in order to police the peak cell rate of a source. For policing the sustained cell rate of a source, it uses the parameters T_s and τ_s. T_s is the emission interval when the source transmits at its sustained cell rate, and it is equal to 1/SCR. τ_s is known as the *burst tolerance* (BT) and it is calculated from the maximum burst size (MBS) provided by the source using the expression:

$$\tau_s = (MBS-1)(T_s-T).$$

If the inter-arrival time of cells is equal to or greater than T_s, then the cells are compliant. However, some cells may arrive every T units of time, where T<T_s, if they are transmitted at peak cell rate. Since these cells arrive every T units of time, they are in essence non-compliant as far as GCRA is concerned. How many such cells should GCRA tolerate, before it starts classifying them as non-compliant? Obviously, the maximum number of cells that can arrive every T units of time is equal to the source's MBS minus the first cell that initiates the burst. That is, we expect a maximum of (MBS-1) cells to arrive (T_s-T) units of time faster. This gives a total time of (MBS-1)(T_s-T), which is the burst tolerance τ_s. In conformance testing, τ_s is set equal to:

$$\tau_s = (MBS-1)(T_s-T) + CDVT.$$

7.7.2 Packet discard schemes

As we saw in the previous section, GCRA will either accept a cell or classify it as non-compliant. The question, therefore, that arises is what to do with non-compliant cells. The simplest scheme is to just drop them. A more popular mechanism, known as *violation tagging*, attempts to carry the non-compliant cells if there is slack capacity in the network. The violating cells are tagged at the UNI and then they are allowed to enter the network. If congestion arises inside the network the tagged cells are dropped. Tagging of a cell is done using the cell loss priority (CLP) bit in the cell's header. If the cell is untagged, then its CLP=0. When a cell is tagged, its CLP=1.

Violation tagging introduces two types of cells: the untagged cell and the tagged cell. A simple way to handle tagged cells is through a priority mechanism, such as the *push-out* scheme and the *threshold* scheme. In the push-out scheme, both untagged and tagged cells are freely admitted into a buffer as long as the buffer is not full. If a tagged cell arrives during the time that the buffer is full, the cell is lost. If an untagged cell arrives during the time that the buffer is full, the cell will take the space of the last arrived tagged cell. The untagged cell will get lost if all the cells in the buffer are untagged. In the threshold scheme, both untagged and tagged cells are admitted as long as the total number of cells is below a threshold. Over the threshold, only untagged cells are admitted, and the tagged cells are rejected. The push-out priority scheme is more efficient than the threshold priority scheme, but the latter is preferable because it is simpler to implement. Other priority mechanisms have also been proposed such as dropping from the front. This mechanism is similar to the threshold mechanism, only cells are dropped from the front. That is, when a tagged cell is ready to begin its service, the total number of cells in the buffer is compared against the threshold. If it is below, service begins, else the cell is dropped.

A discarded cell may be part of a user packet, such as a TCP packet. In this case, the receiving TCP will detect that the packet is corrupted and it will request the sending TCP to retransmit it. In view of this, when discarding a cell we can save bandwidth by discarding the subsequent cells that belong to the same user packet since the entire packet will have to be retransmitted anyway. For applications using AAL 5, it is possible to

identify the beginning and the end of each user packet, and consequently drop the subsequent cells that belong to the same packet. There are two such discard mechanisms, namely *partial packet discard* (PPD) and *early packet discard* (EPD). Partial packet discard can be applied when the discarded cell is not the first cell of an AAL 5 frame. In this case, all subsequent cells belonging to the same AAL 5 frame are discarded except the last cell. This cell has to be kept so that the destination can determine the end of the AAL 5 frame. Early packet discard can be applied when the discarded cell is the first cell of an AAL 5 frame. In this case, all cells belonging to the same frame, including the last one, are discarded.

7.8. Reactive congestion control

Reactive congestion control is based on a different philosophy to the one used in preventive congestion control. In preventive congestion control we attempt to prevent congestion from occurring. This is done by first reserving bandwidth for a connection on each switch along the connection's path, and subsequently policing the amount of traffic transmitted on the connection. In reactive congestion control, at least in its ideal form, we let sources transmit without bandwidth reservation and policing, and we take action only when congestion occurs. The network is continuously monitored for congestion. If congestion begins to build up, a feedback message is sent back to each source requesting them to slow down or even stop. Subsequent feedback messages permit the sources to increase their transmission rates. Typically, congestion is measured by the occupancy level of critical buffers within an ATM switch, such as the output port buffers in a non-blocking switch with output buffering.

The available bit rate (ABR) service, described below, is the only standardized ATM service category that uses a reactive congestion control scheme.

7.8.1 The available bit rate (ABR) service

This is a feedback-based mechanism whereby the sending end-device is allowed to transmit more during the time that there is a slack in the network. At connection set-up time, the sending end-device requests a *minimum cell rate* (MCR). It also specifies a

maximum cell rate, which is its PCR. The network accepts the new connection if it can satisfy its requested MCR. We note that the MCR may also be zero. The transmission rate of the source may exceed its requested MCR, if the network has a slack capacity. When congestion begins to build up in the network, the sending end-device is requested to decrease its transmission rate. However, its transmission rate will never drop below its MCR. The ABR service is not intended to support real-time applications.

It is expected that the sending end-device is capable of increasing or decreasing its transmission rate according to the feedback messages it receives from the network. Also, it is expected that the sending end-device that conforms to the feedback messages received by the network, will experience a low cell loss rate and it will obtain a fair share of the available bandwidth within the network.

The control mechanism through which the network can inform the source to change its transmission rate is implemented using *resource management* (RM) cells. These are ATM cells whose payload type indicator (PTI) is set to 110 (see table 4.2). As shown in figure 7.18, the transmitting end-device generates *forward* RM cells which travel through the network to the receiver following the same path as its data cells. The receiver turns around these RM cells and transmits them back to the sending end-device as

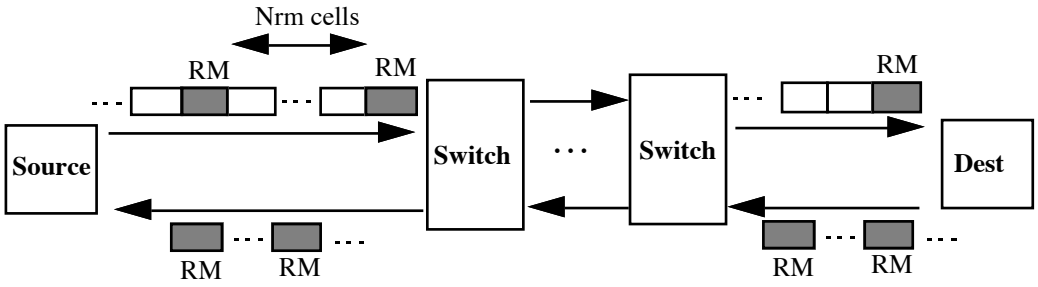


Figure 7.18: The ABR mechanism

backward RM cells. These backward RM cells follow the opposite path to the sending end-device. (We recall that point-to-point connections are bidirectional.) ATM switches along the path of the connection, may insert feedback control information in the RM cells, which is used by the sending end-device to increase or decrease its transmission

rate. Thus, a closed control loop is formed between the sending end-device and its destination end-device. This closed loop is used to regulate the transmission rate of the sending end-device. A similar loop can be set-up to regulate the transmission rate of the destination end-device.

We note that feedback messages may not be effective when dealing with a link which has a long propagation delay. This is because the link may become a temporary storage capable of holding a large number of cells. As an example, let us consider a link which is 125 miles long connecting an end-device to a switch, and let us assume that the end-device transmits at 622 Mbps. This transmission speed translates to about 1462 cells per msec. Since light propagates through a fiber link at approximately 125 miles per msec, a maximum of 1462 cells may be in process of being propagated along the link. Let now assume that at time t the switch sends a message to the end-device requesting it to stop transmitting. Then, by the time the end-device receives the message, the switch is likely to receive a maximum of 2x1462 cells. Of these cells, a maximum of 1462 cells can be already in flight at time t, and another maximum of 1462 cells can be transmitted by the time the end-device receives the message. In order to account for large propagation delays, manufacturers have introduced large buffers in their switch architectures. In addition, several feedback loops can be set-up, as shown in figure 7.19, aiming at reducing the length of the control loop.

The ABR service does not include a formal conformance definition. However, verification that the source complies can be done using a dynamic GCRA, where the monitored transmission rate is modified based on the receipt of backwards RM cells.

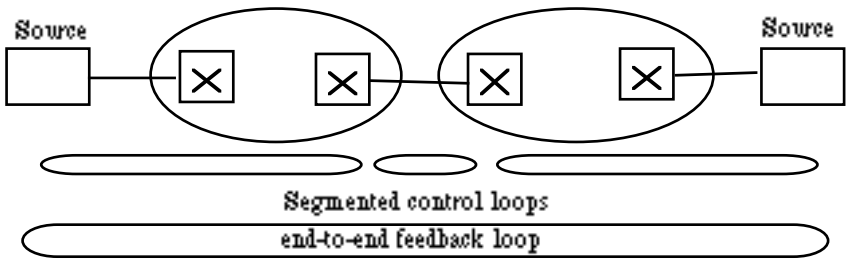


Figure 7.19: Feedback loops

RM cell structure

The RM cell's payload contains a number of different fields. Below, we describe some of these fields.

Message type field: This is a one-byte field and it contains the following 1-bit subfields:

DIR: This bit indicates the direction of the RM cell, i.e. whether it is a forward or a backward RM-cell.

BN: This bit indicates whether the RM cell is a *backward explicit congestion notification* (BECN) cell. As will be seen later on, an ATM switch or the destination end-device may independently generate a backward RM cell in order to notify the sending end-device, instead of having to wait for an RM cell generated by the sending end-device to come by. This RM cell has its BN bit set to 1. RM cells generated by the source, have their BN field set to zero.

CI: This congestion indication bit is used to by an ATM switch or the destination end-device, to indicate to the sending end-device that congestion has occurred in the network.

NI: No increase indicator, used to prevent the sending end-device from increasing its *allowed cell rate* (ACR), which is its current transmission rate.

Explicit rate (ER): This is a 2-byte field used to carry the explicit rate calculated by an ATM switch along the path. The ER is used to limit the sending end-device's transmission rate. This field may be subsequently reduced by another ATM switch, if it calculates an ER which is lower than the one indicated in the ER field of the RM cell.

Current cell rate (CCR): A 2-byte field used by the sending end-device to indicate its ACR, i.e. its current transmission rate.

Minimum cell rate (MCR): This the minimum cell rate that the connection has requested and the network has agreed to guarantee.

The ABR mechanism

The source sends an RM cell every Nrm-1 data cells. The defaulted value for Nrm is 32. The RM cells and data cells may traverse a number of switches before they reach their destination end-device. The destination turns around the RM cells, which become backward RM cells, and transmits them back to the sending end-device. Each switch writes information about its congestion status onto the backward RM cells, which eventually reach the sending end-device. The feedback information send to the source depends on the mode of the ABR scheme. There are two modes, namely the *binary mode* and the *explicit rate mode*.

In the binary mode, the switch marks the EFCN bit in the header of the data cells to indicate pending congestion. (We recall that the EFCN bit is one of the three bits defined in the payload type indicator of the cell header.) The destination translates the EFCN information into bits such as the CI or NI, which are marked in the corresponding backward RM cell. Upon receipt of this, the source takes appropriate action. The action is a) increase the transmission rate, b) decrease the transmission rate, or c) no change to the transmission rate. This mode is used to provide backward compatibility with ATM switches that conformed to earlier standards.

In the explicit rate mode, a switch computes a local fair share for the connection and marks the rate at which the source is allowed to transmit in the ER field of the backward RM cell. The switch does that only if the bandwidth it can offer to the connection is lower than what it is already marked in the backwards RM cell. The source, upon receipt of the backward RM cell, extracts the ER field and sets its transmission rate to the ER value. When detecting congestion, a switch can generate a backwards RM cell in order to convey the congestion status, without having to wait for a backwards RM cell to arrive.

Source behaviour

The source is responsible for inserting an RM cell every Nrm-1 data cells. These RM cells are part of the source's allowed cell rate (ACR). If the source does not have enough data cells to send, an RM cell is generated after a timer has expired and Mrm data cells have been transmitted. Mrm is fixed to 2. The data cells are sent with EFCN=0.

The source adjusts its ACR according to the information received in an RM cell. ACR is greater than or equal to MCR and less than or equal to PCR. The ACR is adjusted as follows:

a. If CI=1, then the ACR is reduced by at least ACRxRDF, where RDF is a pre-specified *rate decrease factor*. If the reduction results to a value below the MCR, then the ACR is set equal to the MCR.
b. If the backward RM cell has both CI=0 and NI=0, then the ACR may be increased by no more than RIFxPCR, where RIF is a pre-specified *rate increase factor*. The resulting ACR should not exceed the source's PCR.
c. If the backward RM cell has NI=1, then the ACR is not increased.

After ACR has been adjusted as above, it is set to at most the minimum of ACR as computed above and to the ER field, but no lower than MCR.

Destination behaviour

When a data cell is received, its EFCN is saved in the EFCN status of the connection. Upon receiving a forward RM cell, the destination turns around the cell and transmits it back to the source. The DIR bit is changed from forward to backward, BN=0, and the fields CCR, MCR, ER, CI, and NI in the RM cell remain unchanged, except in the following cases:

a. If the saved EFCN status of the connection is set, then the destination sets CI=1 in the RM cell, and re-sets the EFCN state.
b. If the destination is experiencing internal congestion, it may reduce the ER to whatever rate it can support and set either CI=1 or NI=1.

The destination may also generate a new backward RM cell, with CI=1 or NI=1, DIR=1, and BN=1. This permits the destination to send feedback information to the source without having to wait for a source-generated RM cell to come by. The rate of these backwards RM cells is limited to 10 cells/sec.

Switch behaviour

At least one of the following methods is implemented in a switch:

a. *EFCN marking:* The switch may set the EFCN bit in the header of the data cells.
b. *Relative rate marking:* The switch may set CI=1 or NI=1 in forward and/or backward RM cells.
c. *Explicit rate marking:* The switch may reduce the ER field of forward and/or backward RM-cells.

The first two marking methods are part of the binary mode, whereas the third one is for the explicit rate mode. The term binary is used because the switch provides information of the type: congestion/no congestion.

A switch may generate backwards RM cells, in order to send feedback information to the source, without having to wait for a source-generated RM cell. The rate of these backwards RM cells is limited to 10 cells/sec. Its fields are marked as follows: CI=1 or NI=1, BN=1, DIR=1.

A switch may also segment the ABR closed loop using a virtual source and destination. This can be useful in cases where the loop between the source and destination involves many hops, or long haul links with a large propagation delay. In such a case, the time it takes for the RM cells to return to the source may be significant. This may impact the time required for the source to react to an RM cell.

The calculation of the ER has to be done in such a way so that the available bandwidth in the switch has to be shared fairly among all the competing ABR connections. A number of different algorithms for calculating the ER have been proposed in the ATM Forum standards.

In the binary mode operation, the switch has to decide when to raise the alarm that congestion is pending. If we consider a non-blocking switch with output buffering, then if congestion occurs at an output port, the number of cells in its associated output buffer will increase dramatically. Typically, there are two thresholds associated with this buffer. A low threshold, T_{low}, and a high threshold, T_{high}. When the number of cells goes over T_{high}, the switch can start marking the EFCN bit of the data cells or turn on the CI or NI bit in a forward or backward RM cell. As the sources begin to react to the feedback information, the number of cells in the buffer will go down below T_{high}. However, the

switch continues marking until the number of cells in the buffer goes below T_{low}. At that moment, the switch stops the binary marking.

Simulation studies have shown that in a binary feedback scheme as the one presented above, it is possible that some connections may receive more than their fair share of the bandwidth. Let us consider the case where source A is very close to a switch, and source B very far away. Then A will react to the feedback information from the switch much faster than B. For instance, if congestion occurs in the switch, it will decrease its transmission rate quicker than B. By the same token, when the congestion is lifted, it will increase its transmission faster than B. As a result, source A may put through more traffic than B.

Problems

1. Consider a 64 Kbps voice connection transmitted at constant bit rate (silence periods are also transmitted).
   a. What is its PCR?
   b. What is its SCR?
   c. What is its average cell rate?

2. This is the continuation of problem 4, Chapter 5. On the average, a voice source is active (talkspurt) for 400 msec and silent for 600 msec. Let us assume that a voice call is transported over an ATM network via AAL 2. The voice is coded to 32 Kbps and silent periods are suppressed. We assume that the SSCS has a timer set to 5 msec. That is, each time the timer expires, it sends whatever data it has gathered to CPS as a CPS-packet. In problem 4, Chapter 5, you were asked to calculate the length of each CPS-packet, and the number of CPS-packets produced in each active period. Using this information, answer the following two questions:
   a. What is the peak and average transmission bit rate of the voice source including the CPS-packet overhead?
   b. What is the peak and average transmission rate of the voice source including the overheads due to the CPS-packet, the CPS-PDU and the ATM cell, assuming one CPS-packet per CPS-PDU?

3. Consider an on/off source where the off period is constant and equal to 0.5 msec. The MBS of the source is 24 cells. During the on period, the source transmits at the rate of 20 Mbps.
   a. What is its PCR?
   b. What is the maximum length of the on period in msec?
   c. Assuming a 1 msec period, calculate its SCR.

4. Explain why the end-to-end cell transfer delay consists of a fixed part and a variable part. What is the fixed part equal to?

5. Explain why jitter is important to a delay-sensitive applications.

6. Consider an on/off source with a peak bit rate=500 Kbps, and an average on period=100 msec. The source will be transmitted over the output port of a non-blocking switch, which has a buffer K=500 cells. Plot the average bit rate and the equivalent bandwidth of the source as a function of r, i.e., the fraction of time that the source is active. You should observe that the equivalent bandwidth tends to its

peak bit rate when the source is bursty and to its average bit rate when the source is regular, i.e. not bursty. (Remember to express K in bits!).

7. Consider the virtual scheduling algorithm for policing the PCR. Assume that T (1/PCR) =40 units of time and the CDVT=60 units of time. The arrival times are: 0, 45, 90, 120, 125, 132, 140, and 220. Which of these arrivals will get tagged?

8. Repeat problem 7 using the continuous state leaky bucket algorithm.

A simulation model of an ATM multiplexer – Part 2

This simulation project is the continuation of the simulation project entitled "A simulation model of an ATM multiplexer – Part 1" described at the end of Chapter 6. In Part 1, you were asked to plot the cell loss rate as a function of the arrival rate which was controlled by the probability p that a slot contains a cell. You should have observed that the cell loss rate of an ATM multiplexer increased, as the arrival rate of each source feeding the multiplexer increased.

The objective of this project is to show that the cell loss rate of an ATM multiplexer increases as the burstiness of each source increases, while keeping the peak and average cell rate of each source constant. This will be demonstrated by extending task 2 of the previous simulation project in order to allow for bursty arrival sources, modelled by an interrupted Bernoulli process (IBP).

You can either write your own simulation program following the instructions given below, or use a simulation language.

Project description

You will use the simulation model that you developed for task 2 in the previous simulation model. The structure of this simulation model will remain the same, except for the arrival processes. Specifically, in the previous simulation project, you assumed that the arrival process for each stream was Bernoulli. In this project, you will replace the arrival process for each stream by an IBP. All four arrival processes will be identical.

We will assume that time is slotted, and each slot is long enough so that a cell can completely arrive. As described in section 7.1.3, an IBP is an on/off process defined over a slotted time axis. The on and off periods are geometrically distributed. That is, given that in slot i the process is in the on period, then in the next slot i+1 it will be still in the on period with probability p, or it will change to the off period with probability 1-p. Likewise, if it is in the off period in slot i, it will stay in the off period in the next slot i+1 with probability q, or it will change to the on period with probability 1-q. If in the next slot it will be in the on period, then a cell may arrive with probability a.

In order to simplify the traffic model, we will assume that each slot during the on period contains a cell. That is, a=1. This traffic model can be completely defined if we know p and q. These probabilities can be obtained using the PCR, the average cell rate and the burstiness of the source as follows.

The burstiness of the source is defined by the squared coefficient of variation c^2. This quantity is defined as the variance of the inter-arrival time of cells divided by the mean inter-arrival time squared. For the above traffic model, it is given by the following expression:

$$c^2 = \frac{(1-p)(p+q)}{(2-p-q)^2} \quad (1)$$

The larger its value, the burstier is the source. For instance, compressed voice has a c^2 of 19. To create extremely bursty sources, c^2 can be set as high as 200.

We also have that

$$\text{Average cell rate} = \text{PCRx} \frac{\text{Average on period}}{\text{Average on period} + \text{Average off period}}$$

where

$$\text{Average on period} = \frac{1}{1 - p}, \text{ and}$$

$$\text{Average off period} = \frac{1}{1 - q}.$$

Substituting these two expressions for the average on and off periods into the above expression for the average cell rate, we have:

$$\text{Average cell rate} = \text{PCRx} \frac{1 - q}{2 - p - q} \quad (2)$$

We can estimate the parameters p and q of an IBP (assuming that during the on period cells arrive back-to-back), using expressions (1) and (2). For instance, let us consider a source with a peak bit rate of 1.544 Mbps, an average bit rate of 772 Kbps, and a c^2 of 20. Then, from (2) we have

$$\frac{1 - q}{2 - p - q} = 0.5$$

and from expression (1) we have:

$$\frac{(1 - p)(p + q)}{(2 - p - q)^2} = 20.$$

After some calculations we can obtain that p=q=0.9756.

Structure of the simulation

The structure of the simulation will remain the same as in task 2. The generation of an arrival will be done according to an IBP, whose PCR, average cell rate and c^2, will be specified as input to the simulation. Given a PCR, average cell rate and c^2, you should first calculate p and q. An IBP can be simulated as follows:

Draw a pseudo-random number r, where 0<r<1. If in current slot it is in the on period, then in the following slot it will remain in the on period if r<p. Else it will shift to the off period. On the other hand, if it is in the off period, then in the next slot it will remain in the off period if r<q. Else, it will shift to the on period. If it will be in the on period in the next time slot, then we have an arrival.

Once a cell has arrived, draw a new pseudo-random number r, 0<r<1, to decide which queue it will join. For that, follow the logic outlined in the previous simulation project.

Resultss

Calculate the cell loss probabilities per QoS queue and the total cell loss rate, for different values of c^2, and plot out your results. (If the curves are not smooth, increase the simulation run. This may happen as c^2 increases.

Estimating the ATM traffic parameters of a video source

An MPEG video encoder generates frames which are transmitted over an ATM link. Write a program to simulate the traffic generated by the MPEG video encoder with a view to characterizing the resulting ATM traffic.

Problem description

An MPEG video encoder generates frames with a group of pictures consisting of 16 frames. The first frame is an I-frame, and the subsequent 15 frames are P-frames. For a specific video clip, the number of bits X_n generated for the n^{th} frame can be obtained as a function of the number of bits generated for the previous 16 frames, using the following auto-regressive expression:

$$\begin{aligned} X_n = \quad & 0.412X_{n-1} + 0.12X_{n-2}+0.11X_{n-3}+ 0.07X_{n-4}+0.06X_{n-5}+0.05X_{n-6} \\ & +0.001X_{n-7}+0.032X_{n-8}-0.001X_{n-9}+0.001X_{n-10}-0.032X_{n-11}-0.002X_{n-12} \\ & -0.05X_{n-13}-0.041X_{n-14}-0.1X_{n-15}+0.37\ X_{n-16} + e_n \end{aligned}$$

where e_n is white noise and it follows the distribution $N(0,\sigma^2)$, with $\sigma^2 = 20{,}000$ bits. The following initial values are used: $X_1 = 150{,}000$ bits, and $X_i = 70{,}000$ bits for i = 2,3,…,16.

An I- or P- frame is generated every 30 msec. The information generated for each frame is transmitted over an ATM link using AAL1 unstructured PDUs. Assume that it takes zero time to pack the bits of a frame into AAL1 PDUs and subsequently into ATM cells. Also, assume that the ATM cells generated by a single frame are transmitted out back-to-back over a slotted link, with a slot equal to 3 μsec.

Assume now that you are obesrving the ATM cells transmitted out on this slotted link. Due to the nature of the application, you will see that the ATM traffic behaves like an on/off model. You are required to measure the following parameters of this ATM traffic: average cell rate, sustained cell rate with T=900 msec, MBS, average off period, and the squared coefficient of variation of the inter-arrival c^2.

Simulation structure

The simulation program can be organized into three parts. In the first part, you generate the size of the next frame, and in the second part you collect statistics on the ATM cells generated by this frame. You will repeat these two parts until you have generated 5,000 frames. Then, you will go to part 3, where you will calculate and print the final statistics.

Part 1:
Use the above auto-regressive model to generate the size (in bits) of the next frame. For this, you will need to keep the size of the previous 16 frames. Start generating from frame number 17 using the initial values X_i, i = 1,2,…, 16, given above. In addition to calculating the weighted sum of the previous 16 frames, you will also need to generate an estimate for e_n. This is a random variate drawn from the distribution $N(0,\sigma^2)$, with $\sigma^2 = 20{,}000$ bits. It can be generated using the following procedure:

1. Draw two random numbers r_1 and r_2, $0<r_1, r_2<1$.
   Calculate $v = 2r_1 - 1$, $u = 2r_2 - 1$, and $w = v^2 + u^2$
2. If $w>1$ go back and repeat step 1, else, $x = v\ [(-2\log_e w)/w]^{1/2}$
3. Set $e_n = 141.42x$

Part 2:
A new frame is generated every 30 msec. Having generated the frame size, calculate how many ATM cells are required to carry this frame. Let X be the number of required ATM cells. These ATM cells are generated instantaneously, and they are transmitted out back-to-back, with a transmission time equal to 3 μsec per cell. Calculate how many slots will be idle before the next frame arrives. Let the number of idle slots be Y. Update the following variables:

```
frame_counter = frame_counter + 1
total_simulation_time = total_simulation_time + 30
total_cells_arrived =  total_cells_arrived + X
```

MBS = max{MBS, X}
on_period = on_period + X
off_period = off-period + Y

For the sustained rate, set-up a loop to calculate the total number of cells S arrived in 30 successive frames, i.e. in 900 msec. When 30 frames have been generated, compare this value against S, and save the largest of the two back in S. (Initially, set S = 0).

To calculate c^2 you will need to keep all the inter-arrival times of the ATM cells. The inter-arrival is 1 between two cells that are transmitted back-to-back, and Y+1 between the last cell of a frame and the first cell of the next frame. Maintain two variables, Sum and SqSum. For each inter-arrival time t, do the following:

Sum = Sum + t
SumSq = SumSq + t^2

If frame_counter < 5,000, continue to generate frames, that is repeat parts 1 and 2. Otherwise go to part 3.

Part 3: Calculate and print out the required ATM traffic parameters:

Average cell rate = total_cells_arrived/total_simulation_time
SCR = S/900
MBS
average on period = on_period/frame_counter
average off period = off_period/fraame_counter
c^2 = Var/MeanSq, where
Var = [SumSq - (Sum^2/total_cells_arrived)]/(total_cells_arrived - 1)
MeanSq = $(Sum/total\_cells\_arived)^2$

PART THREE:
DEPLOYMENT OF ATM

Part three deals with two different topics, namely, how IP traffic is transported over ATM, and ADSL-based residential access networks. Part Three consists of Chapters 8 and 9.

Chapter 8: Transporting IP Traffic Over ATM

This Chapter deals with the very interesting topic of how IP traffic is transported over an ATM network. The following schemes are presented in this Chapter: *LAN emulation* (LE), *classical IP and ARP over ATM*, *next hop routing protocol* (NHRP), *IP switching*, *tag switching*, and *multi-protocol label switching* (MPLS).

Chapter 9: ADSL-Based Access Networks

In this Chapter, we present the *asynchronous digital subscriber line* (ADSL) technology, and discuss schemes for accessing *network service providers* (NSP)

.

CHAPTER 8
Transporting IP Traffic Over ATM

In this Chapter, we present various solutions that have been proposed to carry IP traffic over ATM. We first present ATM Forum's *LAN emulation* (LE), a solution that enables existing LAN applications to run over an ATM network. Then, we describe IETF's schemes *classical IP and ARP over ATM* and *next hop routing protocol* (NHRP) designed for carrying IP packets over ATM. The remaining of the Chapter is dedicated to the three techniques *IP switching*, *tag switching*, and *multi-protocol label switching* (MPLS). IP switching utilizes the label swapping functionality of an ATM switch to transport IP packets in an efficient manner. IP switching had a short life span, but it inspired the development of tag switching, which has being standardized by IETF under the name of multi-protocol label switching. Tag switching, and also MPLS, have been primarily designed for IP networks, but they can also be used for ATM networks.

8.1 Introduction

In recent years we have witnessed a tremendous growth in the number of hosts attached to the Internet. As the Internet traffic increases, the need to route IP packets faster increases as well. Several solutions have been proposed to switch IP traffic ranging from gigabit routers to using the switching capability and functionality of ATM.

Since the early 90s, there has been a quest for finding a solution to the problem of transporting connectionless traffic over ATM, which is inherently connection-oriented. The IETF and the ATM Forum have proposed several techniques for using the ATM network to transport IP packets, such as *LAN emulation*, *classical IP and ARP over ATM*,

and *next hop routing protocol* (NHRP). The development of these techniques was motivated by the desire to introduce ATM technology with as little disruption as possible to the existing IP model.

LAN emulation, as the name implies, emulates the characteristics and behaviour of a LAN over an ATM network. It allows existing LAN applications to run over an ATM network without any modifications. LAN emulation was considered to be a good solution for providing faster connectivity to the desktop, since ATM could run at speeds such as OC-3, 100 Mbps TAXI, and 25 Mbps. These transmission speeds should be contrasted with the 10Mpbs Ethernet that was available at that time, which due to software bottlenecks had an effective bandwidth of around 2 Mbps. LAN emulation was implemented and deployed successfully in the field. However, it never became the dominant technology in the LAN environment. This was due to the advent of the 100 Mbps Ethernet which provided a high-speed connectivity to the desktop without having to change the underlying transport technology. The dominance of Ethernet in the LAN environment was further strengthened with the advent of Ethernet switches, and later on with the advent of gigabit Ethernet.

Classical IP and ARP over ATM was developed for a single IP subnet, that is, for a set of IP hosts that have the same IP network number and subnet mask. The members of the subnet communicate with each other directly over ATM, and they communicate with IP hosts outside their subnet via an IP router. Address resolution within the subnet is an important function of the protocol. This is necessitated by the fact that IP addresses are different to ATM addresses. Thus, there is a need to translate the IP address of a host to its corresponding ATM address and vice versa. Finally, the next hop routing protocol (NHRP), pronounced nerp, is an address resolution technique for resolving IP addresses with ATM addresses in a multiple subnet environment.

A different approach to switching IP packets, referred to as *IP switching*, was proposed by Ipsilon Networks (Ipsilon was later on purchased by Nokia). This technique is based on the label swapping functionality of an ATM switch. IP switching inspired the development of CISCO's *tag switching*, which was designed primarily for IP routers. Tag switching was proposed in order to circumvent the CPU-intensive table look-up in the forwarding routing table necessary to determine the next-hop router of an IP packet. It

was also proposed as a means of introducing quality-of-service in the IP network. Tag switching served as the basis for a new protocol known as *multi-protocol label switching* (MPLS). This is an exciting new protocol has been developed by IETF. Interestingly enough, since the introduction of tag switching, several CPU-efficient algorithms for carrying out look-ups in the forwarding routing table were developed. The importance of MPLS, however, was by no means diminished since it is regarded as a solution for introducing quality-of-service into the IP networks.

Figure 8.1: The protocol stack with the LE layer

8.2 LAN emulation (LE)

LAN emulation is a service developed by the ATM Forum that enables existing LAN applications to run over an ATM network. This service emulates the characteristics and behaviour of LANs. It provides connectionless service, broadcast and multicast service as supported by shared media LANs, it maintains the MAC address identity of each individual device attached to the LAN, and it allows existing LAN applications to work unchanged. LAN emulation provides interconnectivity over an ATM network among 802.3 Ethernet LANs, 802.5 token ring LANs, ATM stations, servers, and stations attached to LANs.

LAN applications and protocols run on top of the logical link control (LLC) layer, described in section 2.6. The LLC layer requires services from a MAC layer. If the LLC layer is kept the same, then the applications and protocols running on top of it do not have to change. In LAN emulation, the MAC layer is replaced by a *LAN emulation* (LE) layer which provides MAC service to LLC and which runs on top of ATM, as shown in figure 8.1. This solution permits various protocols such as IP, IPX, DECnet, and Appletalk, to run in an emulated LAN.

LAN emulation allows an application to run on a computer which has an ATM interface and is directly connected to the ATM network, as shown in figure 8.2. LAN

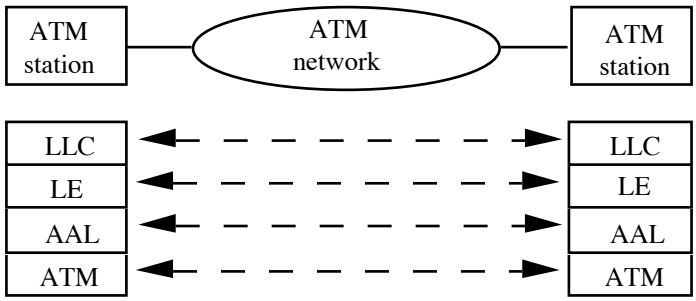


Figure 8.2: An application can run on an ATM station

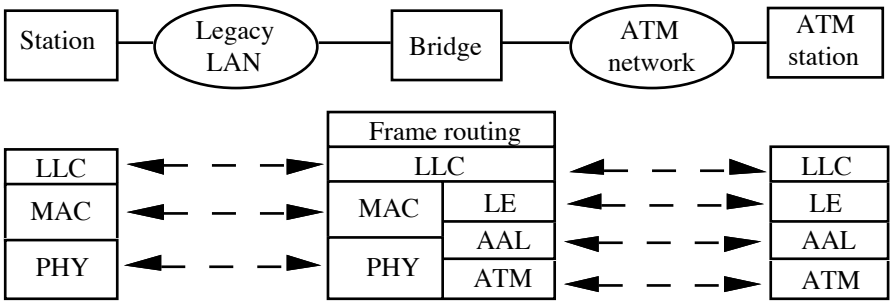


Figure 8.3: A station attached to a LAN can communicate with an ATM station

emulation also allows a station attached to a LAN, such as Ethernet, to communicate with another station which is attached directly to an ATM network. This communication is enabled through a bridge, as shown in figure 8.3. LANs could be interconnected via an ATM network using LAN emulation, as shown in figure 8.4.

An emulated LAN is a single segment LAN that can either be an Ethernet or a token ring. Membership in an emulated LAN is logical, whereas in a LAN membership is defined by who is physically attached to it.

A key component in LAN emulation is the *LAN emulation client* (LE client). The LE client resides at the end-device and it provides a MAC level emulated IEEE 802.3 Ethernet or 802.5 token ring interface to LLC. It performs functions emulating an IEEE 802.3 or 802.5 LAN, such as, control functions, data forwarding, and address resolution.

The interaction between LE clients and the LE servers is done via the *LAN emulation user to network interface* (LUNI), shown in figure 8.5. The following services are provided: initialization, registration, address resolution, and data transfers. Initialization is used to obtain the ATM address of the LE services, and to join or leave a particular emulated LAN. Registration is used to inform the LE services of the list of

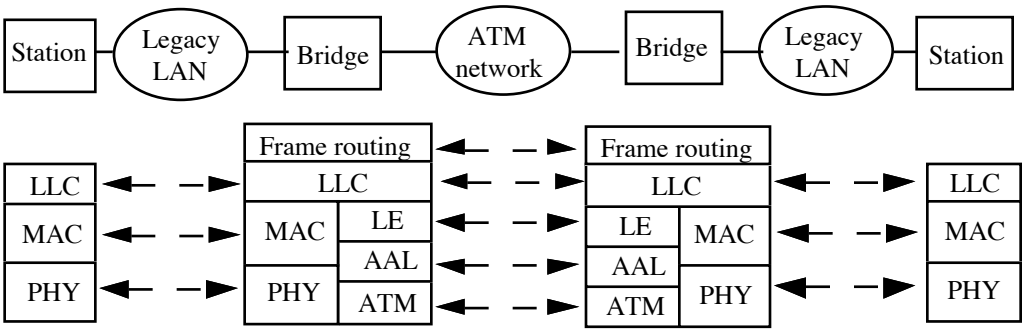


Figure 8.4: LANs could be interconnected via an ATM network

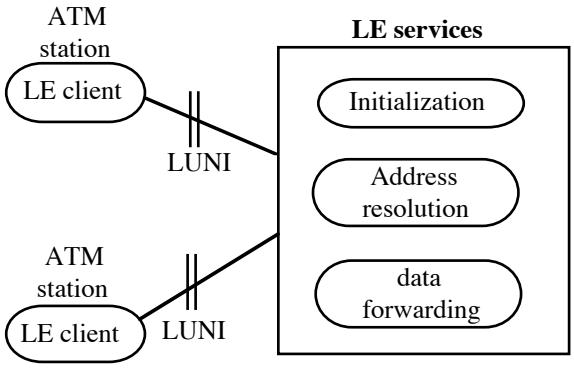


Figure 8.5: The LAN emulation user network interface (LUNI)

individual MAC addresses that the LE client represents, and the list of multicast groups that the LE client belongs to. Address resolution is used to obtain the ATM address of an LE client with a known MAC address.

In addition to the LE client, the following LAN emulation components have also been defined:

LAN emulation server (LE server): The LE server provides an address resolution mechanism for resolving MAC addresses. Also, it performs functions such as

registration of an LE client, forwarding address resolution requests, and managing LE client address registration information.

Multicast servers (MCS): One or more multicast server is used in an emulated LAN to provide the LE clients the connectionless data delivery characteristics found in a shared network. Its main task is to distribute data with a multicast address, to deliver initial unicast data before the destination ATM address has been discovered, and to distribute data with explorer source routing information.

Broadcast and unknown server (BUS): The BUS provides services to support broadcasting and multicasting, and initial unicast frames sent by an LE client before the target ATM address has been resolved. This multicast server must always exist in an emulated LAN and all the LE clients must join its distribution group. If there are no other multicast servers in the emulated LAN, the BUS handles all the multicast traffic.

An LE client has separate virtual circuit connections (VCC) for control traffic and data traffic. The VCCs carry control or data traffic for only one emulated LAN and they may be permanent or switched virtual circuits or a mixture of both. A pictorial view of these VCCs is given in figure 8.6.

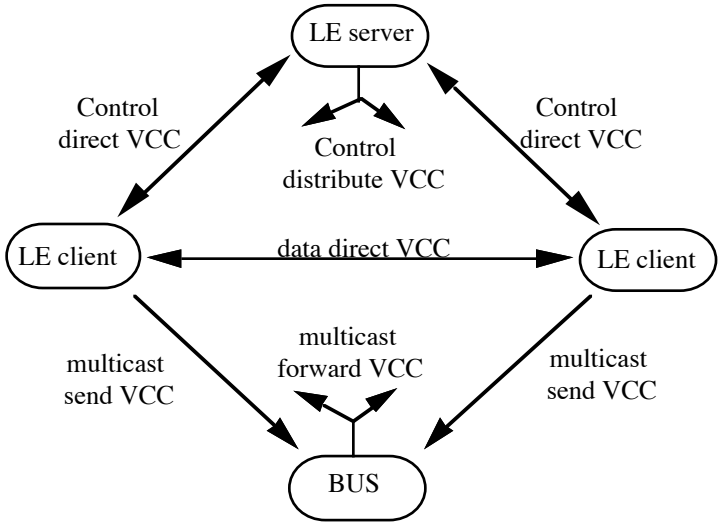


Figure 8.6: LAN emulation virtual circuit connections

Control VCCs link the LE client to the LE server and carry *LAN emulation address resolution* (LE-ARP) traffic and control frames. They are set-up during the initialization phase. The following types of control VCCs are used: *control direct VCC* and *control distribute VCC*. The control direct VCC is set-up by the LE client as a bi-directional point-to-point VCC to the LE server. It is maintained by both the LE client and the LE server as long as the LE client participates in the emulated LAN. The control distribute VCC is set-up by the LE server. It is an optional unidirectional control VCC to the LE clients and it is used by the LE server to distribute control traffic. It may either be point-to-multipoint VCC or a point-to-point VCC.

Data VCCs connect the LE clients to each other and to the multicast servers. The following types of data VCCs are used: *data direct VCC*, *multicast send VCC*, and *multicast forward VCC*. A data direct VCC is a bi-directional point-to-point VCC between two LE clients that want to exchange unicast data traffic. A multicast send VCC is a uni-directional point-to-point VCC established by an LE client to an MCS or BUS. It is used by the LE client to send multicast data to an MCS or to the BUS. In the case of the BUS, it may also send initial unicast data. A multicast forward VCC is a unidirectional VCC from an MCS or the BUS to all the LE clients. It is used for distributing data to the LE clients. This may be a point-to-multipoint or a point-to-point VCC.

Address resolution

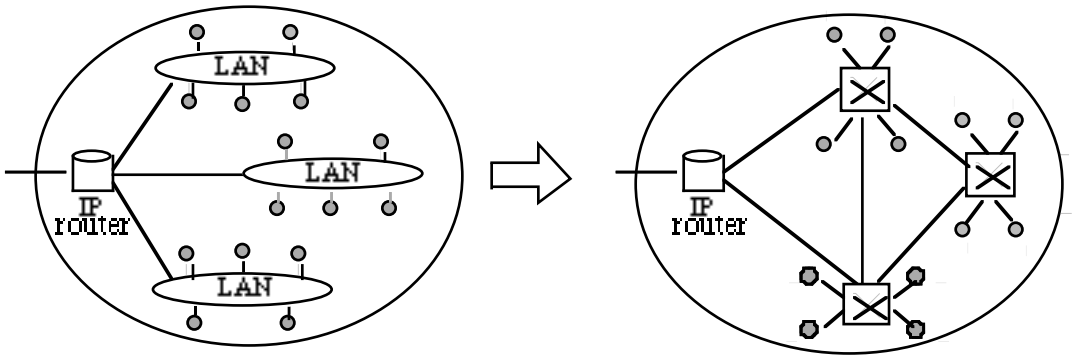
When an LE client is presented with a frame for transmission to a destination LE client, it may or may not know the destination's ATM address. It knows, however, the destination's MAC address since it will be passed on by the LLC. If it knows the ATM address, then it can establish a data direct VCC to the destination LE client and then transmit the frame. If it does not know the ATM address, the LE client sends a *LAN emulation address resolution protocol* (LE-ARP) request frame to the LE server over its control direct VCC. The LE-ARP request includes the source and destination MAC addresses, and the ATM address of the originating LE client. Since the LE server maintains a table of all MAC addresses and corresponding ATM addresses, it may be able to issue an LE-ARP reply to the requesting LE client. Alternatively, the LE server

may forward the LE-ARP request to the appropriate LE client over the control distribute VCC or over one or more control direct VCCs. This will be for the case where the destination MAC address belongs to a workstation attached to a LAN on the other side of the bridge. All LE clients in the emulated LAN are required to accept this request. Each LE client checks the destination MAC address and if it is his, it responds to the LE server over the control direct VCC with an LE-ARP reply. That reply is sent by the LE server over the control direct VCC to the originating LE client.

Alternatively the requesting LE client can elect to transmit the frame to BUS through a multicast send VCC. The BUS then forwards the frame to the designated LE client.

8.3 Classical IP and ARP over ATM

Classical IP and ARP over ATM is a technique standardized by IETF designed to support IP over ATM in a single *logical IP subnet* (LIS). A LIS is a group of IP hosts that have the same IP network address, say 192.43.0.0, and the same subnet mask, as shown in figure 8.7a. Now, let us assume that the LANs are replaced by three interconnected ATM switches, as shown in figure 8.7(b). Each host can communicate directly with any other



(a) IP: 192.43.0.0 (b)

Figure 8.7: A logical IP subnet (LIS)

host in the subnetwork over an ATM connection. The traditional IP model remains unchanged and the IP router is still used to connect to the outside of the subnet.

The term “classical” in the name of “classical IP and ARP over ATM” is a reference to the use of ATM as a networking interface to the IP protocol stack operating in a LAN environment.

IP packets are encapsulated using the IEEE 802.2 LLC/SNAP encapsulation. The protocol used in the payload, such as IP, ARP, Appletalk, and IPX, is indicated in the LLC/SNAP header. An encapsulated packet becomes the payload of an AAL 5 frame. The maximum transfer unit (MTU) is fixed to 9180 bytes. Adding an 8-byte LLC/SNAP header gives a total of 9188 bytes, which is the defaulted size for an AAL 5 frame.

8.3.1 ATMARP

Each member of the LIS is configured with an IP address and an ATM address. When communicating with another member in the same LIS over ATM, it is necessary to resolve the IP address of the destination host with its ATM address. IP addresses are resolved to ATM addresses using the *ATMARP* protocol within the LIS. This protocol is based on ARP, see section 2.8, and it has been extended to operate over a non-broadcast unicast ATM network. The *inverse ATMARP* (InATMARP) protocol is used to resolve an ATM address to an IP address. It is based on RARP, see section 2.8, only it has been extended to support non-broadcast unicast ATM networks.

The ATMARP protocol utilizes an ATMARP server which can run on an IP host or an IP router and which must be located within the LIS. The LIS members are clients to the ATMARP server, and they are referred to as ATMARP clients. The ATMARP server maintains a table or a cache of IP and ATM address mappings. It learns about the IP and ATM addresses of ATMARP clients through a registration process described below. At least one ATMARP server must be configured with each LIS. The following ATMARP messages have been defined.

ATMARP_request: An ATMARP client sends an ATMARP request to the ATMARP server to obtain the ATM address of a destination ATMARP client. The message contains the client’s IP and ATM addresses, and the IP address of the destination client.

ATMARP_reply: This message is used by the ATMARP server to respond to an ATMARP_request with the requested ATM address. It contains the IP and ATM addresses of both the requesting and the destination clients.

ATMARP_NAK: Negative response issued by the ATMARP server to an ATMARP_request.

InATMARP_request: Used to request the IP address of a destination. The message contains the sender's IP and ATM addresses and the destination's ATM address.

InATMARP_reply: This is the response to an InATMARP_request with the destination's IP address. It contains the IP and ATM addresses of both the sender and the destination.

Registration

An ATMARP client must first register its IP and ATM addresses with the ATMARP server. This is done by invoking the ATMARP protocol as follows. Each ATMARP client is configured with the ATM address of the ATMARP server. After the client establishes a connection to the ATMARP server, it transmits an ATMARP_request on that connection. In the message, it provides its own IP and ATM addresses and it requests the ATM address of itself by providing its own IP address as the destination IP address. The ATMARP server checks against duplicate entries in its table, time stamps the entry, and adds it to its table. It confirms the registration of the ATMARP client by sending an ATMARP_reply. If a client has more than one IP address within the LIS, then it has to register each IP address with the ATMARP server.

Entries in the table of the ATMARP server are valid for a minimum of 20 minutes. If an entry ages beyond 20 minutes without being updated (refreshed), then the entry is removed from the table. Each ATMARP client is responsible for updating its entry in the ATMARP server's table at least every 15 minutes. This is done by following the same procedure used to register with the ATMARP server. That is, the ATMARP client sends an ATMARP_request to the ATMARP server with the destination IP address set to its own IP address. The ATMARP server updates the entry and confirms it by responding with an ATMARP_reply.

Address resolution

Let us assume that ATMARP client 1 wants to communicate with ATMARP client 2. We assume that both clients are in the same LIS. If there is already an established connection between the two clients, traffic can flow immediately. Otherwise, a connection can be set-up if client 1 knows the ATM address of the destination client 2. If its destination ATM address is not known, client 1 sends an ATMARP_request to the ATMARP server. If the

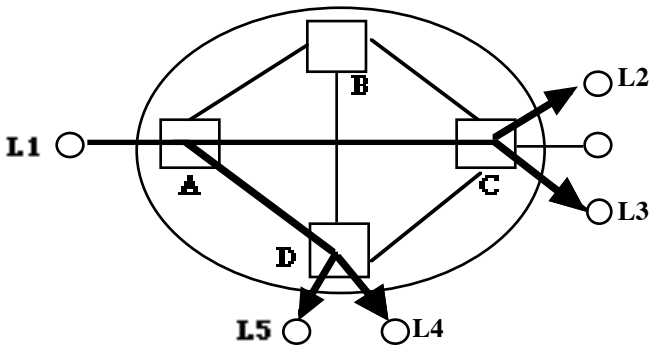


Figure 8.8: Multicasting in ATM

server has the requested address in its table, it returns an ATMARP_reply. Otherwise, it returns an ATMARP_NAK. Upon receipt of the ATMARP_reply, a connection is established and traffic starts flowing.

An ATMARP client creates an entry in its ATMARP table for every connection (PVCs or SVCs) that it creates. An entry is valid for a maximum of 15 minutes. When an entry has aged, the client must update it. If there is no open connection associated with the entry, then the entry is deleted. If the entry is associated with an open connection, then the client must update the entry prior to using the connection to transmit data. In the case of a PVC, the client transmits an InATMARP_request and updates the entry on receipt of the InATMARP reply. In the case of an SVC, it transmits an ATMARP_request to the ATMARP server, and updates the entry on receipt of the ATMARP_reply.

An ATMARP client is also permitted to initiate the above procedure for updating an entry in the table, before the entry has aged.

8.3.2 IP multicasting over ATM

IP uses the class D address space to address packets to the members of a multicast group. Hosts and routers exchange messages using a group membership protocol called the *internet group management protocol* (IGMP). The routers use the results of this message exchange along with a multicast routing protocol, such as MOSPF, to build a delivery tree from the source subnetwork to all other subnetworks that have members in the multicast group.

In ATM, multicasting is implemented using a point-to-multipoint connection between a sending end-device and multiple receiving end-devices. In multicasting, the sender is known as the *root* and the receivers are known as the leaves. An example of a point-to-multipoint connection is shown in figure 8.8. The root is end-device L1 and the

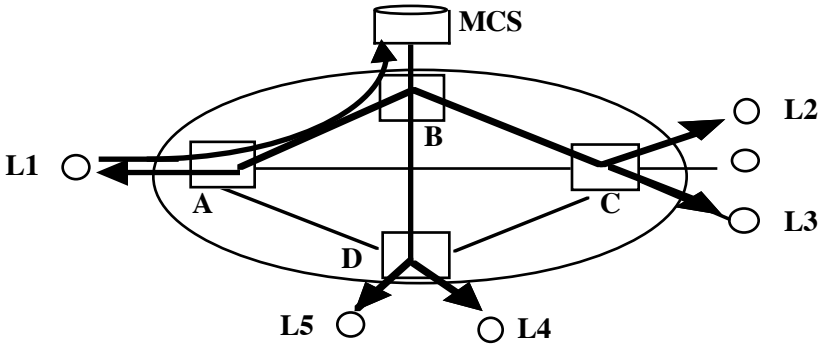


Figure 8.9: Multicasting in ATM

leaves are end-devices L2, L3, L4, and L5. As we can see, the connection is set up utilizing the multicasting feature of an ATM switch. Switch A receives cells from L1 and it transmits them out on two different links, namely on the link to switch C and on the link to switch D Switch C receives cells from A and it transmits them out to its end-devices L2 and L3. Finally, switch D receives cells from A and transmits them to L5 and L6.

A set of hosts that participate in an ATM-based multicast is referred to as a *cluster*. Two models exist to support an ATM-based multicast, namely, the *VC mesh*, and

the *multicast server* (MCS). In the VC mesh solution, a point-to-multipoint VC connection is set-up for each host that wants to transmit multicast traffic. If all the hosts in a multicast cluster want to transmit and receive multicast traffic, then we will have a multicast tree for each host. That is, in the example shown in figure 8.8, if the cluster consists of L1 to L5, then each end-device will have a point-to-multipoint VC connection to the other end-devices, with it being the root. In this case, each host is both the root and also a leaf in many point-to-multipoint VC connections. This criss-crossing of VC connections across the ATM network has given rise to the name of "VC mesh".

In the multicast server solution, each cluster member has a VC connection to a multicast server. The multicast server maintains a point-to-multipoint VC connection to all the cluster members. A host simply forwards its multicast traffic to the multicast server directly over its VC connection, which then multicasts it to the cluster members over its point-to-multipoint VC connection. An example of the multicast server solution is shown in figure 8.9. When L1 wants to multicast traffic to the other cluster members L2 to L5, it sends it to the multicast server, who in turn multicasts it to all the cluster members. A side effect of this scheme is that L1 will receive a copy of its own traffic back from MCS. An alternative solution is for the MCS to establish a point-to-point VC connection to each member of the cluster. This solution avoids the problem of a source receiving its own traffic We note that the MCS solution is also used in LAN emulation, described in section 8.2.

The VC mesh solution offers optimal performance, but it requires more VC connections than the MCS solution. In view of this, it does not scale up well. The MCS solution is easier to manage but congestion bottlenecks may occur within the MCS. In IP and ARP over ATM, both the VC mesh and the multicast server models are implemented.

As we have seen in the previous section where we discussed the unicast case, a mapping between the IP and ATM addresses of a host must be provided. This is done using the ATMARP server. A similar mapping is required in the multicasting case. This mapping is between an IP multicast group address and the ATM addresses of the members of the IP multicast group. This is done using the *multicast address resolution server* (MARS). MARS can be seen as the analog of the ATMARP server in multicasting. It supports a single cluster, and it can co-exist with the ATMARP server or

an MCS on the same ATM switch. MARS does not participate in the actual multicasting of IP packets.

MARS maintains a table of IP group addresses, and for each IP group address it keeps all the ATM addresses of the cluster members that have registered with the specific multicast group. The table contains the following information:

{IP group address, ATM address 1, ATM address 2,..., ATM address N}.

This table is known as the *host map*. The information in a host map is either configured manually or it is learned as will be seen below.

The members of the cluster are clients to MARS. The MARS client protocol runs above the ATM adaptation layer and below the LLC layer, as shown in figure 8.1. A MARS client can run on a IP host or an IP router. A MARS client that wants to join a particular IP multicast group establishes a VC connection to MARS. This connection is torn down if it is not used after a configurable period of time. The minimum suggested time is 1 minute and the recommended default value is 20 minutes. The VC connection is used exclusively to send queries to MARS and receive replies from MARS.

The VC mesh scheme

MARS maintains a point-to-multipoint VC connection to all the members of a cluster which is known as the *ClusterControlVC* That is, the cluster members are leaves of ClusterControlVC. This VC connection is used by MARS to distribute group membership information to the cluster members.

A MARS client who wishes to join a specific IP multicast group sends a MARS_JOIN message over the VC connection that it has established to MARS. A MARS_JOIN message contains the ATM address of the MARS client and the IP multicast address that it wants to join. When MARS receives a MARS_JOIN message, it adds the client as a leaf to the ClusterControlVC, and it then multicasts the MARS_JOIN message to all the cluster members associated with the particular multicast over the ClusterControlVC. The MARS client confirms that it has registered with the multicast group when it receives a copy of the MARS_JOIN message over the ClusterControlVC.

Alternatively, MARS can confirm the registration of the client by sending back a copy of the message over the point-to-point VC connection between the client and MARS.

When a MARS client wants to leave a specific IP multicast group, it sends a MARS_LEAVE message to MARS. This message contains similar information as the MARS_JOIN message, that is, the client's ATM address and the IP multicast address that it wants to leave. When MARS receives the message it removes the client from its ClusterControlVC, and then it multicasts a copy of the message over the ClusterControlVC. MARS confirms that it has received and processed the message by sending back to the client a copy of the message over the point-to-point VC connection.

Now let us see how this information that MARS collects about membership in the various IP multicast groups is used by a MARS client. When the IP layer in a host passes down an IP multicast packet, the host's MARS client ascertains whether a point-to-multipoint VC connection to the other cluster members that participate in the multicast exists. If it does not exist, it issues a MARS_REQUEST to MARS to request address resolution of the IP group address to which the multicast packet should be sent to. MARS responds with a sequence of MARS_MULTI messages which contain the host map of the IP group address. A MARS_MULTI message carries as many ATM addresses as possible, but its length is limited to the maximum transfer unit (MTU) of the underlying ATM connection. Therefore, depending upon the size of the host map, more than one MARS_MULTI message may be required. If MARS does not have a host map for the requested IP group address, it returns a MARS_NAK. Once the MARS client has the host map, it can create its own point-to-multipoint connection in order to multicast its IP packet. After transmitting the packet, the point-to-multipoint connection is kept open for any subsequent IP multicast packets.

As we have seen above, changes to the multicast membership are announced by MARS, so that the host can accordingly add or drop a leaf. The signalling procedures for setting up a point-to-multipoint VC connection and adding or dropping leaves are described in Chapter 10.

The multicast server (MCS) scheme

In this scheme, it is the MCS that maintains a point-to-multipoint VC connection to the members of the cluster. A client simply forwards its traffic to the MCS who in turn multicasts it over the point-to-multipoint VC connection. A MARS client registers with MARS following the same procedure as in the VC mesh case. However, the membership information is sent to the MCS rather than to the cluster members. MARS announces membership information to the MCS over a connection known as the *ServerControlVC*. Since it is possible to have a number of MCSs, ServerControlVC is a point-to-multipoint connection where the leaves are the MCSs. ServerControlVC is analogous to the ClusterControlVC.

An MCS must first register with MARS. For this, the MCS must have the ATM address of MARS which can be configured at start-up time of the MCS. After establishing a point-to-point VC connection to MARS, the MCS issues a MARS_MSERV message. When MARS receives this message, it adds the MCS to its ServerControlVC and returns a copy of the MARS_MSERV back to the MCS in order to confirm its registration. An MCS can drop from MARS by issuing a MARS_UNSERV message. MARS removes the MCS from its ServerControlVC and returns a copy of the MARS_UNSERV message to confirm it.

During registration, no IP multicast groups are identified. An MCS can subsequently register with MARS to support one or more IP group addresses using again a MARS_MSERV message. MARS confirms it by sending back a copy of the MARS_MSERV message. An MCS uses a MARS_UNSERV to specify to MARS that it does not want to support a specific IP group address. MARS confirms it by sending back a copy of the message. The confirmation messages that MARS sends back to the MCS are transmitted either over the ServerControlVC or the VC connection established between the MCS and MARS.

After an MCS registers with MARS to support an IP group address, it issues a MARS_REQUEST message to obtain the host map. MARS sends the information back in a sequence of MARS_MULTI messages. MARS sends a MARS_NAK if there is no host map. Subsequently, the MCS creates a point-to-multipoint VC connection to all the hosts that have registered with MARS for this particular multicast.

For each IP group address, MARS keeps two sets of mappings, namely the host map and the *server map*. The server map contains the information:

{IP group address, ATM address of MCS 1,…, ATM address of MCS K}.

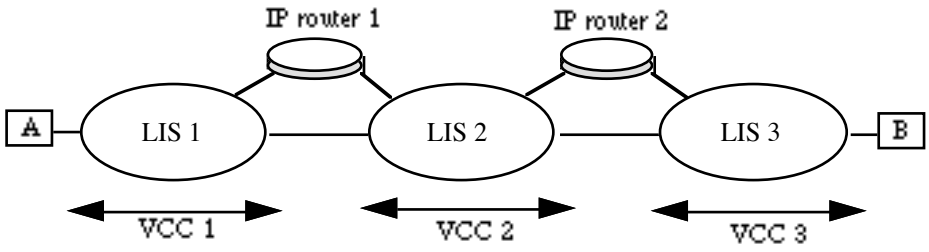


Figure 8.10: Classical IP and ARP over ATM

Typically K is equal to 1, but it can be greater than one if more than one MCS is configured to serve an IP multicast group.

Now let us take a brief look at a host. A host follows the same procedure as in the VC mesh case in order to join or leave a multicast group. That is it registers with MARS to participate in a specific multicast group using the MARS_JOIN, and it notifies MARS that it wants to leave a multicast group using the MARS_LEAVE message. Any changes in the multicast membership are reported to MARS which subsequently announces them to the MCS server using MARS_SJOIN and MARS_SLEAVE messages. When a host sends a MARS_JOIN to MARS requesting to join a particular multicast group, MARS does not return the host map as in the VC mesh case, but it returns the server map. The client is lead to believe that the MCSs are the members of its multicast group, and it uses the server map to build a point-to-multipoint connection to its MCSs.

8.4 Next hop routing protocol (NHRP)

The IP model consists of networks which are interconnected by IP routers. Packets travelling from one network to another have to pass through an IP router. In classical IP and ARP over ATM, connectivity is limited to a single LIS. Traffic between two LISs has to pass through an IP router. In figure 8.10, we give an example of how end-devices A and B can communicate using classical IP and ARP over ATM. As can be seen, A is attached to LIS 1 and B to LIS 3. The two LISs communicate via IP routers 1 and 2. IP

router 1 is attached to LIS 1 and 2, and IP router 2 is attached to LIS 2 and 3. Communication between A and B involves three separate VC connections. A communicates with IP router 1 via VC connection 1, IP routers 1 and 2 communicate via VC connection 2, and IP router 2 communicates with end-device B via VC connection 3. Since VC connection 1 terminates at IP router 1, the router has to re-assemble the original IP packets from the ATM cells, and then for each IP packet it has to do a table look-up in the forwarding routing table in order to determine the packet's next hop. The same applies to

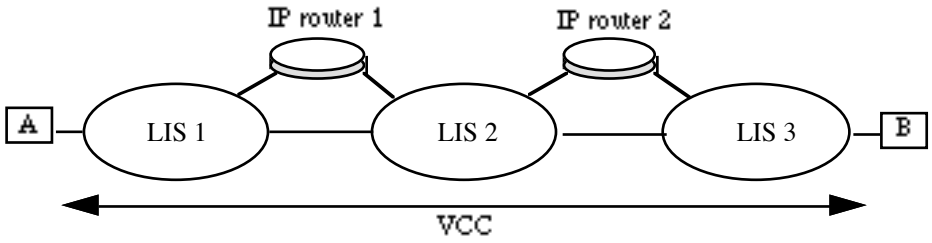


Figure 8.11: Direct connection

IP router 2. This, of course, introduces delays and additional work for the IP routers. Also, by terminating a connection at a router, the quality of service provided over this connection is terminated as well, since there is no quality-of-service guarantees within a router.

It is clear that some of the traffic would benefit if it were routed directly to the destination host without having to go through one or more IP routers. An example of this direct method is shown in figure 8.11. A establishes a direct VC connection to B, thus by-passing the two IP routers along the way. This method can be useful for a source that requires a specific quality of service, whereas the classical IP and ARP over ATM method shown in figure 8.10 can be used for sources which only require best effort.

The *next hop resolution protocol* (NHRP), pronounced *nerp*, is a technique proposed by IETF for resolving IP addresses to ATM addresses in a multiple subnet environment. It provides a host or an IP router with the ATM address of a destination IP address, so that a direct VC connection can be established. NHRP is not a routing

protocol. It was originally designed as an extension to the classical IP and ARP over ATM, but it is not limited to IP networks.

NHRP is a master/slave protocol. The NHRP server is known as the *next hop serve*r (NHS), and the NHRP client is known as the *next hop client* (NHC). A NHRP server provides NHRP service for NHRP clients in a network which does not inherently support broadcasting or multicasting, and where the hosts and IP routers attached to the network can communicate with each other directly. Such a network is known as a *non broadcast multiaccess network* (NBMA). ATM, frame relay, X.25, and SMDS are examples of an NBMA.

A NHRP client must be attached to an ATM network and it must know the ATM address of its NHRP server. NHRP clients can be serviced by one or more NHRP servers. A NHRP server can be located on a peer host or on a default IP router attached to an ATM network. A NHRP server is configured with its own IP and ATM address, and a set of IP address prefixes that correspond to the domains of the NHRP clients that it serves. NHRP can work with any layer-3 internetworking protocol, such as IPX and Appletalk, over any NBMA network. The following NHRP messages have been defined.

NHRP next hop resolution request: Query sent from a NHRP client to a NHRP server requesting resolution of a destination IP address to an ATM address. It contains the IP and ATM addresses of the source, and the destination IP address.

NHRP next hop resolution reply: Response sent by a NHRP server to a query. It contains the IP and ATM addresses of the source, the IP and ATM addresses of the destination, and a NAK code.

NHRP registration request: Sent by a NHRP client requesting to register with the NHRP server. It contains the IP and ATM addresses of the source, and the IP address of the NHRP server.

NHRP registration reply: Response sent by a NHRP server to a registration request. It contains the IP and ATM addresses of the source, the IP address of the NHRP server, and a NAK code.

NHRP purge request: Used to invalidate cached information contained in a NHRP client or server. It contains the IP and ATM addresses of the source, and the IP address to be purged from the receiver's database.

NHRP purge reply: Sent in response to a NHRP purge request.

NHRP error indicated: Used to convey error information to the sender of a NHRP message. It contains the IP and ATM addresses of the source, and an error code.

NHRP address resolution

Let us consider a single NBMA network that contains a number of LISs, namely LIS 1 and LIS 3. The two LISs are connected via LIS 2, as shown in figure 8.12. End-device A is attached to LIS 1 and it wants to establish a connection with end-device B, which is attached to LIS 3. The two LISs are connected by IP routers 1 and 2 which also serve as NHRP servers for LIS 1 and LIS 3 respectively. The two IP routers are connected by a permanent virtual connection. The following steps will take place:

1. A sends a NHRP next hop resolution request message to NHRP server 1 with the information {A's ATM address, A's IP address, B's IP address}.
2. NHRP server 1 checks to see if it serves B. It also checks to see if it has an entry in its cache for B's IP address. If neither is true, it sends the NHRP next hop resolution request to the adjacent NHRP server 2.

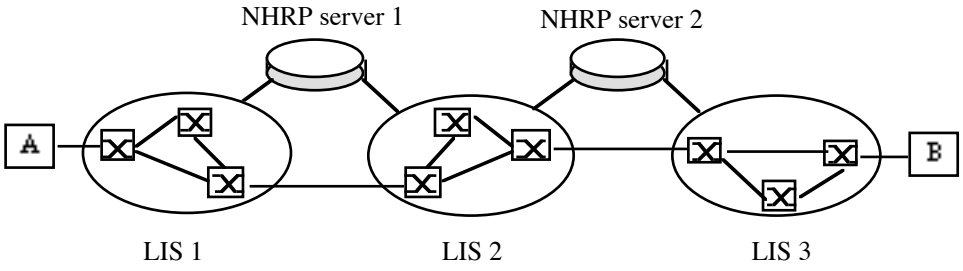


Figure 8.12: An example of address resolution

3. NHRP server 2 receives the request from NHRP server 1, determines that it serves B's IP address, looks up its cache or table which contains the IP and ATM address of B, and returns the information to A over the path that the request came from.

As the NHRP resolution reply goes back to A, NHRP server 1 may cache the information contained in the packet.

4. A sets up a direct VC connection to B, and data starts flowing.

The NHRP protocol is also used in the ATM Forum standard *multi-protocol over ATM* (MPOA). This protocol integrates LAN emulation and NHRP, and it permits two devices attached to separate emulated LANs to set-up an ATM connection and communicate directly.

8.5 IP switching

IP switching is an alternative technique to the schemes described so far in this Chapter, and it was proposed by Ipsilon Networks (Ipsilon was later on purchased by Nokia). Similar schemes were also proposed by other companies. For instance, Toshiba proposed the *flow attribute notification protocol* and CISCO proposed *tag switching*. These techniques are collectively known as *label switching* techniques. IP switching had a short life span, but it inspired the development of CISCO's tag switching, which was standardized by IETF under the name of *multi-protocol label switching* (MPLS).

To understand the motivation behind this technique, let us consider the network shown in the figure 8.13. It consists of an ATM network of five switches and of IP routers A, B, and C. Each IP router is attached to an ATM switch. We assume that routers A and B are interconnected via a PVC using AAL 5. Likewise, routers B and C are also interconnected via a PVC using AAL 5.

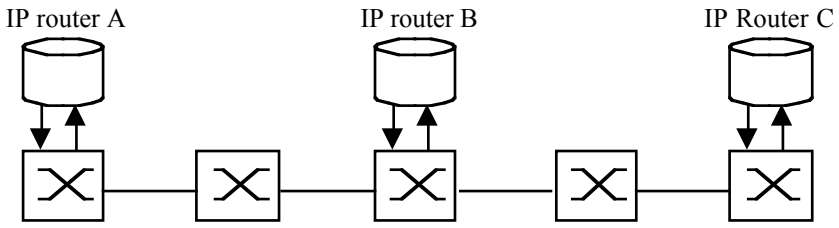


Figure 8.13: An example of three routers connected over ATM

Now, let us assume that IP router A has an IP packet to send to IP router C. From its forwarding table it determines that the next-hop router is B. The IP packet is encapsulated by AAL 5 and then it is segmented to an integer number of 48-byte blocks, each of which is carried in the payload of an ATM cell. The cells are forwarded to B over the PVC between A and B. At IP router B, the cells are re-assembled into the original AAL 5 PDU from which the original IP packet is extracted. B uses the IP address of the packet in its forwarding table and determines the next-hop IP router, which is C. The packet is subsequently encapsulated by AAL 5, and the resulting ATM cells are transmitted to C, where once more they get assembled back to the original IP packet.

IP switching can by-pass the table look-up in the forwarding routing table of an IP router. We note that this table look-up was a time-consuming operation at the time that IP switching was developed. This is not the case anymore, since faster table look-up algorithms were developed in the meantime.

In the above example, IP switching will forward the cells as they arrive at the ATM switch of IP router B directly to IP router C, without having to re-assemble the cells into the original IP packet, and then carry out a forwarding routing decision at IP router B. This is not done for all IP packets. Rather, it is done for IP packets that the router believes that they are part of an *IP flow*. This is a sequence of IP packets from a source IP address to a destination IP address. It can be identified by the pair <source IP address, destination IP address>, or from a more detailed set of parameters, such as <source IP address, source port number, destination IP address, destination port number>. This set of parameters used to identify an IP flow is referred to as the IP flow id. Using these parameters, the IP router can decide whether a particular IP packet it has received is an isolated packet, or whether it is the beginning of a sequence of IP packets. For instance, a DNS query will give rise to one or two IP packets, whereas an FTP is likely to give rise to a sequence of IP packets. Of course, there is no guarantee that an IP router will always guess correctly. For instance, it

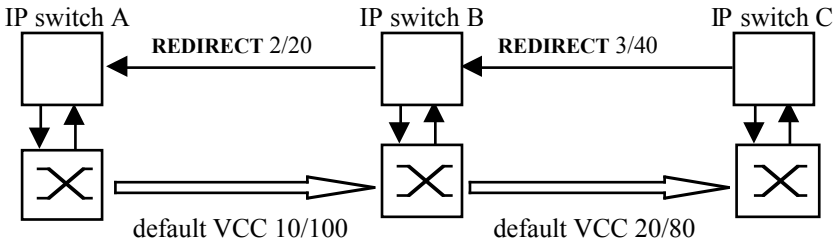


Figure 8.14: IP switching

may determine that an IP packet is the beginning of an IP flow, but this packet may turn out to be just an isolated packet.!

IP switching is implemented through *IP switches*. An IP switch is a general purpose computer which is attached to an ATM switch, and which runs the control and forwarding functions typically found in an IP router. In addition, it runs two protocols associated with IP switching, namely the *Ipsilon flow management protocol* (IFMP) and the *general switch management protocol* (GSMP). The functionality of these two protocols is explained below.

Two adjacent IP switches are connected over ATM via a default PVC. This connection is used to send control traffic, such as IP routing updates and IFMP messages. The same connection is also used to transmit ATM cells between the two IP switches. IP packets are first encapsulated using LLC/SNAP and then they are transported over the ATM network using AAL 5. The LLC/SNAP encapsulation is merely used to indicate which layer 3 protocol is used.

In order to explain the basic principle of IP switching, let us consider three IP switches, namely A, B, and C, connected as shown in figure 8.14. A and B communicate via the default VCC 10/100, and B and C communicate via the default VCC 20/80. We assume that A sends traffic to B which is then forwarded to C. All the cells transmitted from A to B have the virtual channel identifier 10/100. These cells are forwarded by the ATM switch to IP switch B. There, they get assembled back to the original AAL 5 PDUs, from which the IP packets are extracted. The IP switch looks up its forwarding table to identify the packet's next hop, and subsequently the packet is encapsulated and transmitted out to C in a sequence of ATM cells. These VPI/VCI of these cells is 20/80.

Now, let us assume that IP switch B decides that a particular IP packet with an IP flow id x is the beginning of a new flow. B sends a REDIRECT message to A, requesting A to send the cells of the IP packets belonging to this flow x on a new VPI/VCI, say 2/20. These cells still travel over the same link between A and B, but they are now distinguishable by their VPI/VCI. When these cells arrive at B's ATM switch, they are still forwarded to the IP switch where as before they are assembled into their original IP packets. The IP prefix is extracted and B looks it up in its forwarding table to decide the next hop of the IP packet. The packet is then transmitted to C over the ATM network. So, far nothing has changed from the way IP switch B routes its packets.

Now, let us assume that C also decides to redirect IP flow x. It sends a REDIRECT message to B, asking B to send the ATM cells belonging to the IP packets with flow id x on a new VPI/VCI, say 3/40. These cells still travel over the same link from B to C together with the rest of the traffic that B sends to C. However, they are distinguishable by their VPI/VCI.

At that moment, B recognizes that flow x has already been redirected from A to B on VPI/VCI 2/20. Therefore, it instructs its ATM switch to switch these cells directly through to the output port connecting to IP switch C, but with the VPI/VCI label set to 3/40. In view of this, all the cells belonging to the IP packets with flow id x, simply cut through B's ATM switch without having to be forwarded to the IP switch.

If there is more than one IP switch between A and C, eventually all the IP switches will redirect the ATM cells associated with flow id x. As a result, these ATM cells will simply cut-through all the ATM switches associated with these IP switches, without ever being forwarded to any of the intermediate IP switch.

We observe that IP switching is *data-driven*. That is, a flow of IP packets has to be identified first before a cut-through is set-up. A redirect message is always sent by the IP switch which is downstream as far as the data flow is concerned. For instance, in the above example, it is IP switch C that request B to redirect the flow, and it is B that requests A to redirect the flow. As we will see in the next section, this is known in tag switching as *downstream allocation*.

Ipsilon flow management protocol (IFMP)

This protocol runs between two IP switches and it is used to communicate the redirection of an IP flow to a new VPI/VCI, otherwise known as the *label binding information*. It uses the default PVC and it is a *soft-state* protocol, that is, the state that it sets automatically times-out, unless it is refreshed. In view of this, the flow-binding information has a limited life, once it is learned by an upstream IP switch and it must be refreshed periodically as long as it is necessary. The messages that install flow states contain a life time field, which indicates for how long that state is to be considered valid. One advantage of this approach, is that when the IP flow is finished, there is no need to cancel the binding upstream.

IFMP also provides an adjacency protocol, which can be used to identify immediate neighbours and also make sure that a neighbour is alive.

The following five message are used by IFMP:

REDIRECT: Used to bind a VPI/VCI label to an IP flow.

RECLAIM: Used to release a VPI/VCI label for subsequent re-use.

RECLAIM ACK: Used to acknowledge that a RECLAIM message was received and processed.

LABEL RANGE: Used by an IP switch to communicate the acceptable range of VPI/VCI labels to its neighbours.

ERROR: Used for various error conditions.

General switch management protocol (GSMP

This protocol is used to control the ATM switch to which the IP switch is attached. The GSMP is a master/slave protocol, where the ATM switch is the slave and the IP switch is the master. The two systems communicate via an ATM link. The protocol allows the master to establish and release connections across the switch, add/delete leaves to point-to-multipoint connection, perform port management, and request statistics and configuration information. GSMP has an adjacency component and a connection management component.

8.6 Tag switching

This is a label switching technique proposed by CISCO, and it has been standardized by IETF under the name of multi-protocol label switching (MPLS). In tag switching, a label is known as *tag*, which explains its name. Tag switching was developed primarily for IP networks, but it has also been applied to ATM networks.

An IP router implements both control and forwarding components. The control component consists of routing protocols, such as OSPF, BGP, and PIM, used to construct routes and exchange routing information among IP routers. This information is used by the IP routers to construct the forwarding routing table, referred to as the *forwarding information base* (FIB). The forwarding component consists of procedures that a router uses to make a forwarding decision on an IP packet. For instance, in unicast forwarding,

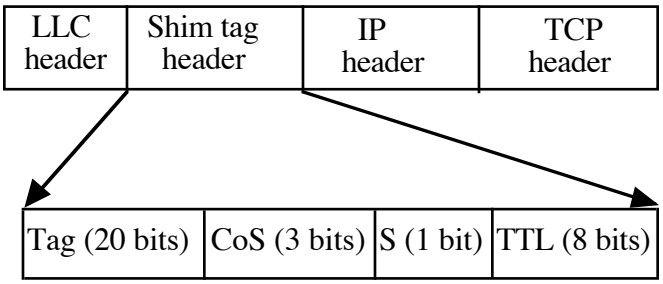


Figure 8.15: The shim tag header

the router uses the destination IP address to find an entry in the FIB, using the longest match algorithm. The result of this table look-up is an interface number, which is the output port connecting the router to the next-hop router, to which the IP packet should be sent.

A router forwards an IP packet according to its prefix (see section 2.8.2). In a given router, the set of all addresses that have the same prefix, is referred to as the *forwarding equivalent class* (FEC), pronounced as *fek*. IP packets belonging to the same FEC have the same output interface. In tag switching, it is a FEC that is associated with a tag. This tag is used to determine the output interface of an IP packet without having to look-up its address in the FIB.

In IPv6, the tag can be carried in the flow label field. In IPv4, however, there is no space for such a tag in the IP header. If the IP network runs on top of an ATM network, the tag is carried in the VPI/VCI field of an ATM cell. If it is running over frame relay,

the tag is carried in the DLCI field. For Ethernet, token ring, and point-to-point connections running a link layer protocol such as PPP, the tag is carried in a special *shim* tag header, which is inserted between the LLC header and the IP header, as shown in the figure 8.15. The first field of the shim tag header is a 20-bit field used to carry the tag. The second field is a 3-bit field used for the *class-of-service* (CoS) indication. This field is used to indicate the priority of the IP packet. The S field is used in conjunction with the tag stack. Finally, the *time-to-live* (TTL) field is similar to the TTL field in the IP header. The use of the CoS field and the tag stack will be explained later on in this section.

We recall that the label switching mechanism in IP switching is data-driven. That is, it is triggered by the arrival of an IP packet which is deemed to be the beginning of a sequence of packets. Tag switching, on the other hand, is *control-driven*. That is, it is triggered when a router discovers a new FEC.

A tag switching network consists of *tag edge routers* (TER) and *tag switching routers* (TSR). A TER has the same functionality as a regular IP router, and in addition it

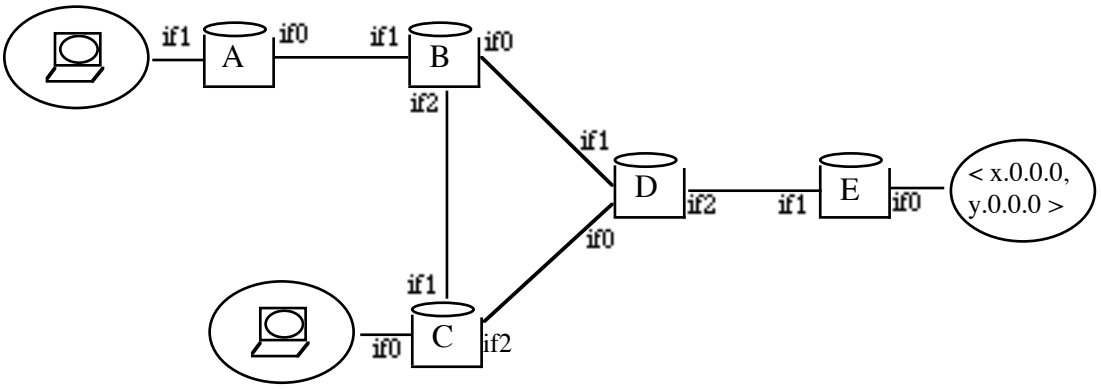


Figure 8.16: An example of tag switchinmg

can bind tags to FECs. A TSR can only bind tags to FECs, and it cannot forward IP packets by carrying out a table look-up in the FIB.

To see how tag switching works, let us consider a network consisting of 5 TSRs, A, B, C, D, and E, linked with point-to-point connections as shown in figure 8.16. A, C, and E are in fact tag edge routers. We assume that a new set of hosts with the prefix <x.0.0.0, y.0.0.0>, where x.0.0.0 is the base network address and y.0.0.0 is the mask, is directly connected to E. The flow of IP packets with this prefix from A to E is via B and

D. That is, A's next-hop router for this prefix is B, B's next-hop router is D, and D's next-hop router is E. Likewise, the flow of IP packets with the same prefix from C to E is via D. That is, C's next-hop router for this prefix is D, and D's next-hop router is E. The interfaces in figure 8.16 show how these routers are interconnected. For instance, A is connected to B via if0, B is connected to A via if1, to C via if2 and to D via if0, and so on.

When a TSR identifies the FEC associated with this new prefix <x.0.0.0, y.0.0.0>, it selects a tag from a pool of free tags and it makes an entry into a table known as the *tag forward information base* (TFIB). This table contains information regarding the incoming and outgoing tags associated with a FEC and the output interface, i.e. the FEC's

TSR	Incoming tag	Outgoing tag	Next hop	Outgoing interface
A			TSR B	if0
B	62		TSR D	if0
C			TSR D	if2
D	15		TSR E	if2
E	60		TSR E	if0

Table 8.1: FEC entry in each TFIB

next-hop router. The TSR also saves the tag in its FIB in the entry associated with the FEC.

The entry in the TFIB associated with this particular FEC for each TSR is shown in table 8.1. (For presentation purposes we have listed all the entries together in a single table). We see that B has selected an incoming tag equal to 62, D has selected 15, and E has selected 60. A and C have not selected an incoming tag for this FEC, since they are tag edge routers and they do not expect to receive tagged IP packets. The remaining information in each entry gives the next-hop router and the output interface for the FEC. For instance, for this FEC the next-hop router for A is B and it is through if0.

An incoming tag is the tag that a TSR expects to find in all the incoming IP packets that belong to a FEC. For instance, in the above example, TSR B expects all the incoming IP packets belonging to the FEC associated with the prefix <x.0.0.0, y.0.0.0> to

be tagged with the value 62. The tagging of these packets has to be done by the routers which are upstream of B. That is, they are upstream in relation to the flow of IP packets associated with this FEC. In this example, the only router that is upstream of B is A. In the case of D, both B and C are upstream routers.

In order for a TSR to receive incoming IP packets tagged with the value that it has selected, the TSR has to notify its neighbours about its tag selection for a particular FEC. In the above example, TSR B sends its information to A, D, and C. A recognizes that it is upstream from B, and it uses the information to update the entry for this FEC in its TFIB. D and C are not upstream from B as far as this FEC is concerned, and they do not use this information in their TFIBs. However, they store it for future use. It is possible for instance, that due to failure of the link between C and D, B becomes the next-hop router for this FEC. In this case, C will use the tag advertised by B to update the entry in its TFIB.

D sends its information to B, C, and E. Since B and C are both upstream of D, they use this information to update the entries in their TFIB. Finally, E sends its information to

TFIB	Incoming tag	Outgoing tag	Next hop	Outgoing interface
A		62	TSR B	if0
B	62	15	TSR D	if0
C		15	TSR D	if2
D	15	60	TSR E	if2
E	60		TSR E	if0

Table 8.2: FEC entry in each TFIB with tag binding information

D, which uses it to update its entry in its TFIB. As a result, each entry in the TFIB of each TSR will be modified as shown in table 8.2.

We note that at E, there is no next-hop router for the FEC associated with the prefix <x.0.0.0, y.0.0.0>. The IP packets associated with this FEC are forwarded to a local destination over if0.

Now, once the tags have been distributed and the entries have been updated in the TFIBs, the forwarding of an IP packet belonging to the FEC associated with the prefix <x.0.0.0, y.0.0.0> is done using solely the tags. Let us assume that A receives an IP packet from one of its local hosts with a prefix <x.0.0.0, y.0.0.0>. A identifies that the packet's IP address belongs to the FEC, and it looks up its TFIB to obtain the tag value and the outgoing interface. It creates a shim tag header, sets the tag value to 62, and forwards it to the outgoing interface if0. When the IP packet arrives at TSR B, its tag is extracted and looked up in B's TFIB. The old tag is replaced by the new one, which is 15, and the IP packet is forwarded to interface if0. TSR D follows exactly the same procedure. When it receives the IP packet from B, it replaces its incoming tag with the outgoing tag, which is 60 and forwards it to interface if2. Finally, E forwards the IP packet to its local destination. The same procedure applies for an IP packet with a prefix <x.0.0.0, y.0.0.0> that arrives at C.

In figure 8.17, we show the tags allocated by the TSRs. The sequence of tags 62, 15, 60 can be seen as being analogous to the VPI/VCI values allocated on each hop in an ATM VC connection. This sequence of tags can be seen as forming a path, referred to as the *tag switched path*, that resembles a point-to-point VC connection. An ATM connection is associated with two end-devices, whereas a tag switched path is associated with a FEC. Typically, there may be several tag switched paths associated with the same FEC which form a tree, as shown in figure 8.17.

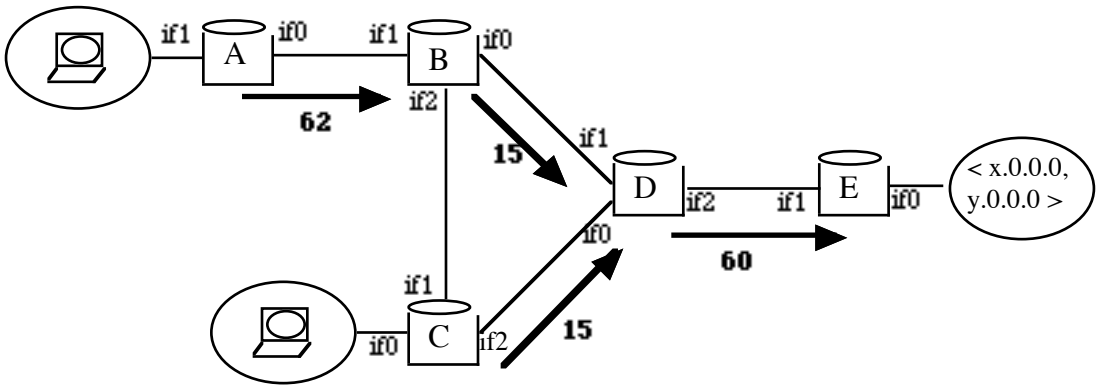


Figure 8.17: Tag switched paths

Tag switching eliminates the CPU-intensive table look-up in the FIB, necessary to determine the next-hop router of an IP packet. A table look-up in the TFIB is not as time-consuming since a TFIB is considerably smaller than a FIB. Since the introduction of tag switching, however, several CPU-efficient algorithms for carrying out table look-ups in the FIB were developed. This did not diminish the importance of tag switching since it was seen as a means of introducing quality of service in the IP network. This is done by associating a priority for each IP packet that gets tagged by a tag edge router (TER). The priority is carried in the class of service (CoS) field shown in figure 8.15.

Tagged IP packets within a TSR are served according to their priority as in the case of an ATM switch. We recall that in ATM networks, each VC connection is associated with a quality-of-service category. An ATM switch can determine the quality-of-service of an incoming cell from its VPI/VCI value, and accordingly it can queue the cell into the appropriate quality-of service queue. As we have seen in section 6.7, an ATM switch maintains different quality-of-service queues for each output buffer. These queues are served using a scheduling algorithm, so that VC connections can be served according to their requested quality of service. A similar queueing structure can now be introduced into an IP router.

Another interesting feature of tag switching is that it can be used to create a dedicated route, known as an *explicit route*, between two IP routers. Explicit routing is described below.

Tag allocation

In the example described above, a TSR allocates a tag for a FEC and saves this information in its TFIB as the incoming tag. It then advertises the binding between the incoming tag and the FEC to its neighbouring TSRs. This information could be carried by either piggy-backing it to a routing protocol or using the *tag distribution protocol* (TDP). When a TSR receives this information, it places the tag in the outgoing tag field of the entry in its TFIB that is associated with this FEC. In view of the fact that this information is generated by the TSR which is at the downstream end of the link, with respect to the flow of the IP packets, this type of tag allocation is known as *downstream tag allocation*.

IP switching also uses downstream allocation. In addition to this scheme, tags can be allocated using *downstream tag allocation on demand*, and *upstream tag allocation*.

In downstream tag allocation on demand, each TSR allocates an incoming tag to a FEC and creates an appropriate entry in its TFIB. However, it does not advertise its tag to its neighbours as in the case of downstream allocation. Instead, an upstream TSR obtains the tag information by issuing a request via TPD.

Upstream tag allocation, as its name implies, works in the opposite direction to downstream allocation. When a TSR discovers a new FEC it selects a tag and it creates an appropriate entry in its TFIB. This tag is the outgoing tag for the FEC, rather than the incoming tag as in the case of downstream tag allocation. The TSR advertises its tag for this FEC to its neighbouring TSRs. A downstream TSR will populate the incoming tag field of its entry associated with the FEC with the advertised tag. If it is not a downstream TSR, it will simply store this information.

Tag stack

The IP routing architecture consists of a collection of routing domains. Intra-domain routing, i.e., routing within a domain, is provided via an interior routing protocol, such as OSPF. Inter-domain routing, i.e. routing in-between domains, is provided by an exterior routing protocol, such as BGP. TSRs allow the decoupling of interior and exterior routing information, so that only the TSRs at the border of a domain are required to maintain routing information provided by the exterior routing protocol. TSRs within a domain maintain only routing information provided by the domain interior routing. (This is not currently the case in IP networks.)

To support this functionality, tag switching allows an IP packet to carry multiple tags organized as a stack. When a packet is forwarded from a border TSR of one domain to a border TSR of another domain, the tag stack contains a single tag. However, when the packet is forwarded within a domain, it contains two tags. The tag at the top of the stack is used for tag switching within the interior TSRs so that the packet is forwarded to the egress border TSR. The tag in the lower level is used by the egress border TSR to forward the packet to the next border TSR.

Explicit routing

As we have discussed above, a router makes a forwarding decision by using the IP address in its FIB in order to determine the next-hop router. Typically, each IP router calculates the next-hop router for a particular destination using the shortest path algorithm. Tag switching follows the same general approach, only it uses tags. This routing scheme is known as *destination-based* routing.

An alternative way of routing a packet is to use *source routing*. In this case, the originating (source) TSR selects the path to the destination TSR. Other TSRs on the path simply obey the source's routing instructions. Source routing can be used in an IP network to evenly distribute traffic among links, by moving some of the traffic from highly utilized links to less utilized links. Tag switching can be used to set-up such routes, which are known as *explicit routes*. The creation of an explicit route is done using RSVP.

Tag switching over ATM

In this scheme, the ATM user plane of an ATM switch remains intact. However, the ATM signalling protocols, such as Q.2931 and PNNI, are replaced by IP protocols such as OSPF, BGP, PIM and RSVP. Such an ATM switch which can also run tag switching is referred to as an ATM-TSR.

In tag switching over ATM, there is no shim tag header, as is the case when tag switching is implemented over Ethernet or token ring or a point-to-point connection. The tag is carried in the VCI field of the cell. If a tag stack is used then up to two tags can be carried, one in the VCI field and the other in the VPI field. A predefined VC connection is used for exchanging tag binding information.

Downstream allocation on demand is used. That is, when an ATM-TSR identifies a new FEC it selects an incoming tag, but it does not advertised it to its neighbours. An upstream ATM-TSR obtains the tag bound to the FEC by issuing a request. Eventually, the connection will be set-up from an ATM-TSR that acts as an edge TSR to the edge TSR that serves the hosts associated with the FEC.

An interesting problem that arises in tag switching over ATM is *VC merging*. This problem arises in destination-based routing when two ATM-TSRs are both connected to

the same downstream ATM-TSR. Let us consider the four ATM-TSRs A, B, C, and D shown in figure 8.18, and let us assume that the flow of IP packets for a specific FEC is from A to C and then to D, and from B to C and then to D. We assume that D is the edge TSR that serves the hosts associated with the FEC. The allocated tags are shown in figure 8.18 in bold. Now let us see what happens when A has an IP packet, call it packet 1, to transmit that belongs to this FEC. This packet will be encapsulated by AAL 5 and then it will be segmented into an integer number of 48-byte blocks. Each block will then be carried in the payload of an ATM cell tagged with the value 15. The cells will be transmitted to C, where they will have their tag changed to 20 and then they will be forwarded to the buffer of the output port that connects to D. In this buffer, it is possible that these cells will get interleaved with the cells belonging to an IP packet, call it packet 2, associated with same FEC and transmitted from B. That is, as the cells are queued-up into the buffer, cells belonging to packet 1 may find themselves in-between two successive cells belonging to packet 2. Since all these cells will be send to D with the tag of 20, D will not be able

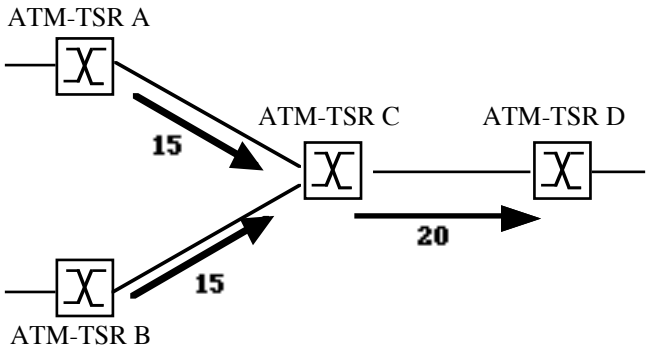


Figure 8.18: VC merging

to identify which of these cells belong to packet 1 or packet 2, and consequently it will be not be able to reconstruct the original AAL 5 PDUs.

A simple solution to this problem is to first collect all the cells belonging to the same IP packet in the buffer of the output port of C. Once all the cells have arrived, then they can be transmitted out back-to-back to D. In order to do this, it will be necessary for

the switch to be able to identify the beginning cell and last cell of an AAL 5 PDU. The mechanism to do this may be in place if the early packet discard and partial packet discard policies have been implemented in the switch (see section 7.7.2). An alternative solution is to use multiple tags so that the path from A to D is associated with a different set of tags than the path from B to D.

8.7 Multi-protocol label switching (MPLS)

MPLS is an IETF standard based on tag switching. The original intention was to be used in conjunction with different networking protocols, such as IPv4, IPv6, IPX and AppleTalk. However, MPLS has been developed exclusively for IP networks, which makes the description of the protocol as being "multi-protocol" more general than it actually is.

Basic features of the MPLS architecture

The main architecture of MPLS is the same as tag switching. A tag in MPLS is called a *label*, a tag switching router (TSR) is called a *label switching router* (LSR), a tag edge router (TER) is called a *label edge router* (LER), and an ATM TSR is called ATM-LSR. Finally, a tag switched path is referred to as a *label switched path* (LSP).

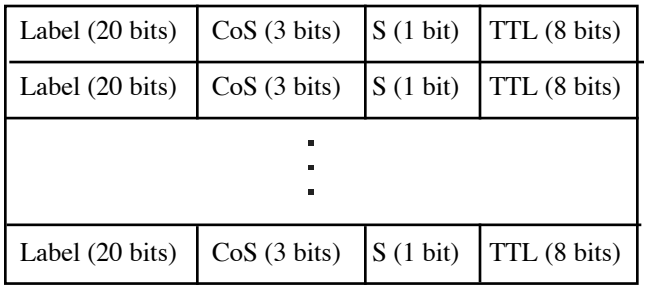


Figure 8.19: The label stack

An LSR can only perform label-based functions. An LER terminates and originates LSPs, and it performs both conventional IP router functions and label-based functions. On the ingress to an MPLS domain, an LER accepts unlabelled IP packets and creates an initial MPLS label stack consisting of one or more shim label headers. On the egress of an LSP, an LER terminates and forwards the IP packet based on their IP

addresses. A hybrid LSR originates and terminates some LSPs while at the same time it acts as an LSR for other LSPs.

Both destination-based routing, called in MPLS *hop-by-hop*, and explicit routing is allowed. Explicit routing is referred to as *constraint routing*. This is because the route is picked up using a constraint other than the number of hops, such as minimize congestion along the path. A constraint route may not necessarily be the shortest path.

Label allocation in MPLS is control-driven. That is, it is triggered when a router discovers a new FEC. (Label allocation can also be data-driven as in IP switching.) The allocation of labels can be done using downstream allocation or downstream allocation on demand. As in tag switching, the label is carried in a *shim label header* if the IP network runs over Ethernet or token ring or a point-to-point protocol such as PPP. The location and structure of the shim label header is the same as the shim tag header shown in figure 8.15. Multiple shim label headers can be stacked together in a *label stack*, as shown in figure 8.19. All shim label headers in a label stack have S=0, except the one in the bottom whose S=1.

When MPLS runs on top of ATM, the label is carried in the VCI/VPI field. The label stack is carried as part of the IP packet. That is, the block of information passed on to AAL 5, consists of the IP packet and the label stack placed in front of the IP header. The top label in the stack is the one used in the VCI/VPI field of the cell header. The remaining labels can be used in case where the MPLS domain is extended passed an ATM network into a non-ATM network which requires the use of the shim label header.

The distribution of labels can be done by piggybacking them on control protocols, such as BGP, PIM, and RSVP. In addition, a new protocol for the distribution of labels, the *label distribution protocol* (LDP) has been developed by IETF.

The label distribution protocol (LDP)

For reliability purposed, the LDP protocol runs over TCP. Two LSRs which use LDP to exchange label-binding information are known as *LDP peers*. In order for two LDP peers to be able to exchange information, they have to establish an *LDP session* between them.

There are four categories of LDP messages: *discovery* messages, *session* messages, *advertisement* messages, and *notification* messages. Discovery messages

provide a mechanism whereby LSRs indicate their presence in a network by sending hello messages periodically. Session messages are used to establish, maintain, and terminate an LDP session between LDP peers. Advertisement messages are used to create, change, and delete label mappings for FECs. Finally, notification messages are used to provide advisory information and signal error information.

All LDP messages have a common structure that uses the *type-length-value* (TLV) encoding. The type specifies how the value field is to be interpreted, the length gives the length of the value, and the value field contains the actual information. The value field may itself contain one or more TLVs. That is, TLVs may be nested.

For constraint routing, the label-binding information is distributed using RSVP or CR-LDP, an extension of LDP. CR-LDP can be used to trigger and control the establishment of an LSP between two LERs. *Strict* routing or *loose* routing can be used. In strict routing, all the LSRs through which the LSP must pass are indicated. In loose routing some LSPs are indicated, and the exact path between two such LSPs is determined using conventional routing based on IP addresses.

Problems

1. What was ATM Forum's motivation for creating LAN emulation?
2. Identify two problems that LAN emulation has to resolve in order to provide LAN services over ATM.
3. Explain the function of each of the control VCCs used in LAN emulation.
4. Explain the function of ATMARP.
5. Explain the function of MARS.
6. In IP and ARP over ATM, describe the sequence of actions that take place when a new leaf is added to a multicast group, assuming an MCS-based architecture.
7. What problem does NHRP address?
8. What is an IP flow?
9. Explain why IP switching is data-driven.
10. Explain the differences between downstream and upstream allocation.
11. In MPLS, what is the label stack used for? Give an example.
12. Explain the difference between destination-based, or hop-by-hop, routing and explicit routing.

.

CHAPTER 9

ADSL-Based Access Networks

In recent years, a number of different technologies have been developed with a view to providing high-speed access to residential users and small offices. Specifically, a family of technologies known as xDSL has been developed to provide access to the Internet over the telephone line, in addition to basic telephone services. Cable modems have also been developed to provide access to the Internet over the TV cable, in addition to TV channel distribution. In addition, wireless access technologies, such as *local multipoint distribution services* (LMDS), make use of new wireless spectrums to provide high-speed access. Recently, *ATM passive optical networks* (APON) have emerged as an alternative to providing high-speed access over fiber.

In this Chapter, we first review these access technologies, and then we present in detail the *asynchronous digital subscriber line* (ADSL) technology. We also discuss schemes for accessing *network service providers* (NSP) over ADSL.

9.1 Introduction

Cable TV has been widely deployed in USA. Typically, each home is wired with a telephone line and a TV cable. The telephone line is a twisted pair that terminates to the nearest telephone switch, referred to as the *central office* (CO). The TV cable is a coax cable that connects the home to the cable distribution network. The telephone and cable networks are two separate systems. One receives TV channels over the TV cable, and uses the telephone for telephone services and also to connect to the Internet over a modem. Due to the enormous business opportunities, both cable and telephone operators

are interested in providing a complete set of services over the same wire. Currently, telephone operators provide access to the Internet over the twisted pair in addition to basic telephone services. Video on demand will also be provided over the twisted pair in the future.

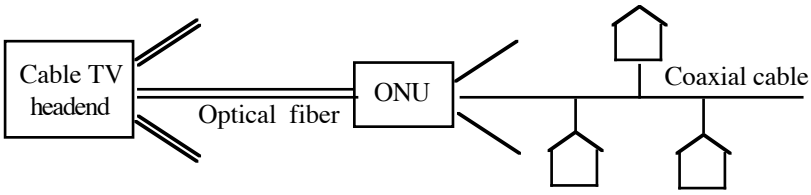


Figure 9.1: The HFC architecture

Cable operators provide access to the Internet over their coax TV cable distribution system in addition to the distribution of TV channels. It is expected that telephone services (voice over IP) will also be provided over the cable. The battle between the telephone and the cable operators for dominance in the access network area will only intensify!

A cable plant architecture consists of the headend, fiber trunks extending from the headend, and coaxial cables. The headend is the source of the TV channels that are distributed over the cable plant. The homes served by the cable plant are divided into small neighbourhoods of about 500 homes, and each neighbourhood is connected to the headend via a dedicated fiber. Each fiber extends from the headend and terminates at an *optical network unit* (ONU). From the ONU, a number of coaxial cables fan out into the neighbourhood, each serving a number of homes as shown in figure 9.1. Due to the combination of fiber optics and coaxial cables, this architecture is known as the *hybrid fiber coaxial* (HFC) architecture. Access to the Internet is provided over an HFC plant using cable modems. These modems convert data packets to analog signals which are transmitted over an analog path in the same way as analog video. Due to the enormous bandwidth of cable TV networks, access to the Internet can be achieved at speeds of 6 Mbps or higher. Moreover, existing cable TV systems have upstream channels built in for interactive services. These upstream channels can be used to provide a path to the IP network to which the headend is connected. The transmission of high-speed data over

cable systems is specified in the *data-over-cable service interim specification* (DOCSIS), Cable-based access networks have been designed to transport IP traffic.

ADSL is one of the access technologies that can be used to convert the telephone line into a high-speed digital link. It is part of a family of technologies called the *x-type digital subscriber line* (xDSL), where x stands for one of several letters of the alphabet and it indicates a different transmission technique. Examples of the xDSL family technologies are: *asymmetric DSL* (ADSL), *high data rate DSL* (HDSL), *symmetric DSL* (SDSL), *ISDN DSL* (IDSL), and *very high data rate DSL* (VDSL). Some of the

Name	Downstream/upstream rate	Use
ADSL	8.128 Mbps/ 800 Kbps	Data
HDSL	1.544 /2.048 Mbps	T1/E1 replacement
SDSL	768 Kbps	Fractional T1/data
ISDL	128 Kbps	Data
VDSL	52 Mbps/6 Mbps	Video/data

Table 9.1: xDSL specifications

xDSL technologies use analog signalling methods to transport analog or digital information over the twisted pair, while others use true digital signaling to transport digital information. A list of specifications for the xDSL family technologies is given in table 9.1. In access networks, *downstream* means from the network to the user, and *upstream* means from the user to the network.

VDSL, as its name implies, achieves very high data rates over the twisted pair. However, the distance over which such rates can be transported is limited. Currently, it can achieve a downstream data rate of 52 Mbps and an upstream data rate of 6.4 Mbps over a distance of up to 1,000 feet. For the same distance, it can also provide symmetric rates of 26 Mbps downstream and 26 Mbps upstream. The longest distance it can be transported is currently 5,000 feet for which it can achieve 13 Mbps downstream and 1.6 Mbps upstream. VDSL can be used to deliver high quality video together with access to the Internet and regular telephone services. Because of the distance limitation, it is

envisioned that it will be used to deliver information from a cabinet in the street which is connected to an access network via optical fibers.

Recently, ATM passive optical networks (APON) have emerged as an alternative to providing high-speed access over fiber. APONs were standardized by ITU-T in 1998 (ITU-T Recommendation G.983.1). Also, the *full service access network* (FSAN) consortium identified the APON as the most cost-effective architecture for high-speed access networks. The FSAN consortium is a group of telecommunication operators and manufacturers which works towards developing a consensus on the systems required to deliver a full set of telecommunication services over an access network. The APON, as its name implies, supports the ATM architecture. It can provide high-speed access to the Internet, voice over packet, and video services equivalent to those provided by a cable operator.

An APON consists of *optical network units* (ONU) which are connected to an *optical line terminator* (OLT) via a passive optical network. The OLT is connected to an

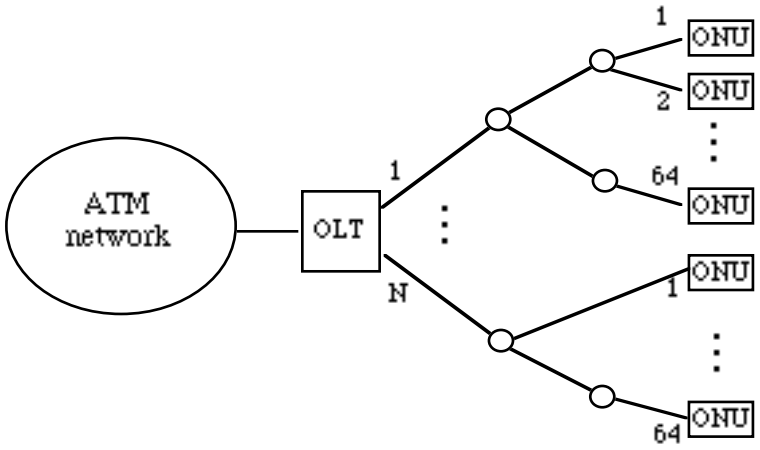


Figure 9.2: The APON architecture (FTTH)

ATM network, and it provides and receives traffic to/from the ONUs. The passive optical network is a point-to-multipoint network with the OLT as the root and the ONUs as the leaves. As shown in figure 9.2, there are N different fibers fanning out from the OLT. Each fiber is split into multiple fibers using passive optical splitters, shown as circles, so

that it can support a maximum of 64 ONUs. The maximum length between the OLT and an ONU is 30 miles.

There are several configurations, depending upon where the ONUs are located. If each ONU is located inside the home, then the configuration is known as *fiber to the home* (FTTH). Alternatively, the ONUs may be located in a cabinet in the street or in the basement of a building. Interconnectivity between the ONU and the homes can be provided using VDSL over copper. This configuration is known as *fiber to the curb* (FTTC) or *fiber to the basement* (FTTB) or *fiber to the cabinet* (FTTCab). These configurations differ only in the way in which they are implemented, and from the point of view of the G.983.1 standard, they are all the same.

The point-to-multipoint passive optical network supports bi-directional transmission using *wavelength division multiplexing* (WDM). Two wavelengths are used, one for downstream transmission and one for upstream transmission. In the downstream direction, the operating wavelength range on a single fiber is from 1480 to 1580 nm. In the upstream direction the operating wavelength range is from 1260 to 1360 nm. The optical signal transmitted by the OLT is propagated on all the N fibers, and at each splitter, the incoming optical signal is split as many times as the number of outgoing fiber. Eventually, all the ONUs will receive the same optical signal that was transmitted by the OLT. A time division multiplexing scheme is used that enables the APON user to transmit upstream. Vendors may utilize additional wavelengths for video services.

The transmission rates in the APON could be symmetric or asymmetric. In the symmetric case, the nominal rate is 155 Mbps in both the downstream and upstream direction. In the asymmetric case, the nominal rates are 622 Mbps in the downstream direction and 155 Mbps in the upstream direction.

9.2 The ADSL technology

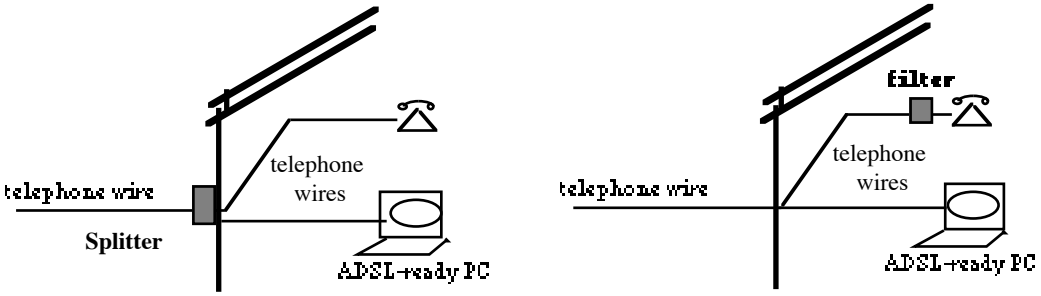
The asymmetric digital subscriber line (ADSL) technology utilizes the existing twisted pair from the central office to the home to transport data in addition to the basic telephone services. It was originally designed to provide video on demand services transported over switched DS-1 or E1. This type of traffic is referred to in the ADSL standard as the *synchronous transfer mode* (STM) traffic. In its current standard (ANSI T1.413 issue 2 or

ITU-T G.992.1) full rate ADSL has been defined to carry either ATM or STM traffic or both. ADSL is primarily used for ATM traffic, and there is a limited number of applications for STM traffic.

As its name implies, ADSL provides asymmetrical data rates with the downstream rate being considerably higher than the upstream rate. The data rate depends on the length of the twisted pair, the wire gauge, presence of bridged taps, and cross-couple interference. Ignoring bridged taps, currently ADSL can deliver a full DS-1 or E1 signal downstream over a single unloaded 24 gauge twisted pair for a maximum distance of 18,000 feet. Up to 6.1 Mbps is possible for a maximum distance of 12,000 feet, and 8.128 Mbps for a maximum distance of 9,000 feet. Upstream data rates are presently in the 64 to 800 Kbps range.

The deployment of ADSL over the twisted pair, requires an ADSL transmission unit at either end of the line. The ADSL transmission unit at the customer premises is referred to as the *ADSL transceiver unit, remote terminal* (ATU-R), and the ADSL transmission unit at the central office is referred to as the *ADSL transceiver unit, central office (*ATU-C).

The signal that arrives at the ATU-R over the twisted pair, contains both ADSL data and voice. It is necessary to split this signal and deliver the voice signal to the telephone sets and the ADSL signal to a PC. This can be achieved using a splitter as shown in figure 9.3a. This solution utilizes the fact that category 3 or 5 telephone wires used in a home typically consist of 2 sets of wires. (They may also consist of 3 sets of wires). Voice



a) Splitter **b) Splitterless**

Figure 9.3: Two possible solutions at the customer's premises

is usually carried on the inner pair. The splitter splits the incoming signal, so that the voice is directed to the inner pair and the ADSL data to the outside pair. This is achieved using passive elements that simply perform a low pass filter that directs voice signals to the inner pair. The high frequency ADSL traffic is coupled via a transformer to the outer pair. A phone, or a fax machine, or an answering machine, can be plugged into any telephone plug in the house. The PC used to access the Internet has to have a built-in or an external ATU-R that permits it to receive and transmit ADSL data, and it can be plugged into any telephone plug. If the quality of the twisted pair in a home may not be good, a dedicated high-quality line is installed from the splitter to the ATU-R.

The installation of the splitter requires a visit by a technician. An alternative solution, is to use *splitterless* ADSL, as shown in figure 9.3b. Splitterless ADSL does not require an installation by a technician, and consequently it is cheaper to deploy than the ADSL modem that requires a splitter. In this case, the signal transmitted over the twisted pair, which contains both ADSL data and voice, is propagated throughout the home telephone wires. The voice signal is filtered out using a high pass filter inside the ATU-R. On the other hand, the ADSL signal can cause a strong audible noise through the telephone set. Therefore, each phone is attached to a telephone plug through a filter, which filters out the ADSL signal and at the same time it isolates voice events, such as ring and on/off hook, from the ADSL signal. The PC can be plugged in to any telephone plug.

The splitterless ADSL standard is referred to as G.LITE (ITU-T recommendation G.992.2). G.LITE is low-speed splitterless ADSL solution, with a downstream data rate of 1.536 Mbps and an upstream rate of 512 Kbps. Current developments in silicon migration, better power management techniques, and splitterless implementation, has permitted the full rate ADSL to be also used in a splitterless mode. As a result, in most countries the G.LITE solution never became popular.

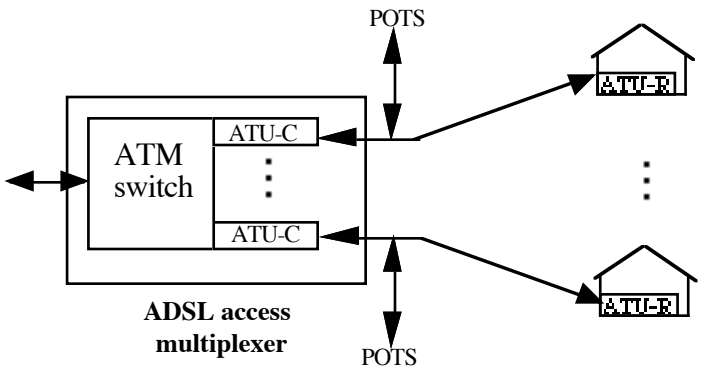


Figure 9.4: The ADSL access multiplexer (DSLAM)

Now, let us take a look at the ATU-C at the central office. As shown in figure 9.4, in the downstream direction the voice signal is added after the ADSL signal leaves the ATU-C. In the upstream direction, the voice signal is extracted from the ADSL signal, before the ATU-C. The ATU-C generates the ADSL signal in the downstream direction, and terminates the ADSL signal in the upstream direction. A number of ATU-Cs are serviced by an *ADSL access multiplexer*, known as DSLAM, which provides connectivity to IP and ATM networks.

The DSLAM is in fact an ATM switch. It typically has an OC-3 or higher link to an ATM access backbone network, and it also has ADSL links serving a number of customer premises. Each ADSL link at the DSLAM is associated with an ATU-C, which is the physical layer associated with the link (see section 4.5).

We now proceed to describe how an ATU-C or an ATU-R works. The protocols used to provide IP and ATM services over ADSL are described in section 9.3.

9.2.1 The discrete multi-tone (DMT) technique

The *discrete multi-tone* (DMT) technology is the standardized line coding technique used for ADSL. DMT devices can easily adjust to changing line conditions, such as moisture or interference, and it is resistant to noise and the presence of digital signals or adjacent wire pairs (cross-talk).

In the DMT technique, the entire bandwidth of the twisted pair is divided into a large number of equally spaced sub-channels, also known as *tones*. The twisted pair's

Bearer channel	Lowest required multiple	Largest required multiple	Corresponding highest data rate
AS0	1	192	6144 Kbps
AS1	1	144	4608 Kbps
AS2	1	96	3072 Kbps
AS3	1	48	1536 Kbps
LS0	1	20	640 Kbps
LS1	1	20	640 Kbps
LS2	1	20	640 Kbps

Table 9.2: Data rates for the bearer channels

bandwidth extends to 1.1 MHz, and it is divided to 256 sub-channels, each occupying 4.3125 KHz. The lower sub-channels 1 through 6 are reserved for the voiceband region and are used to provide basic telephone services. The remaining sub-channels are used by ADSL.

ADSL is bi-directional which means that both the upstream and downstream data is sent over the same twisted pair. In ADSL, bi-directional transmission over the twisted pair can be implemented using either *frequency division multiplexing* (FDM) or echo cancellation. In FDM, there are up to 32 upstream sub-channels, i.e., from the customer premises to the network, occupying the frequencies immediately above the voiceband region. Also, there are up to 218 downstream sub-channels, i.e., from the network to the customer premises, occupying the frequencies above the upstream sub-channels. An alternative solution is to let the upstream and downstream sub-channels use the same frequencies, and separate them using echo cancellation. Not all the sub-channels are used for the transfer of information. Some are used for network management and performance

monitoring. All sub-channels are monitored constantly for performance and errors and the speed of each sub-channel or group of channels can actually vary with a granularity of 32 Kbps.

Transmission is achieved by dividing time into fixed-sized intervals. Within each interval, DMT transmits a data frame which consists of a fixed number of bits. The bits in a data frame are divided into groups of bits and each group is transmitted over a different sub-channel. The number of bits sent over each sub-channel can be varied depending upon the signal and noise level in each sub-channel. Using the *quadrature amplitude modulation* (QAM) technique, the bits allocated to each sub-channel are converted into a

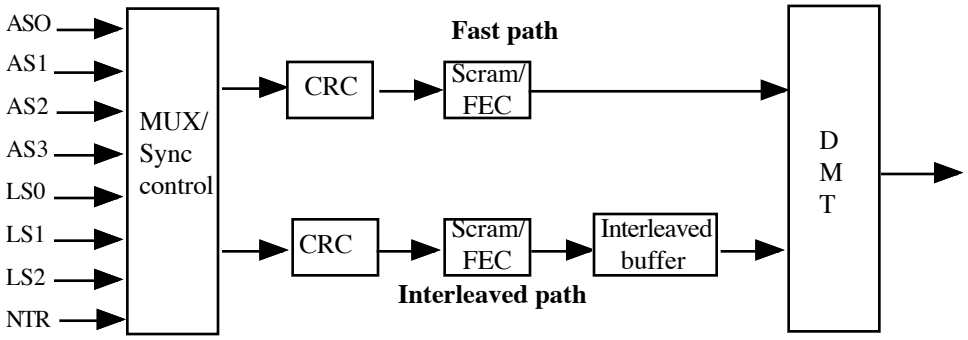


Figure 9.5: The fast path and the interleaved path in ATU-R

complex number which is used to set the sub-channel's amplitude and phase for the interval. The signals are all added up and sent to the twisted pair. This signal resulting from each data frame is known as the *DMT symbol*.

9.2.2 Bearer channels

A diagram of the ATU-R showing the flow of data in the downstream direction is given in figure 9.5. The flow of data in the upstream direction in the ATU-C has a similar structure. The data transported by an ATU-R or ATU-C is organized into 7 independent logical bearer channels. Of these 7 channels, 4 are unidirectional channels from the network to the customer premise. These four channels are referred to as the *simplex bearer channels*, and they are designated as AS0, AS1, AS2, and AS3. The remaining 3 channels are duplex, and they are referred to as the *duplex bearer channels*. They are bi-

directional channels between the network and the customer premise, and they are designated as LS0, LS1, LS2. The three duplex bearer channels may also be configured as independent unidirectional simplex channels. All bearer channels can be programmed to transmit at a speed which is an integer multiple of 32 Kbps, as shown in table 9.2. The maximum total data rate of the ADSL system depends on the characteristics of the twisted pair on which the system is deployed.

STM traffic is mapped in bearer channels AS0 and LS0 in the downstream direction, and in LS0 in the upstream direction. Other bearer channels can also be provisioned. ATM traffic is mapped in the downstream direction in bearer channel AS0 and in LS0 in the upstream direction. Other bearer channels can also be provisioned.

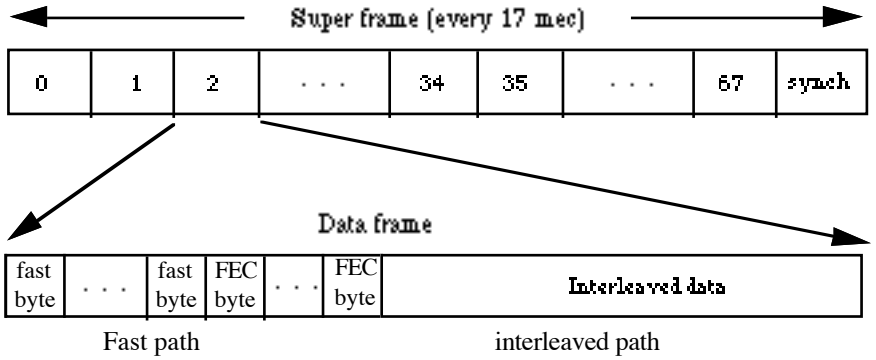


Figure 9.6: The super frame

Some applications running at the customer premises may require a reference clock. In view of this, in addition to transporting user data, an ATU-C may optionally transport a *network timing reference* (NTR) to an ATU-R.

A bearer channel in an ATU-R or ATU-C can be assigned either to the *fast path* or to the *interleaved path*. The two paths in the ATU-R are shown in figure 9.5. The fast path provides low delay, whereas the interleaved path provides greater delay but lower error rate. CRC, forward error correction (indicated in diagram 9.5 by the abbreviation FEC) and scrambling can be applied to each path. Bearer channel AS0 carrying downstream ATM traffic can be transmitted over either the fast path or the interleaved path. Upstream ATM data are transmitted in LS0 either over the fast path or the interleaved path. The bearer channels carrying STM traffic in either direction are

transmitted over either the fast path or the interleaved path. The choice between the fast path and the interleaved path in the downstream direction may be independent of that in the upstream direction

9.2.2 The ADSL super frame

As mentioned above, a data frame consists of a fixed number of bits and it is transmitted every fixed interval using the DMT technique. Each data frame combines bits interleaved from the fast and the interleaved paths. The data frames are combined into a super frame consisting of 68 data frames plus a synchronization data frame, as shown in figure 9.6. Each data frame is transmitted on the twisted pair as a DMT symbol. The rate of transmission of DMT symbols is 4000 symbol/sec. Since a synchronization data frame is transmitted for every 68 data frames, the transmission rate on the twisted pair is actually slightly higher, i.e., (69/68)*4000 symbols/sec. That is, the super frame repeats every 17 msec.

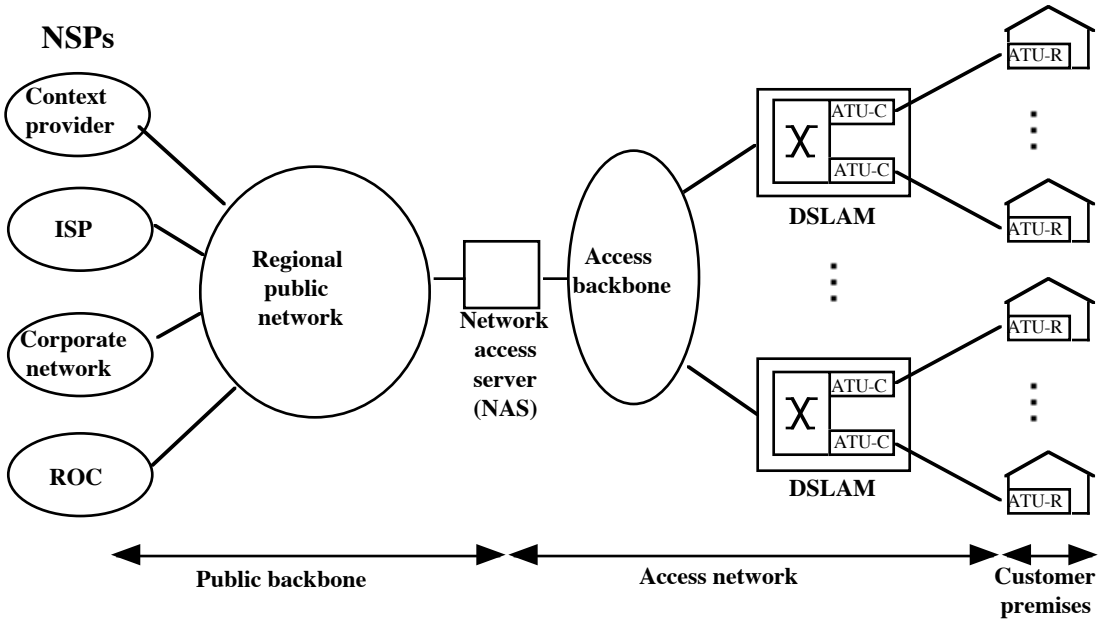


Figure 9.7: The ADSL reference architecture

9.3 Schemes for accessing network service providers

Network service providers (NSP) are content providers, Internet service providers, and corporate networks. Providing access to NSPs is an important service to the ADSL users. In this section, we describe two different schemes that can be used to provide connectivity to network service providers, namely the *L2TP access aggregation* scheme and the *PPP terminated aggregation* scheme.

The ADSL Forum's reference architecture, shown in figure 9.7, defines the connectivity between ADSL users and NSPs. This reference architecture consists of customer premises, an access network, a regional public network, and NSPs. The customer premises may be a residence, a home office, or a small business office. At a customer premise, there may be one or more computers interconnected by a network. The access network includes the ATU-Rs at the customer premises, the DSLAMs that serve these ATU-Rs, and an access backbone network that interconnects all the DSLAMs. Connectivity to NSPs and to a *regional operations center* (ROC) is provided through a regional public network. Typically, the access network is managed by a telephone operator, who controls it via an ROC. The telephone operator may be the local telephone operator known as the *incumbent local exchange carrier* (ILEC), or a national or newcomer telephone operator known as the *competitive local exchange carrier* (CLEC).

In most cases, the ATU-Rs do not have the necessary functionality to set up SVCs. Instead, PVCs are used. Providing each ATU-R with a PVC to each NSP requires a large number of PVCs to be set-up and managed. A more scaleable approach is to provide a *network access server* (NAS), as shown in figure 9.7. The role of the NAS is to terminate all the PVCs from the ATU-Rs and then aggregate the traffic into a single connection for each NSP.

ADSL users set up sessions to NSPs using the *point-to-point protocol* (PPP). This protocol was designed to provide a standard method for transporting datagrams from different protocols over a full-duplex link. PPP provides a number of functions, such as assignment of an IP address by a destination network, domain name auto-configuration, multiplexing of different network layer protocols, authentication, encryption, compression, and billing. PPP frames are transported using a default HDLC-like encapsulation. When PPP runs on top of ATM, PPP frames are mapped into AAL 5 PDUs using either the *VC-multiplexed PPP* scheme or the *LLC encapsulated PPP*

scheme. In the former scheme, a PPP frame is directly carried in an AAL 5 PDU. In the latter scheme, a PPP frame is also carried in an AAL 5 PDU after it is further encapsulated with a 2-byte LLC header and a 1-byte network layer protocol identifier.

9.3.1 The L2TP access aggregation scheme

This scheme is based on IETF's *layer 2 tunneling protocol* (L2TP). The protocol stacks involved in this scheme are shown in figure 9.8. For simplicity we assume that an ADSL user at a customer premises is a single computer, rather than a network of computers interconnected via an ATM network. An ADSL user is connected to the DSLAM over ADSL, and the DSLAM is connected to an NAS, referred to as the *L2TP access concentrator* (LAC), over an ATM network. Finally, the LAC is connected to the *L2TP network server* (LNS) of each NSP over a network, such as IP, frame relay, and ATM.

The ADSL user is connected to the LAC with an ATM PVC via the DSLAM. This connection uses AAL 5. The LAC and the LNS of an NSP are connected by an L2TP *tunnel*. An L2TP tunnel is not an actual connection in the sense of an ATM connection. Rather, it is a logical connection between the L2TP on the LAC and its peer L2TP on the LNS. A PPP session between the ADSL user and the LNS is established as follows. The ADSL user sends a request to the LAC over AAL 5 to initiate a PPP session to an LNS. This request is forwarded by the LAC to the LNS over an L2TP tunnel. Once the PPP session is established, IP packets can begin to flow between the ADSL user and the LNS.

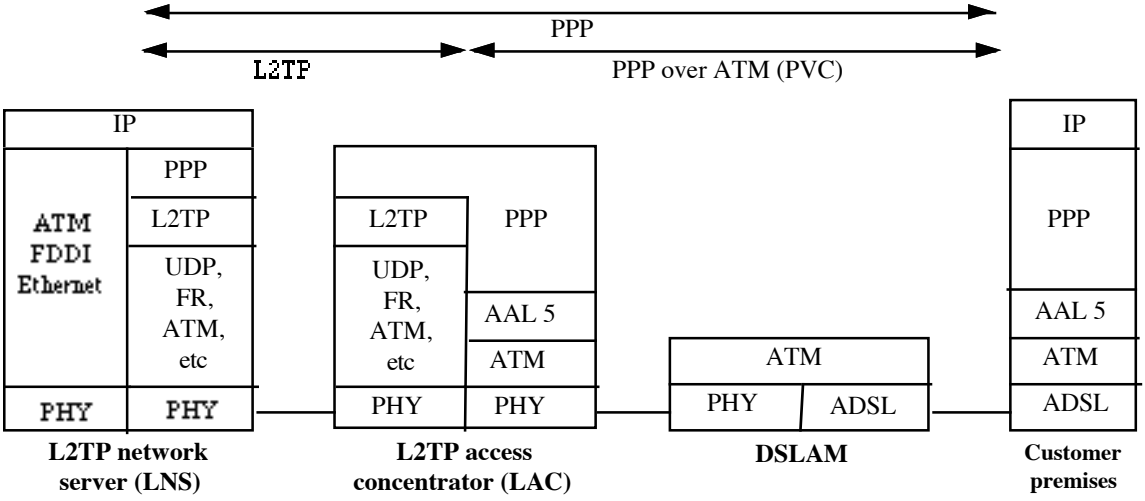


Figure 9.8: The L2TP access aggregation scheme

A tunnel between the LAC and an LNS can multiplex several PPP sessions, each associated with a different ADSL user. Also, there may be several tunnels between the LAC and an LNS. L2TP utilizes two types of messages, namely, *control messages* and *data messages*. Control messages are used to establish, maintain, and clear tunnels and PPP sessions on demand. Data messages are used to carry PPP frames over a tunnel.

The control and data messages are encapsulated with a common L2TP header. Some of the fields in this header are: type bit (T), length bit (L), priority bit (P), sequence bit (S), length, tunnel ID, session ID, sequence number (Ns), and expected sequence number (Nr). The type bit field indicates whether the L2TP packet is a control or a data message. The length bit field indicates whether the length field is present. If it is present, the length field gives the total length of the L2TP packet in bytes. The priority bit is used for data messages. If it is set to 1, then the L2TP packet is to be given preferential treatment within the L2TP queues. The L2TP packet is associated with a tunnel ID and a PPP session ID, given in the tunnel IP field and the session ID field respectively. The sequence bit indicates whether sequence numbers are being used. If they are used, then they are carried in the Ns and Nr fields, which are similar to the N(R) and N(S) fields in the HDLC header (see section 2.4). That is, the Ns field contains the sequence number of the transmitted L2TP packet, and the Nr field contains the next sequence number the transmitting L2TP expects to receive from its peer L2TP.

A reliable channel between two L2TP peers is provided by L2TP for control messages only. The Ns and Nr sequence numbers are used to detect out-of-sequence

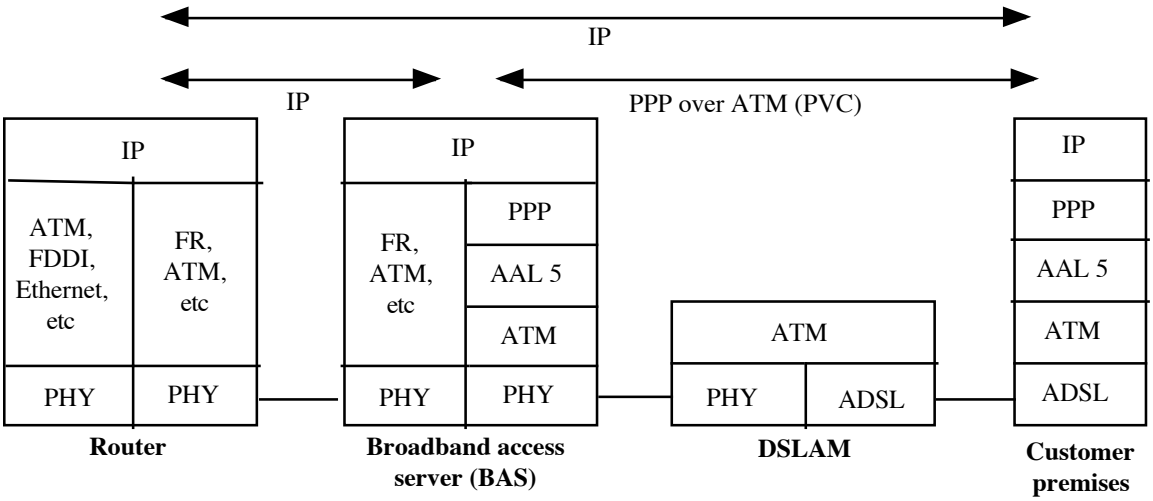


Figure 9.9: The PPP terminated aggregation scheme

packets and missing packets. Lost packets are recovered by retransmission. Data messages may use sequence numbers optionally to reorder packets and detect lost packets. However, no retransmission of data messages takes place. L2TP runs over a network such as IP using UDP, frame relay, and ATM.

The establishment of a session within a tunnel is triggered when the LAC receives a request from an ADSL user to initiate a PPP session to an NLS. Each session within a tunnel corresponds to a single PPP session. Once the session is established, PPP frames can flow between the ADSL user and the LNS. Specifically, PPP frames are transmitted to the LAC over the ATM PVC. The LAC receives the PPP frames from AAL 5, encapsulates each frame with an L2TP header, and transmits it to the LNS as a data message.

9.3.2 The PPP terminated aggregation scheme

This scheme is based on the *remote authentication dial in user service* (RADIUS) protocol. This is a client/server protocol used for authentication, authorization, and accounting. An NAS, referred to as the *broadband access server* (BAS), acts as a client to a RADIUS server which is managed by an NSP. The BAS is responsible for passing authentication information, such as user login and password, to the RADIUS server. This authentication information is submitted by an ADSL user when it initiates a PPP session.

The RADIUS server is responsible for authenticating the ADSL user, and then returning configuration information necessary for the BAS to deliver service to the ADSL user. A BAS also sends the RADIUS server accounting information.

The protocol stacks involved in this scheme are shown in figure 9.9. As in the previous scheme, we assume that an ADSL user at a customer premises is a single computer, rather than a network of computers interconnected via an ATM network. The ADSL user is connected to the DSLAM using ADSL. On the side of the access backbone network, the DSLAM is connected to the BAS via an ATM network. Finally, the BAS is connected to NSP routers over a network, such as IP, frame relay, and ATM.

An ADSL user is connected to the BAS with an ATM PVC via the DSLAM. A PPP session initiated by an ADSL user terminates at the BAS, instead of being tunneled to the NSP as in the previous scheme. The BAS sends the user authentication information to the appropriate RADIUS server, and the PPP session is established after the RADIUS server validates the user. The ADSL user can now transmit IP packets, which are forwarded by the BAS to the router of the appropriate NSP.

Problems

1. You need to download a 20 MB file to your computer at home.
   a. How long does it take to download it assuming that you are connected with a 56K modem, that gives a throughput of 49 Kbps?
   b. How long does it take to download the same file assuming that your computer is connected to an ADSL modem, which provides a throughput of 1 Mbps?

2. Explain why splitterless ADSL modem is preferable over the ADSL modem that requires a splitter.
3. What is the difference between the fast path and the interleaved path in an ADSL modem?
4. What is the advantage of using the network access server (NAS)?
5. In L2TP, why are control messages transmitted over a reliable channel and not data messages?

.

PART FOUR:
SIGNALLING IN ATM NETWORKS

In Part Four, we present the signalling procedures used to establish a point-to-point SVC and a point-to-multipoint SVC. Part Four consists of Chapters 10 and 11.

Chapter 10: Signalling over the UNI

In this Chapter we present the signalling protocols used to establish a point-to-point SVC and a point-to-multipoint SVC over the private UNI.

Chapter 11: The Private Network-Network Interface (PNNI)

This Chapter deals with the procedures used to route a new connection from the originating UNI to the destination UNI.

.

CHAPTER 10
Signalling over the UNI

In this Chapter, we describe the signalling protocols that are used to establish a point-to-point SVC and a point-to-multipoint SVC over the private UNI. ITU-T recommendation Q.2931 is used to establish a point-to-point VC connection, and ITU-T recommendation Q.2971 is used to establish a point-to-multipoint VC connection. Both signalling protocols run on top of a specialized AAL, known as the *signalling AAL* (SAAL). A special sublayer of this AAL is the *service-specific connection oriented protocol* (SSCOP).

We first describe the main features of SAAL and SSCOP, and present the various ATM addressing schemes. Then, we discuss in detail the signalling messages and procedures used by Q.2931, Q.2971, and the *leaf initiated join* (LIJ) capability.

10.1 Connection types

We recall that in ATM networks there are two types of connections, permanent virtual connections (PVC) and switched virtual connections (SVC). PVCs are established using network management procedures, whereas SVCs are established in real-time using signalling procedures. In general, PVCs remain established for long periods of time, whereas SVCs remain active for an arbitrary amount of time. PVCs and SVCs may be point-to-point, point-to-multipoint, and multipoint-to-multipoint.

A point-to-point virtual circuit connection is bi-directional, and it is composed of two unidirectional connections, one in each direction. Both connections are established over the same physical route. Bandwidth requirements and quality of service may be specified separately for each direction. This type of connection is defined by ITU-T as

type 1. Point-to-point connections can be established over the private UNI using the signalling protocol Q.2931.

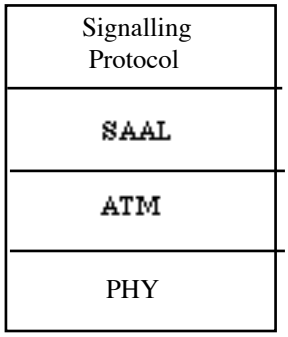


Figure 10.1: The signalling protocol stack

A *type 2* virtual circuit connection is a unidirectional point-to-multipoint connection. It consists of an ATM end-device, known as the *root*, which transmits information to a number of other ATM end-devices, known as the *leaves*. Signalling is provided to establish a type-2 connection and also to add/remove leaves on demand. A type 2 connection is established using Q.2971 in conjunction with Q.2931. Specifically, the connection to the first leaf is established using Q.2931. Q.2971 signalling procedures are then used to add new leaves or drop existing leaves. Q.2971 was designed so that a leaf can only be added or dropped by the root of the connection. In addition to Q.2971, the ATM Forum has introduced the *leaf initiated join* (LIJ) capability, that permits a leaf to add or drop a point-to-multipoint VC connection without intervention from the root.

We note that both Q.2931 and Q.2971 have been modified by the ATM Forum. For simplicity, we refer to these modified protocols by their original ITU-T recommendation names Q.2931 and Q.2971.

A *type 3* connection is a unidirectional multipoint-to-point connection, used primarily in a residential environment. Finally, a *type 4* connection is a multipoint-to-multipoint connection, that can be used for conferencing. An equivalent connection to type 4 can be set-up by allowing each ATM end-device to set-up its own point-to-multipoint connection.

A call whether point-to-point or point-to-multipoint may require multiple connections, each with a different quality of service. This type of connection is useful for multimedia calls which combine video-conferencing, file transfers, and a shared white board. Each part of the application may be carried by a different connection. This could be accomplished by setting up one connection at a time using the appropriate signalling

protocol. Alternatively, these connections can be established as a group in a single request using appropriate signalling. Such signalling is currently under development.

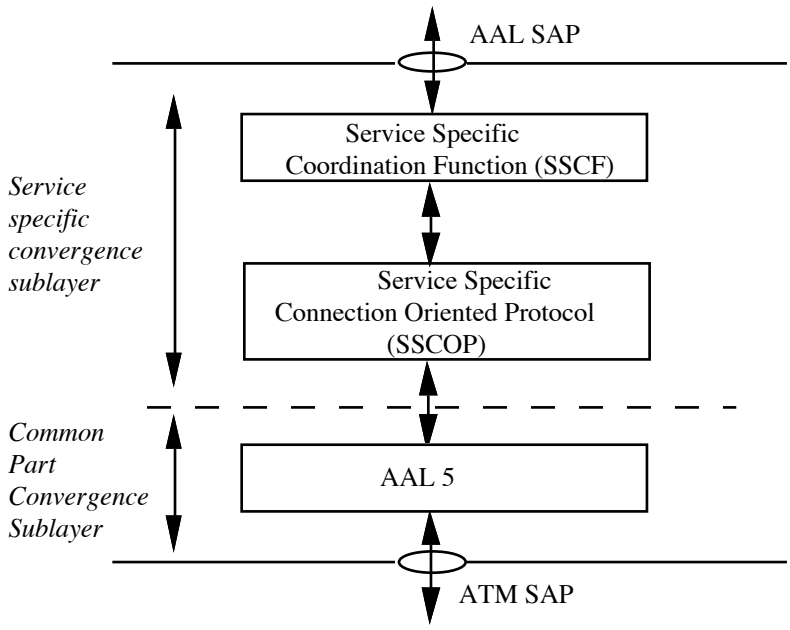


Figure 10.2: The signalling AAL (SAAL)

10.2 The signalling protocol stack

The signalling protocol stack is shown in figure 10.1. It is analogous to the ATM protocol stack shown in figure 4.6 (see section 4.5). The ATM protocol stack shows the protocol layers used for the transfer of data, whereas the stack in figure 10.1 shows the protocol layers used for setting-up an SVC. As we can see, a signalling protocol, such as Q.2931, Q.2971, and PNNI, is an application that runs on top of a specialized ATM adaptation layer known as the *signalling AAL* (SAAL). Below SAAL, we have the familiar ATM layer and the physical layer. The signalling protocol stack is often referred to as the *control plane*, as opposed to the *data plane* that refers to the ATM protocol stack.

10.3 The signalling ATM adaptation layer (SAAL)

As shown in figure 10.2, SAAL consists of an SSCS, which is composed of the *service-specific coordination function* (SSCF) and the *service-specific connection oriented protocol* (SSCOP). The common part of the convergence sublayer is AAL 5, described in Chapter 5. There is no SAR layer. The SSCF maps the services required by the signalling protocol to the services provided by SSCOP. The SSCOP is a protocol designed to provide a reliable connection over the UNI to its peer SSCOP. This connection is used by a signalling protocol to exchange signalling messages over the UNI with its peer protocol.

Name	Description
BEGIN	Used to initially establish an SSCOP connection or to re-establish an existing SSCOP connection
BEGIN ACKNOWLEDGE	Used to acknowledge acceptance of an SSCOP connection request by the peer SSCOP
END	Used to release an SSCOP connection between two peer SSCOP entities
END ACKNOWLEDGE	Used to confirm the release of an SSCOP connection requested by the peer SSCOP
BEGIN REJECT	Used to reject the establishment of a connection requested by the peer SSCOP
RESYNCHRONIZE	Used to resynchronize the buffer and the data transfer state variables in the transmit direction of a connection
RESYNCHRONIZE ACKNOWLEDGE	Used to acknowledge the resynchronization of the local receiver in response to the resynchronize frame
SEQUENCED DATA (SD)	Used to transfer sequentially numbered frames containing user information
STATUS REQUEST (POLL)	Used by transmitting SSCOP to request status information from the receiving SSCOP
SOLICITED STATUS RESPONSE (STAT)	Used to respond to a POLL. It contains the sequence numbers of outstanding SD frames and credit information for the sliding window
UNSOLICITED STATUS RESPONSE (USTAT)	Similar to STAT message, but issued by the transmitter when a missing or erroneous frame is identified.

Table 10.1: The SSCOP frames

10.3.1 The SSCOP

SSCOP provides most of the services provided by LAP-D in ISDN and *message transfer part* (MTP) level 2 in *signalling system no. 7* (SS7). It was designed for ATM networks with a high bandwidth-delay product. That is, the transmission speed of an ATM link and the propagation time are both high. In such an environment, the traditional ARQ schemes described in section 2.3, are not very effective.

SSCOP transfers data to its peer SSCOP in a variable-length PDU, referred to as a *frame*. SSCOP's main function is to establish and release a connection to a peer SSCOP, and to maintain an assured transfer of frames over the connection. This is done using the frames given in table 10.1. The establishment and release of a connection is achieved using the BEGIN and the END frames described in table 10.1.

The assured transfer of frames is achieved using error detection and recovery by retransmission, and flow control which is based on an adjustable sliding window. The data is carried in SEQUENCED DATA (SD) frames. An SD frame uses a sequence number and it can carry a variable length payload of up to 65,535 bytes. The frames STATUS REQUEST (POLL), SOLICITED STATUS RESPOSNE (STAT), and UNCOLICITED STATUS RESPONSE (USTAT) are used to implement a retransmission scheme for erroneously received SD frames or lost SD frames. Specifically, the transmitter periodically sends a POLL frame to request the status of the receiver. This POLL is either triggered when a timer expires or after a certain number of SD frames has been sent. The POLL contains the sequence number of the next SD frame to be sent by the transmitter and a poll sequence number which essentially functions as a time-stamp. The receiver, upon receipt of a POLL, responds with a STAT frame which contains the following information: an SD sequence number up to which the transmitter may transmit (i.e. the window), the number of the next SD frame expected, the echoed poll sequence number, and a list of all SD frames that are currently missing or have been received erroneously. The receiver can determine the missing SD frames by checking for gaps in its buffer and by examining the SD sequence number contained in the POLL. Based on the received STAT message, the transmitter retransmits the outstanding SD frames and advances the transmit window.

If the receiver detects an erroneous or missing SD frame, it sends a USTAT, instead of having to wait for a POLL. A USTAT frame is identical to a STAT frame except that it is not associated with a POLL. The USTAT frame can also be used by the receiver to ask the transmitter to increase or decrease the frequency of POLL frames.

10.3.2 Primitives

SAAL functions are accessed by a signalling protocol, such as Q.2931, Q.2971 and PNNI, through the AAL-SAP, using the following primitives: AAL-ESTABLISH, AAL-RELEASE, AAL-DATA, and AAL-UNIT-DATA.

The AAL-ESTABLISH is issued by a signalling protocol to SAAL in order to request the establishment of a connection over the UNI to its peer protocol. This is necessary, in order for the two peer signalling protocol to exchange signalling messages. This is a reliable connection and it is managed by the SSCOP as described above.

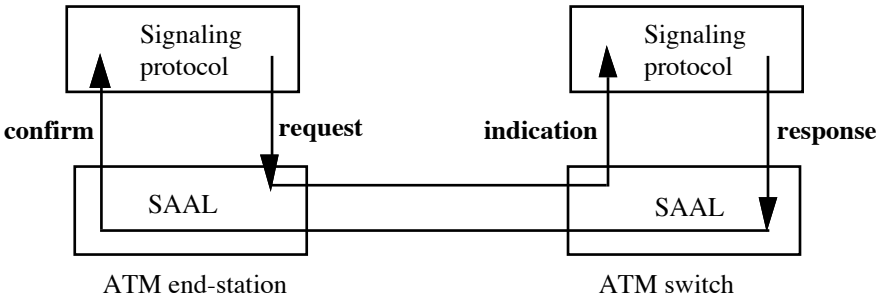


Figure 10.3: The four primitive types

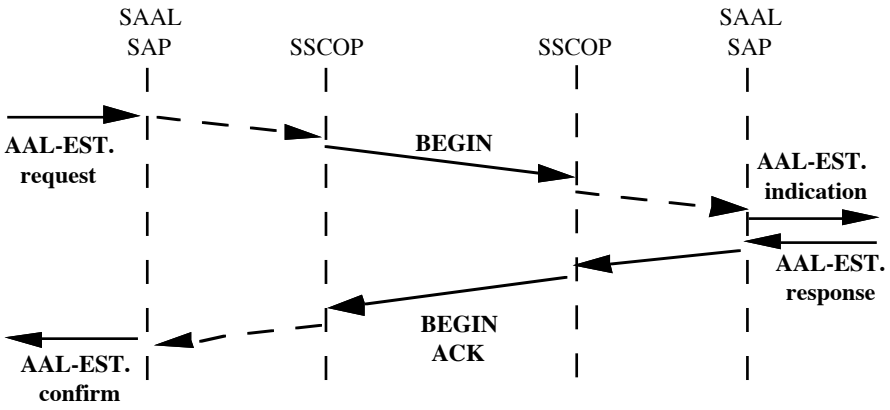


Figure 10.4: Establishment of a connection between two peer signalling protocols

The AAL-RELEASE primitive is a request by a signalling protocol to SAAL to terminate a connection established earlier on using the AAL-ESTABLISH primitive.

The AAL-DATA primitive is used by a signalling protocol to request the transfer of a signalling message to its peer signalling protocol. Signalling messages have a

specific structure, and they will be discussed below in detail. Finally the AAL-UNIT-DATA is used to request a data transfer over an unreliable connection.

These primitives can be one of the following four types: *request*, *indication*, *response*, and *confirm*. These types are shown in figure 10.3. A request type is used when the signalling protocol wants to request a service from SAAL. An indication type is used by SAAL to notify the signalling protocol of a service-related activity. A response type is used by the signalling protocol to acknowledge receipt of a primitive typed indication. A confirm type is used by SAAL to confirm that a requested activity has been completed.

An example of how these primitives and their types are used to establish a new connection over the UNI between two peer signalling protocols is shown in figure 10.4. The primitive AAL-ESTABLISH.request is used to request SAAL to establish a connection. (In order to simplify the presentation, we do not present the signals exchanged between the SSCF and the SSCOP). In response to this request, SSCOP sends a BEGIN frame to its peer SSCOP. The peer SAAL generates an AAL-ESTABLISH.indication to the peer signalling protocol, and its SSCOP returns a BEGIN ACKNOWLEDGE frame, upon receipt of which, the SAAL issues a AAL-ESTABLISH.confirm to the signalling protocol.

An example of how a connection over the UNI between two peer signalling protocols is terminated is shown in figure 10.5. The signalling protocol issues an AAL-RELEASE.request to SAAL, in response of which the SSCOP sends an END frame

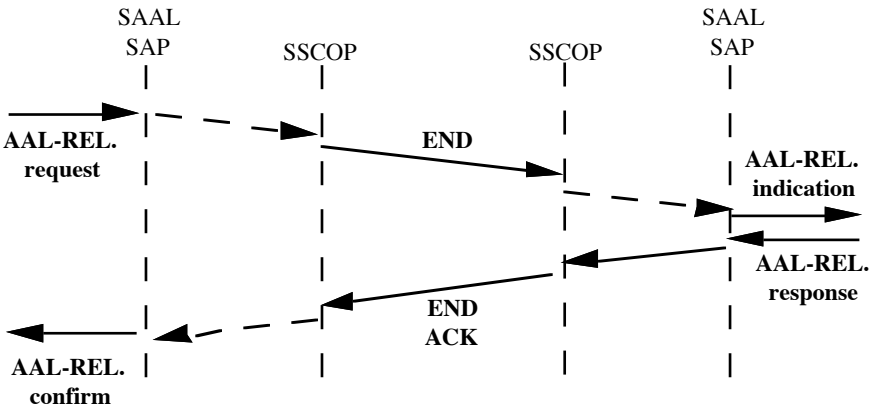


Figure 10.5: Termination of a connection between two peer signalling protocols

to its peer SSCOP. The peer SAAL sends an AAL-RELEASE.indication to the peer signalling protocol, and its SSCOP returns an END ACKNOWLEDGE frame, upon receipt of which the SAAL issues a AAL-RELEASE.confirm to the signalling protocol.

An example of how a signalling protocol transfers messages to its peer protocol is shown in figure 10.6. The signalling protocol transfers a message to SAAL in an AAL-DATA.request, which is then transferred by SSCOP in an SD frame. The SD frame is passed onto AAL 5, which encapsulates it and then breaks it up to 48 byte segments, each of which is transferred by an ATM cell. Figure 10.6 also shows the POLL/STAT frames exchanged between the two peer SSCOPs. The SD frame at the destination side is delivered to the peer signalling protocol using the AAL-DATA. indication primitive.

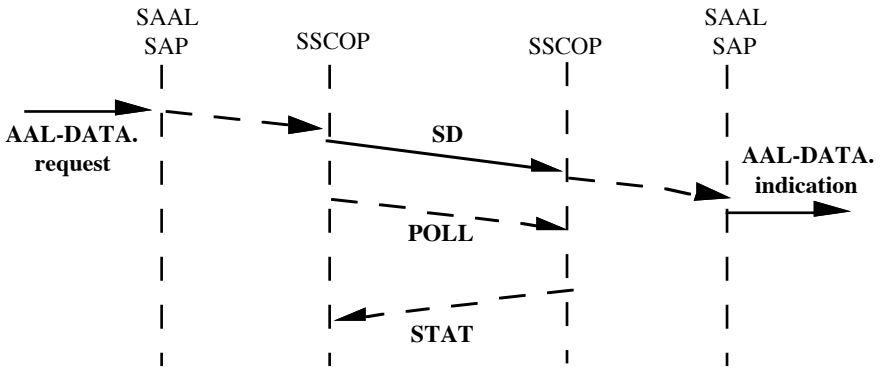


Figure 10.6: Transfer of a signalling message

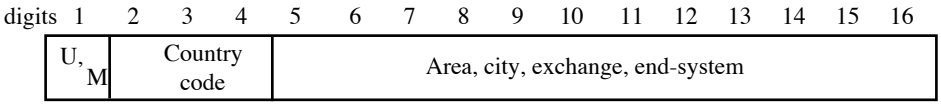


Figure 10.7: The E.164 addressing scheme

10.4 The signalling channel

This is a VC connection that is used exclusively to carry the ATM traffic that results from the exchange of signalling messages between two peer signalling protocols. It is a default connection identified by VPI=0 and VCI=5. This signalling channel is used to control VC

connections within all the virtual paths. It is also possible to set-up a signalling channel with a VCI=0 within a virtual path connection with a VPI other than 0, say with a VPI=x. In this case, this signalling channel can only be used to control VC connections within the virtual path x.

The signalling channel VPI/VCI=0/5 is used in conjunction with the signalling mode known as *non-associated signalling*. In this mode, all the VC connections are created, controlled, and released via the signalling channel VPI/VCI=0/5. A signalling channel within a VPI=x, where x>0, is used in conjunction with the signalling mode known as *associated signalling*. In this mode, only the VC connections within the virtual path x are created, controlled, and released via the signalling channel VPI/VCI=x/5.

10.5 ATM addressing

Each ATM end-device and each ATM switch has a unique ATM address. Private and public networks use different ATM addressing formats. Public ATM networks use E.164 addresses, whereas ATM private network addresses use the OSI *network service access point* (NSAP) format.

The E.164 addressing scheme is based on the global ISDN numbering plan, shown in figure 10.7. It consists of 16 digits, each coded in binary coded decimal (BCD) using 4 bits. Thus, the total length of the E.164 address is 64 bits, or 8 bytes. The first digit indicates whether the address is a unicast or multicast. The next 3 digits indicate the country code, and the remaining digits are used to indicate an area or city code, an exchange code, and an end-device identifier. When connecting a private ATM network to a public network, only the UNIs connected directly to the public network have an E.164 address.

The private ATM addresses are based on the concept of hierarchical addressing domains, and they are 20 bytes long. As shown in figure 10.8, the address format consists of two parts, namely, the *initial domain part* (IDP) and the *domain-specific part* (DSP).

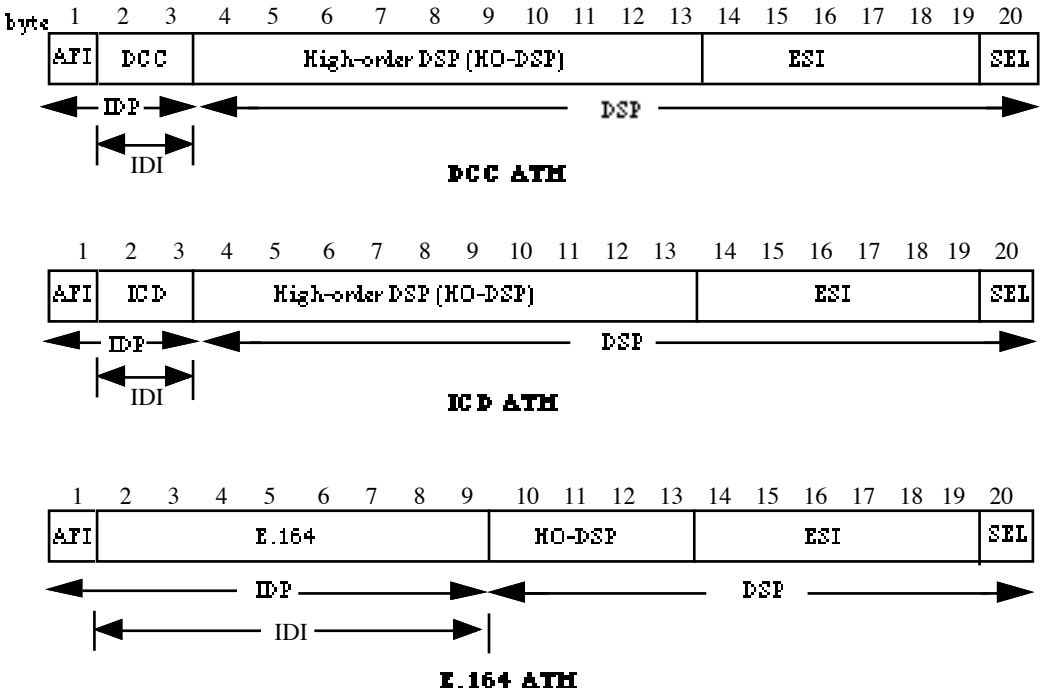


Figure 10.8: The NSAP ATM formats

The IDP specifies an administration authority which has the responsibility for allocating and assigning values for the DSP. It is sub-divided to the *authority and format identifier* (AFI) and the *initial domain identifier* (IDI). AFI specifies the format of the IDI, and the abstract syntax of the DSP field. The length of the AFI field is one byte. The IDI specifies the network addressing domain, from which the DSPs are allocated and the network addressing authority responsible for allocating values of the DSP from that domain. The following three IDIs have been defined by the ATM Forum:

a. DCC (data country code): This field specifies the country in which the address is registered. The country codes are specified in ISO 3166. These addresses are administered by the ISO's national member body in each country. The length of this field is two bytes, and the digits of the data country code are encoded using BCD.

b. ICD (international code designator): The ICD field identifies an authority which administers a coding scheme. This authority is responsible for the allocation of identifiers within this coding scheme to organizations. The registration authority

for the international code designator is maintained by the British Standards Institute. The length of the field is two bytes and the digits of the international code designator are encoded using BCD..

c. E.164 addresses.

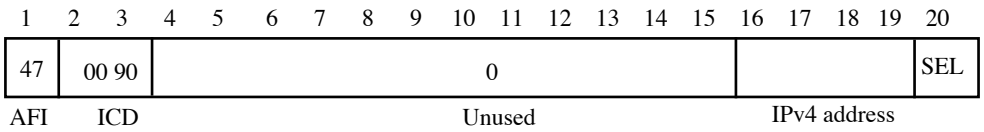


Figure 10.9: The NSAP ATM format for IP addresses

The DSP field consists of the *high-order DSP* (HO-DSP) field, the *end system identifier* (ESI) field and the SEL (selector) field. The coding for the HO-DSP field is specified by the authority or the coding scheme identified by the IDP. The authority determines how identifiers will be assigned and interpreted within that domain. The authority can create further sub-domains. That is, it can define a number of sub-fields of the HO-DSP and use them to identify a lower authority which in turn defines the balance of HO-DSP. The content of these sub-fields describe a hierarchy of addressing authorities and convey a topological structure.

The end system identifier (ESI), is used to identify an end-device. This identifier must be unique for a particular value of the IDP and HO-DSP fields. The end system identifier can also be globally unique by populating it with a 48-bit IEEE MAC address. Finally, the selector (SEL) field has only local significance to the end-device and it is not used in routing. It is used to distinguish different destinations reachable at the end-device.

Of interest is how IP addresses can be mapped to the NSAP structure. Figure 10.9 shows a mapping which can eliminate the use of ATMARP.

In order to show how the NSAP ATM addresses can be used, we describe the ATM addressing scheme of the private ATM network NCANet (The North Carolina Advanced Network). NCANet is a production network in the state of North Carolina, used for research and education purposes. The ICD format, shown in figure 10.8, was selected. The US GOSIP coded HO-DSP field was used, which consists of a 1-byte domain format identifier (DFI) field, a 3-byte administrative authority field (AA), a 2-

byte reserved field, a 2-byte routing domain number (RDN) field, and a 2-byte AREA field. The fields were populated as follows (in hexadecimal):

AFI=47, indicates that an ICD ATM format is used.

ICD=0005, indicates that a GOSIP (NIST) coded HO-DSP field is used.

DFI=80, indicates that the next 3 bytes represent the administrative authority, in this case the Micro Electronics Center of North Carolina (MCNC) which is responsible for handling the regional traffic.

AA=FFEA00, assigned to MCNC by GOSIP (NIST).

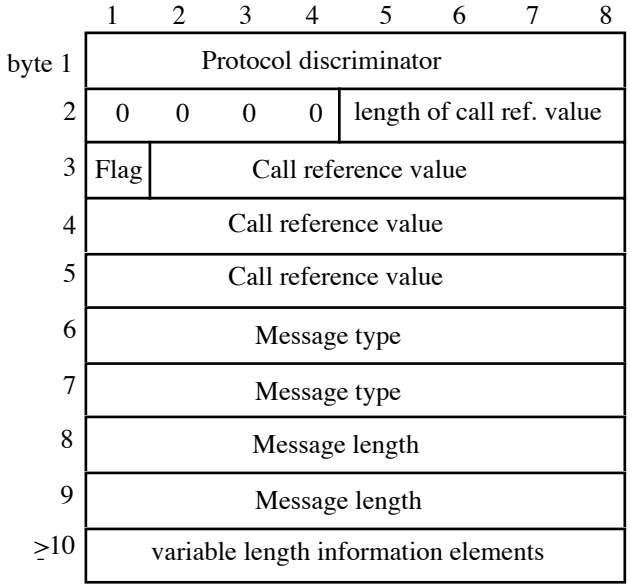


Figure 10.10: The signalling message format

Reserved field=0000.

RDN=xxxx, to be assigned by MCNC. For instance, North Carolina State University is part of NCANet and it has been assigned the RND value of 0101.

AREA=yyyy, to be assigned by the RDN owner. For instance, a group of ATM addresses in North Carolina State University have been assigned the AREA value of 1114.

As a result, all ATM addresses of ATM end-devices and ATM switches in NCANet have the following NSAP prefix (in hexadecimal): 47.0005.80.FFEA00.0000.xxxx.yyyy.

The following address(in hexadecimal) is an example of the complete ATM address of an ATM switch in the ATM Lab of North Carolina State University:

47.0005.80.FFEA00.0000.0101.1114.400000000223.00.

The first 13 bytes is the prefix and it is equal to: 47.0005.80.FFEA00.0000.0101.1114. The next field is the end system identifier (ESI) and it is populated with the value of 400000000223 which is the IEEE MAC address of the switch. The final field is the selector (SEL) and it is populated with the value 00.

10.6 The format of the signalling message

The format of the signalling message is shown in figure 10.10. This message format is used by different signalling protocols, such as, Q.2931, Q.2971, and PNNI. The protocol discriminator field is used to identify the signalling protocol. Bytes 3 to 5 give the call reference number to which the signalling message pertains. This is simply a number assigned to each call, i.e., connection, by the side that originates the call. It is a unique number that has local significance, and it remains fixed for the lifetime of the call. After the call ends, the call reference value is released and it can be used for another call. The call reference value is used by the signalling protocol to associate messages to a specific call, and it has nothing to do with the VPI/VCI values that will be assigned to the resulting ATM connection. The length of the call reference value is indicated in byte 2. For instance, 0011 indicates a 3-byte length. Since the call reference value is selected by the side which originates the call, it is possible that two calls originating at the opposite sides of the interface may have the same call reference value. The call reference flag, in byte 3, is used to address this problem. Specifically, the side that originates the call sets the flag to 0 in its message, whereas the destination sets the flag to 1 when it replies to a message sent by the originating side.

The message type field of the signalling message, bytes 6 and 7, is used to identify the type of the signalling message.

The message length field, bytes 8 and 9, is used to indicate the length of the signalling message, excluding the first 9 bytes. Typically, there is a variety of information that has to be provided with each signalling message. This information is organized into

different groups, known as *information elements* (IE). Each signalling message contains a variable number of information elements, which are appended to the signalling message starting at byte 10. The total length of all the information elements appended to a signalling message is given in the message length field. The structure of an information element is shown in figure 10.11. The first byte contains the IE identifier, which is used to uniquely identify the information element. The second byte contains various fields, such as

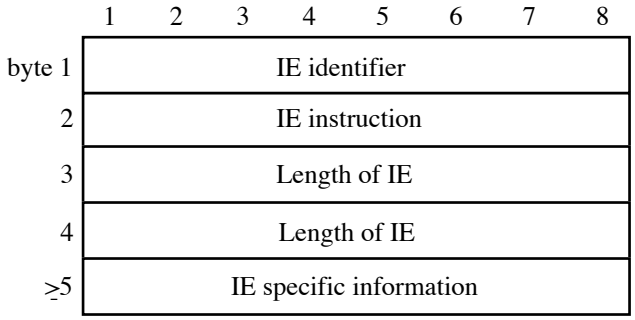


Figure 10.11: The structure of an information element

the coding standard, i.e., ITU-T, ISO/IEC, national, network specific standard (private or public), and the IE action indicator. Bytes 3 and 4 give the length of the information element, excluding the first four bytes, and the remaining bytes starting at byte 5 contain the information specific to the IE.

10.7 The signalling protocol Q.2931

We recall that this protocol is used to establish a point-to-point SVC over the private UNI in real-time. In this section, we first examine the information elements used in the Q.2931 messages, and then we describe the Q.2931 messages and we show how they are used to establish and terminate a call. We will make use of the term *calling user or party* and *called user or party*. The calling user is a user in the end-device that initiates a call, whereas the called user is the user at the end-device that is being called.

10.7.1 Information elements (IE)

Each signalling message contains a variety of information organized into different groups, known as information elements (IE). The following are some of the information elements used in Q.2931 messages:

AAL parameter IE: It is used to indicate the AAL parameter values used between the end-devices.

ATM traffic descriptor IE: It is used to specify the traffic parameters in the forward and backward direction of the connection.

Broadband bearer capability IE: It is used to define the ATM service requested for a new connection.

Broadband high-layer IE, broadband low-layer IE: They are used for compatibility checking by the called user.

Broadband repeat indicator IE: It is used to indicate how repeated IEs are to be interpreted.

Call state: It is used to describe the current status of the call.

Called party number IE, and called party sub-address IE: They are used to identify the called user.

Calling party number IE, and calling party sub-address IE: They are used to identify the calling user.

Cause IE: It is used to describe the reason for generating certain messages and indicates the location of the cause originator.

Connection identifier IE: It is used to identify the VPI/VCI allocated to the connection at the UNI.

End-to-end transit delay IE: It is used to indicate the maximum acceptable transit delay and the cumulative transit delay to be expected for the connection.

Extended QoS parameters IE: It is used to specify the acceptable values and the cumulative values of some of the QoS parameters.

Transit network selection IE: It is used to identify a transit network that the call may cross.

An ATM end-device or an ATM switch may not be able to process every information element included in a signalling message. In this case, the ATM equipment simply uses only the information elements that it needs, and it ignores the rest of them.

Below, we describe some of the information elements. For a detailed description of all the information elements and their fields, the reader is referred to the ATM Forum UNI signalling specification.

AAL IE

The purpose of this IE is to indicate the requested AAL and associated parameters. Some of the fields defined in this IE are: AAL type, bit rate (for AAL 1), source clock frequency recovery method, error correction method, structured data transfer block size, partially filled out cell method, forward and backward CPCS PDU size, MID range, and SSCS type.

ATM traffic descriptor IE

This IE is used to specify the traffic parameters in the forward and backward direction of the connection. We recall that a point-to-point connection is bi-directional. The traffic parameters and quality of service is defined separately for each direction.

In this IE, the PCR, SCR, and MBS for the forward and backward direction can be specified as follows: forward PCR_0, backward PCR_0, forward PCR_{0+1}, backward PCR_{0+1}, forward SCR_0, backward SCR_0, forward SCR_{0+1}, backward SCR_{0+1}, forward MBS_0, backward MBS_0, forward MBS_{0+1}, and backward MBS_{0+1}. The subscript 0 implies that the traffic parameter applies to the traffic consisting of untagged cells, that is, cells with CLP=0. The subscript 0+1 applies to traffic consisting of both tagged and untagged cells, that is, cells with CLP=0 or CLP=1.

Other traffic parameters are also specified in this IE, such as, best effort indicator used for UBR traffic, forward discard enable, backward discard enable, forward violation tagging allowed/disallowed, and backward violation tagging allowed/disallowed.

Additional network-specific code can be used to specify non-standardized traffic parameters, such as average cell rate and average burst size.

Broadband bearer capability IE

This IE is used to indicate the requested bearer service. It contains the fields: bearer class, and ATM transfer capability (ATC).

The following five bearer classes have been defined:

Broadband connection oriented bearer service, class A (BCOB-A),
Broadband connection oriented bearer service, class C (BCOB-C),
Frame relay bearer service,
Broadband connection oriented bearer service, class X (BCOB-X), and
Transparent VP service.

BCOB-A is a connection-oriented service that provides a constant bit-rate and end-to-end timing requirements. BCOB-C is also a connection-oriented variable-bit rate service with no end-to-end timing requirements. BCOB-X is a connection-oriented service where the AAL, traffic type (VBR or CBR) and timing requirements are defined by the user. The frame relay bearer service, as its name implies, is for frame relay services. Finally, in the transparent VP service, both the VCI field and the payload type indicator field are transported transparently by the network. We recall that in section 5.1 of Chapter 5, we introduced the service classes A, B, C, D, and X. BCOB-A is the same as class A, BCOB-C is the same as class C, and BCOB-X is the same as class X.

The ATM transfer capability (ATC) field gives the ATM service class requested by the calling party. We recall from section 7.3 of Chapter 7, that an ATM transfer capability is ITU-T's term for an ATM service category. The following service categories can be defined in the IE:

CBR
CBR with CLR commitment on the traffic with CLP=0+1
Real-time VBR
Real-time VBR with CLR commitment on the traffic with CLP=0+1
Non-real-time VBR
Non-real-time VBR with CLR commitment on the traffic with CLP=0+1
ABR
ATM block transfer-delayed transmission

ATM block transfer-immediate transmission

End-to-end transit delay IE

This IE is used to indicate the maximum acceptable transit delay and the cumulative transit delay to be expected for the connection. These delays are expressed in msec.

The transit delay is defined as the one-way end-to-end delay between the calling user and the called user. It consists of the total processing delay in the end-device of the calling user, the total processing delay in the end-device of the called user, and the max CTD (see section 7.2). The total processing delay in either end-device consists of the AAL handling delay, the ATM cell assembly delay, and other processing delays.

The maximum acceptable transit delay may be indicated by the calling user in its SETUP message to the network. The cumulative transit delay indicated in the SETUP message by the calling user includes only the total processing delay at its end-device. The network indicates the end-device total processing delay plus the network transfer delay to the end-device of the called user. The network transfer delay is calculated cumulatively by adding up the propagation delay on each hop and the switching delay in each ATM switch that lies on the path of the connection. The cumulative transit delay which includes the total processing delays in both end-devices and the network transfer delay is communicated in the CONNECT message.

Extended QoS parameters IE

The information specified in this IE is in addition to the information specified in the end-to-end transit delay IE. It is used to indicate the values of the peak-to-peak cell delay variation, and the CLR (see section 7.2). The acceptable value for each of these two QoS parameters, can be specified in the forward and backward directions. Also, the cumulative values for each of these two QoS parameters can be indicated in the forward and backward direction.

10.7.2 Q.2931 messages

The Q.2931 messages can be grouped into the following three categories: call establishment, call clearing, and miscellaneous. The messages for each category are given in table 10.2. Each Q.2931 message uses the signalling message format described in section 10.6, with the protocol discriminator set to 00001001, and it contains a set of

Call establishment messages	ALERTING CALL PROCEEDING CONNECT CONNECT ACKNOWLEDGEMENT SETUP
Call clearing messages	RELEASE RELEASE COMPLETE
Miscellaneous messages	NOTIFY STATUS STATUS ENQUIRY

Table 10.2: The Q.2931 messages

information elements. Below, we describe the function of each message, and then we show how they are used to establish and clear a call.

ALERTING: This message is sent by the called user to the network and by the network to the calling user to indicate that "called user alerting" has been initiated. Called user alerting is used for calls that require human interface, such as voice. The information element used in this message is the connection identifier.

CALL PROCEEDING: The message is sent by the called user to the network or by the network to the calling user, to indicate that the requested establishment of the call has been initiated and no more call information is accepted. The information element used in this message is the connection identifier.

CONNECT: The message is sent by the called user to the network, or by the network to the calling user to indicate that the called user has accepted the call. The following information element are used in this message: AAL parameter, broadband low-layer, connection identifier, and end-to-end transit delay.

CONNECT ACKNOWLEDGEMENT: This message is sent by the network to the called user to indicate that the user has been awarded the call. It is also sent by the calling user to the network to allow symmetrical call control procedures.

RELEASE: This message is sent by the user to request the network to clear an end-to-end connection. It is also sent by the network to indicate that an end-to-end connection is cleared and that the receiving equipment release the connection identifier and prepare to release the call reference value after sending RELEASE COMPLETE. The cause information element is carried in this message.

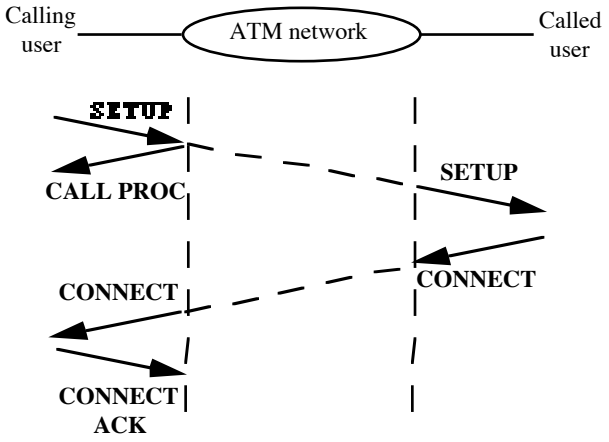


Figure 10.12: Call establishment

RELEASE COMPLETE: This message is sent by the calling user or the network to indicate that the equipment sending the message has released its call reference value and if appropriate, the connection identifier. The cause information element is carried in this message.

SETUP: This message is sent by the calling user to the network and by the network to the called user to initiate the establishment of a new call. The following information elements are used: AAL parameter, ATM traffic descriptor, broadband bearer capability, broadband high-layer, broadband low-layer, called party number, called party sub-address, calling party number, calling party sub-address, connection identifier, end-to-end transit delay, extended QoS parameters, and transit network selection.

The NOTIFY message is sent by the user or the network to indicate information pertaining to a call. The STATUS message is sent by the user or the network in response to a STATUS ENQUIRY message. Finally, the STATUS ENQUIRY message is sent by the user or the network to solicit a STATUS message from the peer Q.2931 protocol.

a. Call establishment

The steps involved in establishing a call are shown in figure 10.12. The calling user initiates the procedure for establishing a new call by sending a SETUP message to its ingress ATM switch across the UNI. The ingress switch sends a CALL PROCEEDING message to the calling user if it determines that it can accommodate the new call. (If it cannot accommodate the new call, it rejects it by responding with a RELEASE COMPLETE message.) The ingress switch calculates a route to the destination end-device over which the signalling messages are transferred. The same route is used to set-up a connection over which the data will flow. It then forwards the SETUP message to the next switch on the route. The switch verifies that it can accommodate the new connection, and the forwards the SETUP message to the next switch, and so on, until it reaches the end-device of the called user. The PNNI protocol, described in Chapter 11, is used to progress the SETUP message across the network.

If the called user can accept the call it responds with CALL PROCEEDING, ALERTING, or CONNECT message. (Otherwise, it sends a RELEASE COMPLETE message.) Upon receiving an indication from the network that the call has been accepted, the ingress switch sends a CONNECT message to the calling user, who responds with a CONNECT ACKNOWLEDGMENT.

b. Call clearing

Call clearing is initiated when the user sends a RELEASE message. When the network receives the RELEASE message, it initiates procedures for clearing the connection to the remote user. Once the connection has been disconnected, the network sends a RELEASE COMPLETE message to the user, and releases both the call reference value and the connection identifier. Upon receipt of RELEASE COMPLETE message the user releases the connection identifier and the call reference value.

10.8 The signalling protocol Q.2971

A point-to-multipoint connection is always unidirectional and it allows an end-device to send its traffic to two or more end-devices. The end-device that generates the traffic is known as the *root* and the receiving end-devices are known as the *leaves*.

In order to establish a point-to-multipoint connection over the private UNI, the root first establishes a point-to-point connection to a single leaf using the Q.2931 signalling procedures. The Q.2971 signalling procedures are used then to add or drop new leaves.

The Q.2971 messages utilize the information elements described in section 10.7.1, and two new information elements, namely, the *endpoint reference* IE, and the *endpoint state* IE. The purpose of the endpoint reference IE is to identify an individual leaf of a point-to-multipoint connection. The endpoint state IE is used to indicate the state of a leaf in a point-to-multipoint connection. The following states have been defined: null, add party initiated, party alerting delivered, add party received, party alerting received, drop party initiated, drop party received, and active.

The following messages are used by Q.2971: ADD PARTY, ADD PARTY ACKNOWLEDGMENT, PARTY ALERTING, ADD PARTY REJECT, DROP PARTY, DROP PARTY ACKNOWLEDGMENT. Below, we discuss the function of each message and give examples of how they are used to add or drop a leaf.

ADD PARTY: This message is used to add a new leaf to an already established point-to-multipoint connection. It is sent from the root to the network to request the addition of a new leaf. The following information elements are used: AAL parameter, broadband high-layer, broadband low-layer, called part number, called party sub-address, calling party number, calling party sub-address, end-to-end transit delay, transit network selection, and the endpoint reference IE.

ADD PARTY ACKNOWLEDGMENT: This message is sent from the network to the root to indicate that the ADD PARTY message was successful. The AAL parameter, broadband low-layer, and end-to-end transit information elements are used in this message.

PARTY ALERTING: This message is sent from the network to the root to notify it that alerting of the called party has been initiated. The endpoint reference IE is used.

ADD PARTY REJECT: This message is sent from the network to the root to indicate that the ADD PARTY request was not successful. The endpoint reference IE and the cause IE are used in this message.

DROP PARTY: This message is sent either from the root to the network or from the network to the root to request to drop a leaf from the connection. The endpoint reference IE is used in this message.

DROP PARTY ACKNOWLEDGMENT: This message is sent from the root to the network or from the network to the root to acknowledge that the connection to a leaf has been successfully cleared. The endpoint reference IE and the cause IE are used in this message.

a. Establishment of a point-to-multipoint connection

The connection to the first leaf is setup using Q.2931, following the procedures described in section 10.7.2. The SETUP message sent by the root is required to contain an endpoint reference IE and the broadband bearer capability IE with the indication that the connection is a point-to-multipoint. The traffic parameters in the traffic descriptor IE are only defined in the forward direction, since the connection is unidirectional. The traffic parameters for the backward direction are set to zero.

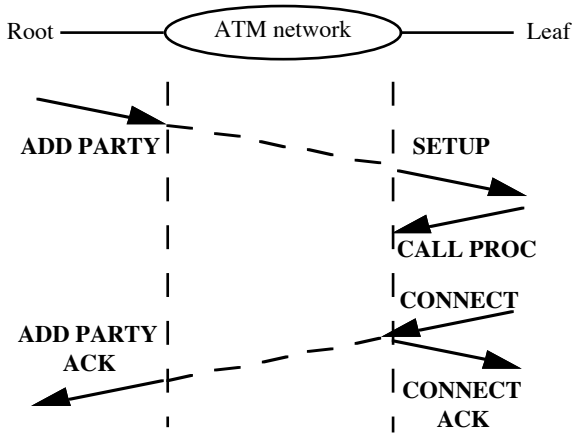


Figure 10.13: Adding a leaf to the point-to-multipoint connection

Adding a new leaf to the connection is effected by sending an ADD PARTY message, as shown in figure 10.13. Each ADD PARTY message has the same call reference value as the one specified in the initial SETUP message used to setup the point-to-multipoint connection to the first leaf. The network issues a SETUP message to the called party, which responds with a CALL PROCEEDING message, and then with a CONNECT message. The calling party finally receives an ADD PARTY ACKNOWLEDGMENT which confirms the addition of the new leaf.

b. Dropping a leaf

A leaf can be dropped from a point-to-multipoint connection due to a request by the root or by the leaf itself. When a leaf wants to drop from the point-to-multipoint connection, it sends a RELEASE message to the network, as shown in figure 10.14. This causes the

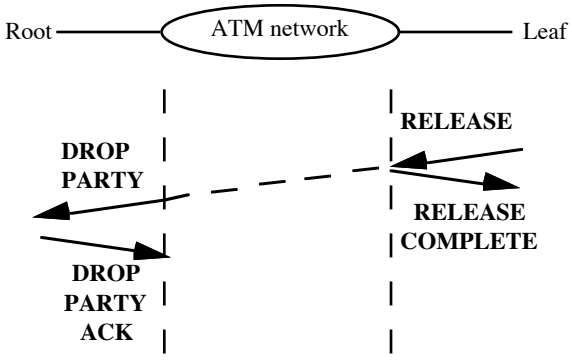


Figure 10.14: Dropping a leaf

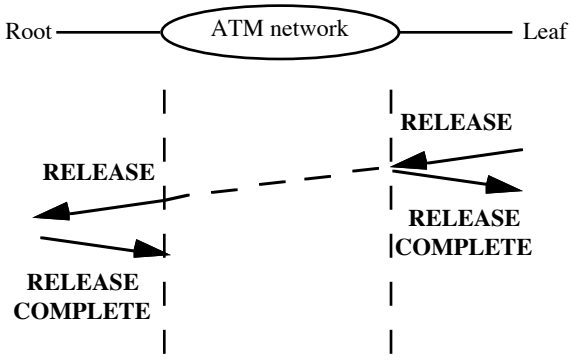


Figure 10.15: Dropping the last leaf

network to send a DROP PARTY message to the root, which responds with a DROP PARTY ACKNOWLEDGMENT message.

When the root wants to drop a leaf, it issues a DROP PARTY message to the network. This causes the network to send a DROP PARTY message to the leaf, to which the leaf responds with a DROP PARTY ACKNOWLEDGE message. The DROP PARTY ACKNOWLEDGE message is delivered by the network to the root.

When the last leaf is dropped from a point-to-multipoint connection, the call is released as shown in figure 10.15.

10.9 Leaf initiated join (LIJ) capability

The Q.2971 procedures described in the previous section allow the root of a connection to add a leaf to its point-to-multipoint connection. It is not possible using Q.2971 for a leaf to join a point-to-multipoint connection without the intervention from the root of the connection. This can be achieved using ATM Forum's *leaf initiated join* (LIJ) capability. Two modes of operation have been defined for the LIJ capability, namely, *leaf-prompted join without root notification*, and *root-prompted join*.

In the leaf-prompted join without root notification mode, a leaf can send a request over its UNI to join a particular point-to-multipoint connection. If the leaf's request is for an existing connection, then the request is handled by the network and the root is not notified when the leaf is added or dropped from the connection. If the leaf's request is for a connection that is not yet established, then the request is forwarded to the root which then performs the initial set-up of the connection. This type of connection, where the network supports the joining and leaving of leaves is referred to as a *network LIJ connection*.

In the root-prompted join mode of operation, a leaf can send a request over its UNI to join a point-to-multipoint connection, but the request is handled by the root. The root adds and drops leaves from a connection following the Q.2971 procedures described above. This type of connection is called the *root LIJ connection*.

The root LIJ connection should not be confused with the connection that can be set-up using Q.2971. For, in Q.2971, it is the root that initiates a join in order for an end-device to be attached to the point-to-multipoint connection as a leaf. It is beyond the

scope of the signalling protocol to provide the root with the ATM addresses of the end-devices that wants to join the connection. A good example of how the root gets this information is in IP multicasting over ATM, described in section 8.3.2. In this case, we saw that MARS keeps a record of all the ATM addresses of the end-devices that want to be part of an IP multicast address. MARS knows about these end-devices, because each end-device is required to register with MARS. The set of all ATM addresses associated with an IP multicast is known as the host map. MARS downloads the host map to the root (in the VC mesh case) or to the multicast server (in the multicast server case), which is responsible for setting up, maintaining, and releasing the point-to-multipoint connection. As we can see, in this case the root is simply provided with the ATM addresses of all the leaves.

In order to support the LIJ capability two new signalling messages were defined, namely, LEAF SETUP REQUEST and LEAF SETUP FAILURE. The LEAF SETUP REQUEST is used by an end-device to request to join a point-to-multipoint connection. If this request fails, the network sends back to the end-device a LEAF SETUP FAILURE message.

Also, the following three new information elements were defined:

Leaf initiated join (LIJ) call identifier IE: It is used to uniquely identify a point-to-multipoint call at a root's interface. The LIJ call identifier is specified in the SETUP message when the root creates a point-to-multipoint call with LIJ capability. It is also specified in the LEAF SETUP REQUEST message when a leaf wishes to join the identified call. How a leaf knows a LIJ call identifier, is outside the scope of this specification. (It could, for instance, be obtained through a directory service.)

Leaf initiated join (LIJ) parameters IE: It is used by the root to associate options with the call when the call is created. Specifically, it can indicate whether the connection is a network LIJ connection.

Leaf sequence number IE: It is used by a joining leaf to associate subsequent signalling messages triggered by a LEAF SETUP REQUEST message that it sent over its UNI.

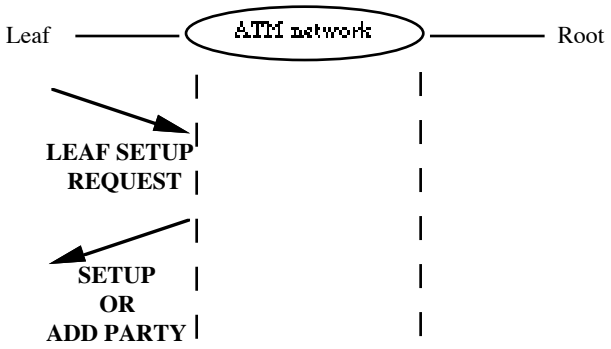


Figure 10.16: Joining an existing network LIJ connection

When a leaf wishes to join a network or root LIJ connection, it sends a LEAF SETUP REQUEST message across its UNI. This message includes the following IEs: calling party number, calling party sub-address, called party number, called party sub-address, LIJ call identifier, leaf sequence number, and transit network selection. The leaf sequence number is used to associate signalling messages sent to the leaf in response to this particular LEAF SETUP REQUEST message. The LIJ call identifier gives the identifier of the connection that the leaf wants to join. The called party number is the address of the root associated with this particular LIJ call, and the calling party number is the leaf's address. As an option, the leaf may include the transit network selection IE. The network uses this information to route the LEAF SETUP REQUEST message towards the root.

If the LEAF SETUP REQUEST is for a network LIJ connection, and the connection already exists, the network sends a SETUP or ADD PARTY message to the leaf, as shown in figure 10.16. If the LEAF SETUP REQUEST is for a root LIJ

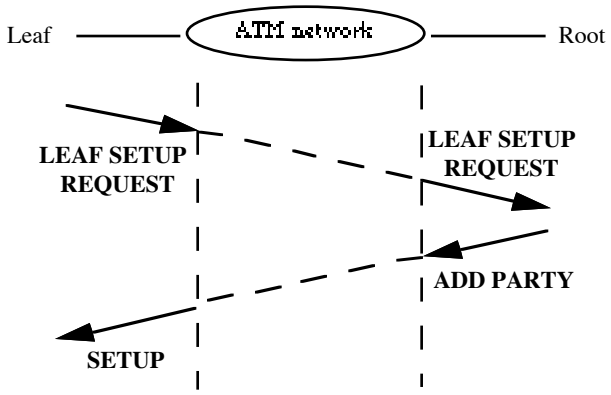


Figure 10.17: Joining an existing root LIJ connection

connection, and the connection already exists, the network forwards the message to the root. To add the leaf to the connection, the root issues an ADD PARTY message following the Q.2971 procedures, as shown in figure 10.17.

If the connection does not exist, but for which a valid root has been specified, the LEAF SETUP REQUEST message will be delivered to the root by the network. Upon receipt, the root creates a point-to-multipoint call using Q.2971. The root has the choice of creating either a network LIJ call or a root LIJ call.

Finally, if the network is unable to complete the LEAF SETUP REQUEST message for any reason, it sends a LEAFSETUP FAILURE message back to the leaf.

10.10 ATM anycast capability

This capability allows a user to request a point-to-point connection to a single end-device that is part of an ATM group address. An ATM group address is a collection of one or more ATM end-devices. An ATM end-device can be a member of zero or more ATM groups at any time. An end-device that is a member of an ATM group, has two types of addresses: an individual address and a group address.

The ATM anycast capability can be requested by a calling user by sending a SETUP message across its UNI. The SETUP message contains the desired ATM group address in the called party number IE. When the connection request reaches a group member, the called member may return its own individual ATM address.

A well-known group address is used to identify an ATM group that implements a well-known service. An example of such a service is the one provided by a LAN emulation configuration server. This server is used to assign an individual LE client to a particular LAN emulation server. ATM Forum well-known addresses are assigned for use in ATM Forum specifications.

Problems

1. Why does the selective-reject ARQ described in section 2.3 does not work well in a network with a high bandwidth-delay product?
2. What are the basic differences between the error recovery scheme in the SSCOP and the more traditional ARQ schemes, such as go-back-n and selective reject?
3. Describe the sequence of primitives issued to set-up a connection between two peer signalling protocols.
4. Identify the sub-fields of the NSAP address of an ATM switch in your organization.
5. What is the purpose of the call reference flag in the signalling message?
6. In which information element, the calling user indicates its traffic parameters?
7. In which information elements the calling user indicates the quality-of-service parameters?
8. Trace the sequence of the signalling messages issued to set-up a connection.
9. Trace the sequence of the signalling messages issued to add a leaf to a point-to-multipoint connection.
10. What is the main difference between Q.2971 and the LIJ capability?

CHAPTER 11

The Private Network-Network Interface (PNNI)

In the previous Chapter, we examined the signalling procedures used to set up a switched virtual connection (SVC) across the private UNI in real-time. In this Chapter, we examine the *private network-network interface* or *private network node interface* (PNNI) protocol used to establish an SVC across a private network. The PNNI protocol consists of the *PNNI routing protocol* and the *PNNI signalling protocol*. The PNNI routing protocol is used to distribute network topology and reachability information between the switches of a private network. The PNNI signalling protocol is used to establish, maintain and clear ATM connections in a private network. We first describe the PNNI routing protocol in detail and then we briefly discuss the PNNI signalling protocol.

11.1 Introduction

Prior to the development of the PNNI protocol, the only standardized mechanism for ATM routing and signalling was the *interim interswitch signalling protocol* (IISP). This protocol uses manually configured static routes and UNI signalling between switches to forward signalling requests across an ATM network. IISP was defined as an interim step until the ATM Forum completed phase I of the PNNI protocol.

As shown in figure 11.1, PNNI provides the interface between two ATM switches that either belong to the same private ATM network or to two different private ATM networks. The abbreviation PNNI can be interpreted as either the *private network node interface* or the *private network-network interface*, reflecting these two possible uses.

The PNNI protocol consists of two components, namely, the *PNNI routing protocol* and the *PNNI signalling protocol*. The PNNI routing protocol is used to distribute network topology and reachability information between switches and clusters of switches. This information is used to compute a path from the ingress switch of the source

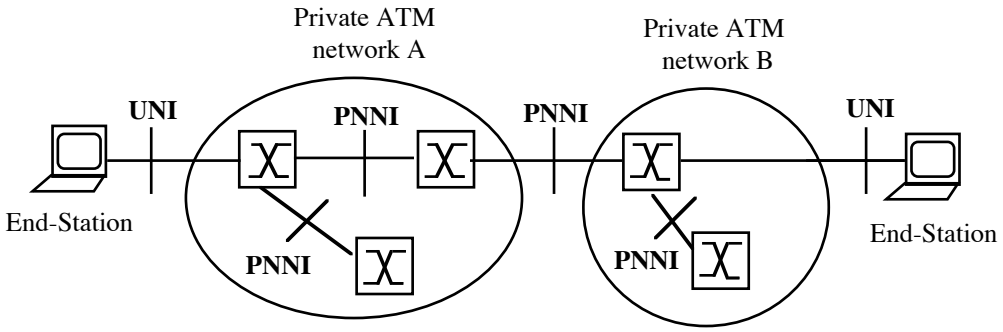


Figure 11.1: The private network-network interface

end-device to the egress switch of the destination end-device over which signalling messages are transferred. The same path is used to set up a connection along which the data will flow. PNNI was designed to scale across all sizes of ATM networks, from a small campus network with a handful of switches to large world-wide ATM networks. Scalability is achieved by constructing a multi-level routing hierarchy based on the 20-byte ATM NSAP addresses. As will be seen, this level of scalability requires a significant complexity in the PNNI protocol.

The PNNI signalling protocol is used to dynamically establish, maintain and clear ATM connections at the private network-network interface and at the private network node interface.

11.2 The PNNI routing protocol

Let us consider the network of private ATM switches shown in figure 11.2. Each circle represents an ATM switch, and a straight line connecting two circles indicates a physical link. The addresses of the nodes used in this example are fictitious, and they are purely

for presentation purposes. In real-life, each node is assigned a 20-byte ATM NSAP address, described in section 10.5. We assume that various end-devices, not shown in figure 11.2, are attached to these switches. Each end-device has also an ATM NSAP address. We recall that only the first 19 bytes of the NSAP address are used to identify an end-device. The last 1-byte selector (SEL) field has only local significance to the end-device and it is used to distinguish different destinations reachable at the end-device.

Now, let us assume that an end-device issues a SETUP message to its ingress switch. The switch calculates a path to the egress switch of the destination end-device, and then it forwards the SETUP message to the next switch along the path. In order for the

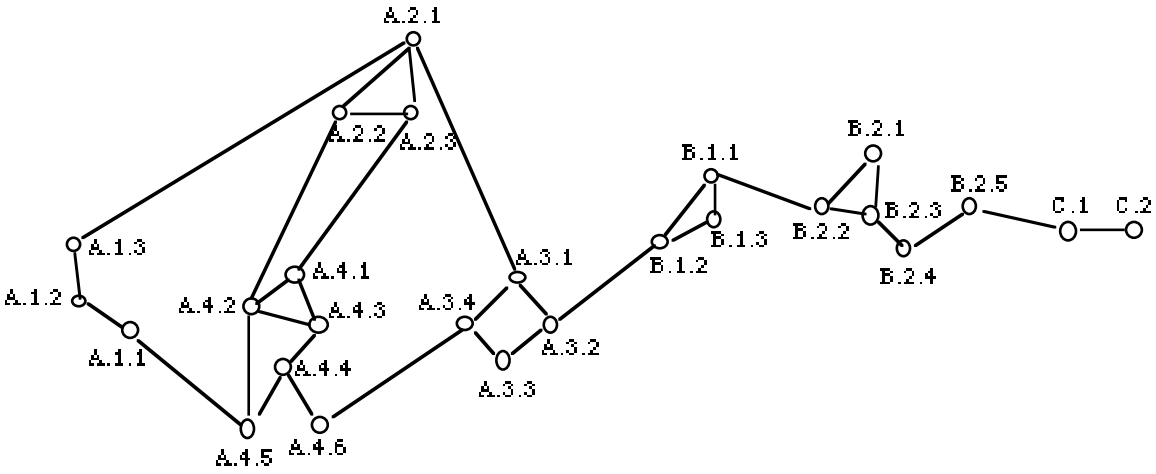


Figure 11.2: A network of ATM switches

switch to calculate the path to the destination egress switch, it has to know the topology of the network and also it has to know the ATM addresses of the end-devices attached to each switch. If the network is small, then it is feasible for each switch to have the complete network topology and reachability information. However, it becomes infeasible when dealing with large networks. PNNI addresses this issue by organizing the network in a hierarchical structure, designed to reduce the amount of topology and reachability information a switch has to keep in order to calculate a path to a destination egress switch. Below, we describe how PNNI works using the example shown in figure 11.2, which was taken from the ATM Forum's PNNI standard.

11.2.1 The lowest-level peer groups

The PNNI hierarchy starts at the lowest level which consists of the actual ATM switches. These switches are referred to as the *lowest-level nodes*. These lowest-level nodes are organized into *peer groups* (PG). The peer groups are then organized into higher-level peer groups, and so on until a hierarchical structure is constructed which encompasses the entire network.

As shown in figure 11.3, the lowest-level nodes are organized into peer groups A.1, A.2, A.3, A.4, B.1, B.2, and C. Peer group A.1, referred to as PG(A.1), consists of the lowest-level nodes A.1.1, A.1.2, and A.1.3, peer group A.2, referred to as PG(A.2), consists of the lowest-level nodes A.2.1, A.2.2, A.2.3, etc.

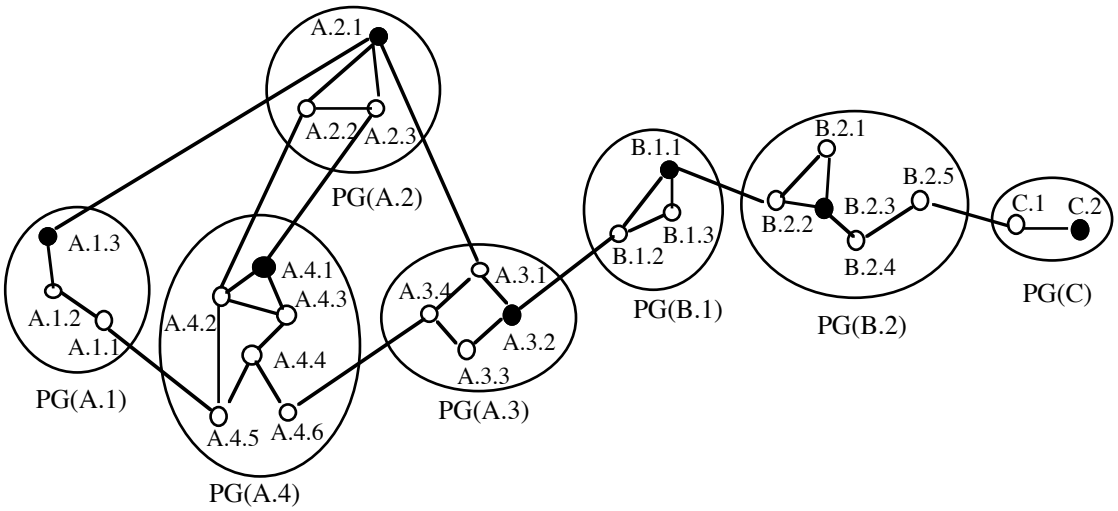


Figure 11.3: Peer groups of the lowest-level nodes

The organization of a set of lowest-level nodes into a peer group is done administratively, by configuring each lowest-level node with the same *peer group identifier*. A peer group identifier is a common prefix of the ATM NSAP addresses of the lowest-level nodes, and it can be at most 13 bytes long (see section 10.5). We recall that the first 13 bytes of an ATM NSAP address correspond to the fields IDP and HO-DSP. The length of this prefix, that is the number of bits that it consists of, is known as the

level indicator. Since the longest prefix can be 13 bytes, the longest level indicator can be 104. Neighbouring nodes exchange peer group identifiers in *hello* packets using the *hello* protocol. If they find that they have the same peer group identifier, then they belong to the same peer group.

The lowest-level nodes in a peer group are connected via physical links. When a physical link becomes operational, the two nodes on either side of the link initiate an exchange of information through hello packets. Each node announces the ATM addresses of the end-devices attached to it, the node's ATM address, and the port identifier of the physical link. Each node then bundles this information into *PNNI topology state elements* (PTSE) and floods them reliably throughout the peer group. Each node also generates a PTSE that describes its own identity and capabilities. Reliable flooding is done as follows. The PTSEs are carried in *PNNI topology state packets* (PTSP). When a PTSP is received, it is acknowledged and then it is sent to all the other neighbours of the node. The PTSEs are subject to aging and they get removed after a predefined duration,

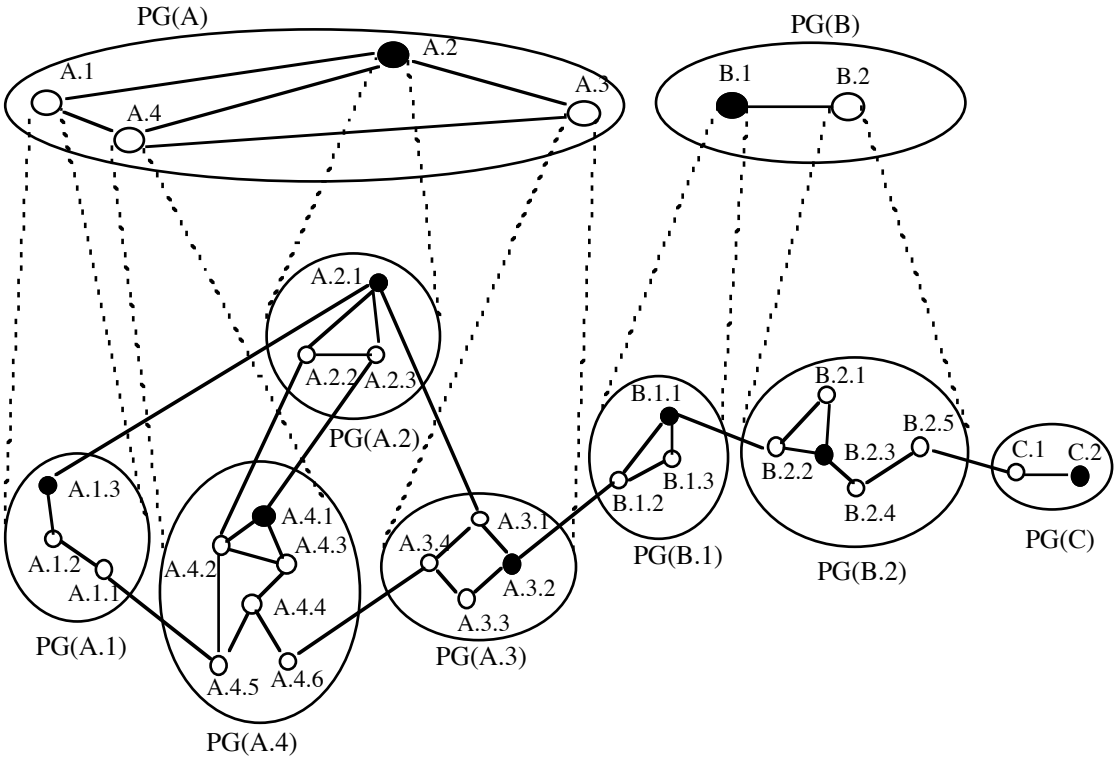


Figure 11.4: Second level of peer groups

unless they are refreshed. So, PTSEs are issued periodically and also when a new link becomes operational. As a result of this mechanism, each node in the peer group has an identical copy of the topology database.

It is possible that a node in a peer group may have one or more links to other nodes, which do not belong to the same peer group. In this case, this node is referred to as a *border* node. For instance, node A.1.1 is connected to A.4.5 via a physical link, and in view of this, it is a border node as far as peer group A.1 is concerned. Likewise, A.4.5 is a border node as far as peer group A.4 is concerned. As we will see, border nodes are important in the creation of the PNNI hierarchy.

11.2.2 The next level of peer groups

The peer groups themselves get organized into higher-level peer groups, as shown in figure 11.4. In this example, the new peer groups are A and B, referred to in the figure as PG(A) and PG(B). Peer group A consists of the lower-level peer groups A.1, A.2, A.3, and A.4, and peer group B consists of the lower-level peer group B.1 and B.2.

Each of the lower-level peer groups is represented in the higher-level peer group as a single node, known as the *logical group node*. For instance, in peer group A, node A.1 is the logical group node that represents peer group A.1, node A.2 is the logical group node that represents peer group A.2, and so on. A logical group node is an abstraction of a peer group for the purpose of representing it in the next level of the PNNI hierarchy. Logical group nodes are connected by *logical links*, which may be either physical links or SVCs.

A logical group node is implemented in one of the lowest-level nodes of the peer group that it represents. It has an ATM address which is basically the address of the lowest-level node on which it has been implemented, but with a different selector (SEL) value. The same lowest-level node is also the *peer group leader* (PGL). The peer group leader for each peer group in figures 11.3 and 11.4, is indicated by a black circle. The peer group leader is elected among the lowest-level nodes of its peer group as follows.

Each lowest-level node in the peer group is configured with a leadership priority which is distributed to the nodes of its peer group. The node with the highest priority becomes the peer group leader.

The only function that a logical group node performs is to distribute aggregated and summarized information to its peer group about the peer group that it represents. Also, a logical group node passes information from its own peer group to the peer group leader of the peer group that it represents. The logical group node does not participate in the PNNI signalling. The peer group leader's function is to distribute information received by the logical group node to all the members of its peer group. Otherwise, the peer group leader acts like the other nodes in its peer group.

Peer group A is called the *parent peer group* of the peer groups A.1, A.2, A.3, and A.4, and each of the peer groups A.1, A.2, A.3, and A.4 is called the *child peer group* of peer group A.

11.2.3 Uplinks

A logical link connecting two logical group nodes may be a physical link or an SVC. PNNI links are classified into *horizontal*, *exterior*, and *outside* links. A horizontal link connects two nodes within the same peer group. An exterior link connects a node within a peer group to a node which does not operate the PNNI protocol. Finally, an outside link connects two nodes within two different peer groups. For example, in figure 11.4, the link that connects the lowest level nodes A.4.5 and A.4.4 is a horizontal link, whereas the link that connects the lowest level nodes A.1.1 and A.4.5 is an outside link. Two nodes

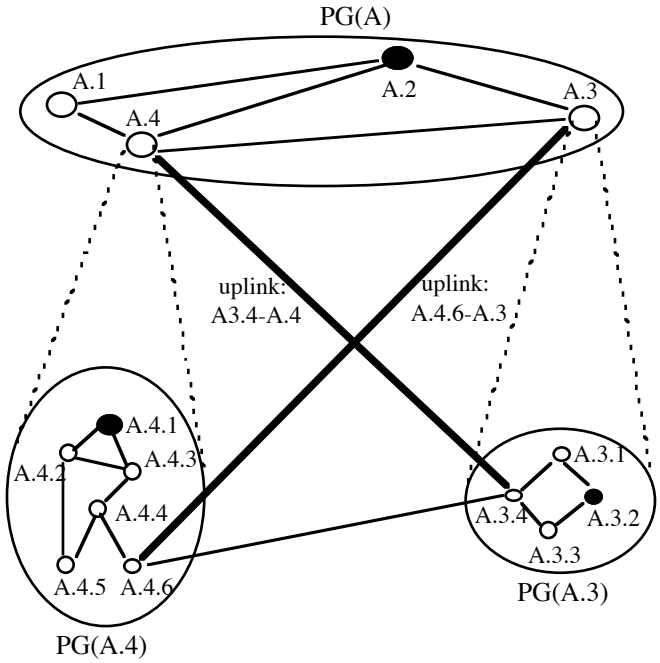


Figure 11.5: Uplinks and upnodes

connected with an outside link are in fact border nodes. This definition of links and border nodes holds at every level of the PNNI hierarchy.

Border nodes extend the hello protocol across outside links to include information about their respective higher-level peer groups and the logical group nodes representing them in these peer groups. This information allows the border nodes to determine that they have a common higher-level peer group and also determine the higher-level logical group nodes to which the border nodes are connected. For instance, border node A.3.4 recognizes that its neighbour A.4.6 is represented by logical group node A.4. Consequently, A.3.4 advertises its link between itself and A.4. This link is known as an *uplink* and node A.4 is known as an *upnode*. Similarly, node A.4.6 advertises its uplink to A.3. These two uplinks are shown in figure 11.5. Border nodes advertise their uplinks in PTSEs flooded in their respective peer groups. This enables all the nodes in the peer group to update their topology database with these new uplinks. This information is also fed up to the logical group nodes by the peer group leaders. As a result, logical group nodes A.4 and A.3 will become aware that they belong to the same peer group, and they will establish a logical link, which will be an SVC.

11.2.4 Information exchange in the PNNI hierarchy

We recall that when a physical link is activated, the two attached lowest-level nodes initiate an exchange of PNNI related information. Similarly, when a logical link becomes operational, the attached logical group nodes initiate an exchange of information. Each node on the link sends hello packets to the node on the other side of the link specifying the addresses of the end-devices that are attached to it, its own ATM address, port identifier for the link, and link status information such as the total allocated bandwidth. Hello packets also support the exchange of peer group identifiers so that neighbouring nodes can determine if they belong to the same peer group or different peer group.

The exchange of PNNI routing information over a physical link is done via the *routing control channel* (RCC) designated with VPI=0 and VCI=18. The exchange of PNNI routing information between logical group nodes, is done by an SVC. The VPI/VCI values for this SVC are assigned in the normal way when it is set-up. The PNNI routing protocol runs on top of AAL 5 is used.

Feeding information up the hierarchy

Within each peer group, the peer group leader has a complete topology state information from all the nodes in its peer group. This information is fed to the logical group node that represents the entire peer group in the parent peer group, which in turn floods it to all the nodes in the parent peer group. The information consists of reachability and topological aggregation.

The reachability information consists of summarized addresses reachable through the lower level peer group. Address summarization is discussed below in section 11.2.7.

Topology aggregation refers to summarized topology information used to route a new connection across a peer group. This aggregation process reduces the amount of information that is needed to be exchanged, and consequently it facilitates scalability in a large network. There are two types of aggregation: *link aggregation* and *nodal aggregation*. In link aggregation, a set of links between two peer groups is aggregated to a single logical link. For instance, in figure 11.4, the two links connecting peer group A.2 and A.4, i.e., links A.2.2-A.4.2 and A.2.3-A.4.1, are represented by a single link. In nodal

aggregation, the topology of an entire peer group is represented in the parent peer group by a *complex* node which indicates the aggregate links between the peer group and other peer groups.

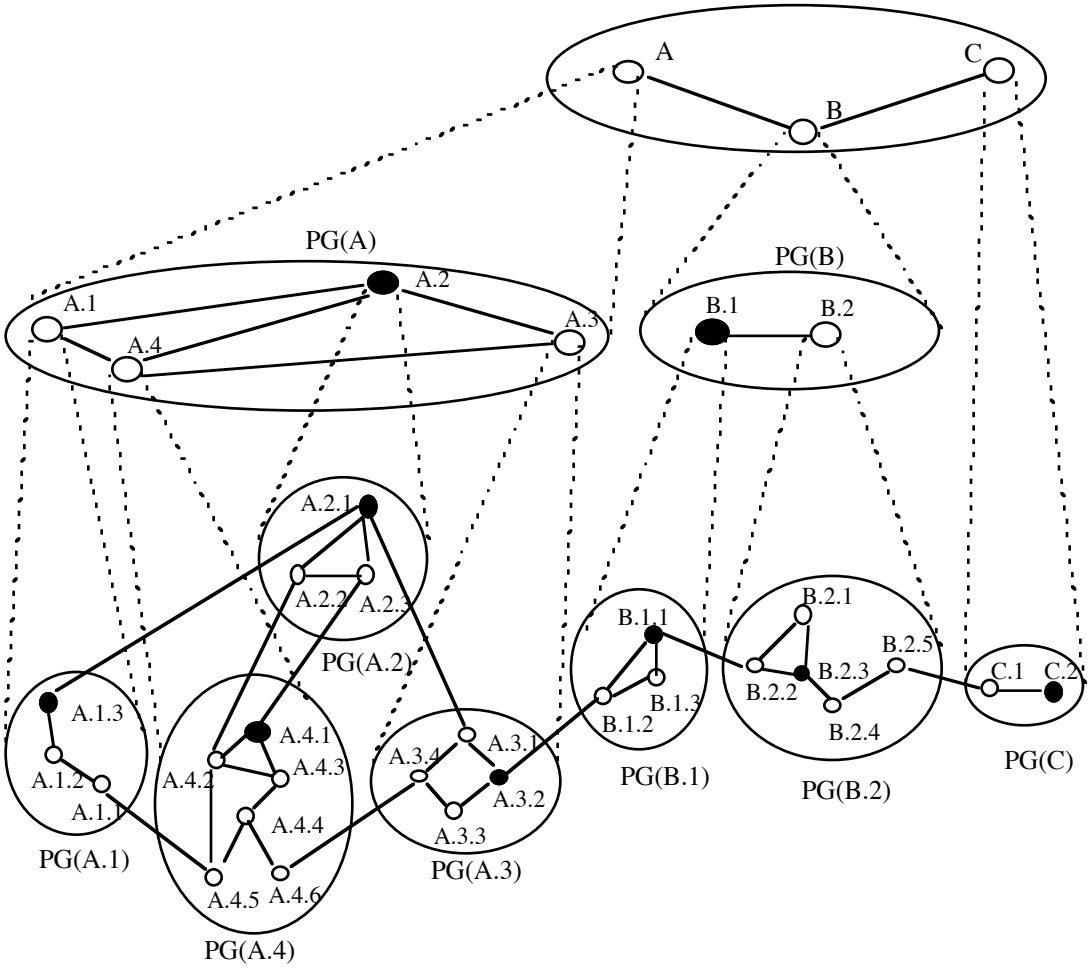


Figure 11.6: The complete PNNI hierarchy

Feeding information down the hierarchy

This is necessary to allow nodes in lower level peer groups to route to all destination reachable via the PNNI hierarchy. Each logical group node feeds down information to its peer group leader, who in turns floods it to all the nodes of the peer group. The information consists of all the PTSEs it originates or receives via flooding from the other

members of its peer group. PTSEs flow horizontally among the nodes of a peer group, and downwards into and through child peer groups.

11.2.5 The highest-level peer group

Let us now go back to our example and complete the construction of the PNNI hierarchy. We note in figure 11.4, that we have not as yet obtained connectivity for all the lowest-level

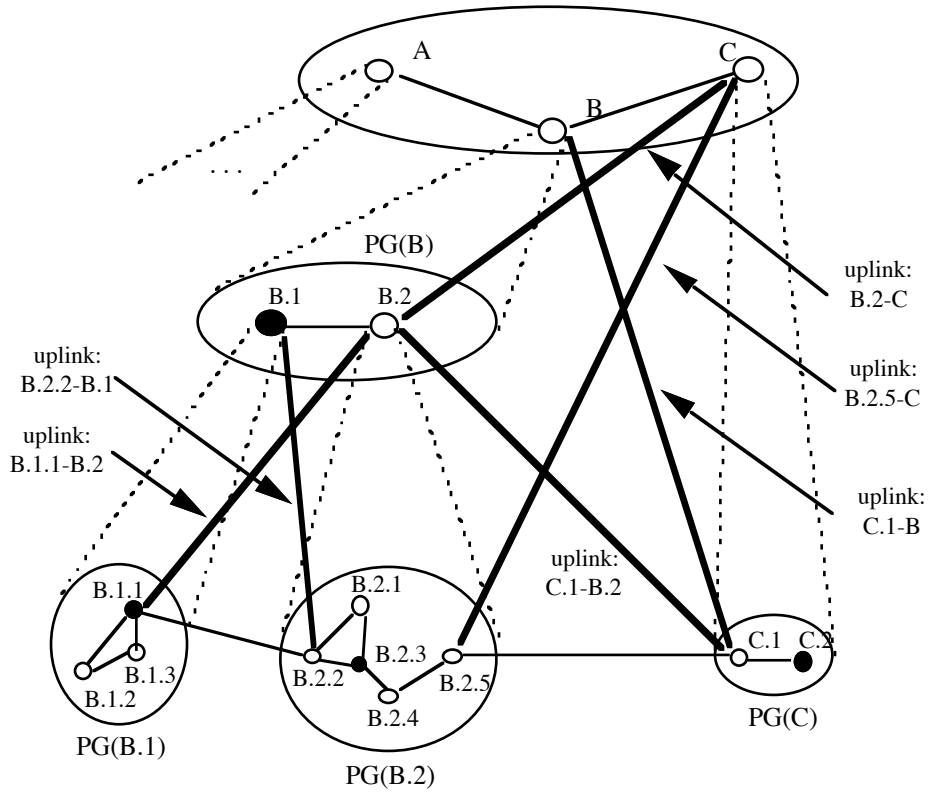


Figure 11.7: Uplinks and upnodes

nodes. This can be achieved by configuring peer groups A, B, and C into a single parent peer group, as shown in figure 11.6. (Another possibility would be to create a parent peer group with nodes B and C and then aggregate that with node A to form the highest level peer group.)

This is achieved following the same procedures as before. The logical group nodes in peer groups A and B elect a peer group leader, who then instantiates a logical

group node in the higher-level peer group. Specifically, A.2 is the peer group leader of peer group A, B.1 is the peer group leader in peer group B, and C.1 is the peer group leader in peer group C. Each of these three peer group leaders instantiates a logical group node which is a member of the higher-level peer group.

PNNI permits the construction of asymmetrical hierarchies. That is, a parent peer group can be also a grandparent or a great-grandparent peer group to some other lower level peer group. For example, the lower-level peer group C in figure 11.6, is directly represented in the highest-level peer group by logical group node C, whereas lower-level peer groups B.1 and B.2 are first grouped into the parent peer group B before they are represented at the highest-level peer group by logical group node B. In view of this, the highest-level peer group is grandparent to peer groups B1 and B2, and a parent to peer group C.

The uplinks are shown in figure 11.7. We see that nodes C.1 and B.2.5 are border nodes connected with the outside link B.2.5-C.1. Consequently, following the discussion on uplinks in section 11.2.3, C.1 advertises an uplink to B, and B.2.5 an uplink to C. Likewise, B.2.2 advertises an uplink to B.1 and B.1.1 advertises an uplink to B.2. The uplink B.2-C, is known as an *induced* uplink, and it is derived as follows. When the peer group leader B.2.3 receives the PTSE flooded by B.2.5 describing the uplink B.2.5-C, it passes the information to the logical group node B.2. This information consists of the common peer group identifier (the highest-level peer group, in this case) and the ATM address of the upnode C. From this information, B.2 recognizes that node C is not a member of peer group B, and it derives the new uplink B.2-C.

Nodes B and C, through the information they receive regarding the uplinks, they recognize that they belong to the same peer group and they establish an SVC between them.

The creation of the PNNI hierarchy can be viewed as the recursive generation of peer groups, beginning with the lowest-level nodes and ending with a single top-level peer group encompassing the entire PNNI routing domain. The hierarchical structure is determined by the way in which peer group identifiers are associated with logical group nodes by configuration.

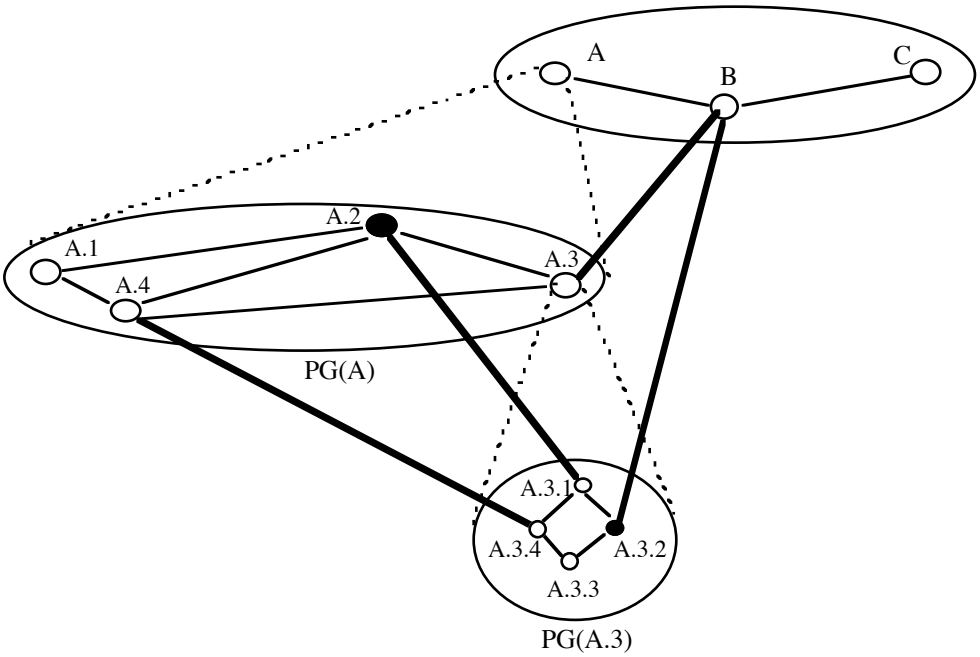


Figure 11.8: A node's view of the PNNI hierarchy

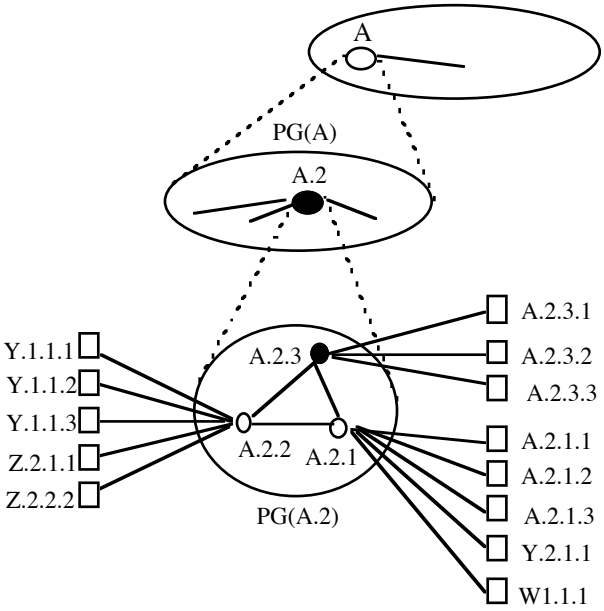


Figure 11.9: End stations attached to nodes A.2.1, A.2.2, and A.2.3

11.2.6 A node's view of the PNNI hierarchy

It is instructive to show how a lowest-level node views the entire PNNI hierarchy shown in figure 11.6. Figure 11.8 shows the view for all the nodes in the lowest-level peer group

A.3. The view is the same for all the nodes in A.3 because flooding within the peer group A.3 ensures that the topology databases of all its members are identical.

Let us consider node A.3.3 and let us assume that the other nodes in peer group A.3 are all active when A.3.3 comes up. A.3.3 will become aware of the links A.3.3-A.3.4 and A.3.3-A.3.2. Through hello packets, A.3.3 will learn about the topology of peer group A.3 and what ATM addresses are reachable through each node. It will also learn about the uplinks A.3.4-A.4, A.3.1-A.2, A.3.2-B, and A.3-B, and it will obtain topology and reachability information of the logical group nodes in peer group A. It will also obtain topology and reachability information about the logical group nodes in the higher-level peer group.

11.2.7 Address summarization

Address summarization reduces the amount of addressing information which is needed to be distributed in a PNNI network. It consists of using a single address prefix to represent a collection of end-devices and node addresses that begin with the given prefix.

Node	Configured summary addresses
A.2.1	P<A.2.1>, P<Y.2>
A.2.2	P<Y.1>, <Z.2>
A.2.3	<A.2.3>

Table 11.1: Configured summary addresses

Address prefixes can be either *summary addresses* or *foreign addresses*. A summary address associated with a node is an address prefix that is either explicitly configured at the node or it takes some default value. A foreign address associated with a node is an address which does not match any of the node's summary addresses. An address that matches one of the node's summary addresses is called a *native address*.

The example given in figure 11.9 is a subset of the PNNI hierarchy shown in figure 11.6. There are five end-devices attached to A.2.1, five end-devices attached to node A.2.2, and three end-devices attached to node A.2.3. As before, we use symbolic

numbers to represent ATM NSAP addresses. The ATM addresses of the end-devices attached to node A.2.1 are: A.2.1.1, A.2.1.2, A.2.1.3, Y.2.1.1, and W.1.1.1. The ATM addresses of the end-devices attached to node A.2.2 are: Y.1.1.1, Y.1.1.2, Y.1.1.3, Z.2.1.1, and Z.2.2.2, Finally, the ATM addresses of the end-devices attached to A.2.3 are: A.2.3.1, A.2.3.2, and A.2.3.3.

We will use the notation P<address> to represent a shorter prefix of an address. That is, P<A.2.1>, P<A.2>, and P<A> are successive shorter prefixes of the address A.2.1.1. An example of configured summary addresses for each node in peer group A.2 is given in table 11.1. Other summary addresses could have been chosen, such as P<Y.1.1> instead of P<Y.1> at node A.2.2, and P<W> at node A.2.1. The summary address P<A.2> could not have been chosen instead of P<A.2.1> or P<A.2.3>, since a remote node selecting a route would not be able to differentiate between the end-devices attached to A.2.3 and the end-devices attached to A.2.1. We note that the address W.1.1.1 is a foreign

Node	Reachable address prefixes
A.2.1	P<A.2.1>, P<Y.2>, P<W.1.1.1>
A.2.2	P<Y.1>, <Z.2>
A.2.3	<A.2.3>

Table 11.2: Advertised reachable address prefixes

address because it does not match the summary addresses of node A.2.1. On the other hand, the address A.2.3.1 is a native address.

The address prefixes advertised by each node in peer group A.2 are shown in table 11.2. Node A.2.1 floods its summary addresses plus its foreign address. Nodes A.2.2 and A.2.3 flood summary addresses only since they do not have any foreign addressed end-devices.

The logical group node A.2 attempts to further summarize the address prefixes flooded in peer group A.2. Specifically, it advertises the following prefixes: P<A.2>, P<Y>, P<Z.2>, and P<W.1.1.1>.

11.2.8 Level indicators

As we have seen, PNNI peer groups occur at various levels. Each peer group is associated with a peer group identifier, which is a prefix of ATM NSAP addresses. The length of prefix, that is the number of bits it consists of, is known as the *level indicator*, and it indicates the level of the peer group within the PNNI hierarchy. The level indicator ranges from 0 to 104 bits. An example of level indicators is given in the PNNI hierarchy associated with problem 1 at the end of this Chapter. (The level indicators are in bold.)

PNNI levels are not dense, in the sense that not all levels are used in a specific topology. For example, a peer group with an identifier of length n bits may have a parent group whose identifier ranges anywhere from 0 to n-1 bits in length. Similarly, a peer group with an identifier of length m bits, may have a child peer group whose identifier ranges anywhere from m+1 bits to 104 bits in length. Similar level indicators are used for nodes and links.

11.2.9 Path selection

Due to the connection-oriented nature of ATM, a source end-device cannot transmit data to a destination end-device unless a connection is first established. We recall from the previous Chapter, that a source end-device requests the establishment of a connection by issuing a SETUP message to its ingress switch. The SETUP message contains a variety of information, such as the ATM address of the destination end-device, the amount of traffic the source wants to submit to the network, and the quality-of-service it expects from the network.

The ingress switch is responsible for the establishment of the connection to the destination end-device. It first calculates a path to the egress switch of the destination end-device. It does that using its local knowledge of the topology of the network that it has acquired through the PNNI routing protocol. Then, it forwards the SETUP message to the next switch on the path who decides whether to accept the new call or not by running its call admission control algorithm (see section 7.6). If it accepts the new call, it propagates the SETUP message to the next switch in the path, and so on until the SETUP

request reaches the egress switch, which forwards it to the destination end-device. If the destination end-device accepts the call, then a CONNECT message is propagated back to the ingress switch following the opposite path. The VPI/VCI values are set-up and bound in the switching table of each switch at that time.

There are two basic routing techniques that are used in networking: *source* routing and *hop-by-hop* routing. In source routing, the ingress switch selects the path to the destination. Other switches on the path simply obey the ingress switch's routing instructions. In hop-by-hop routing, each switch independently selects the next hop for that path, which results in progress towards the destination. Source routing is used in ATM networks, whereas hop-by-hop routing is used in IP networks.

The path that the ingress switch calculates is encoded in a *designated transit list* (DTL), which is included as an information element in the SETUP message. The DTL specifies every node used in transit across the peer group of the ingress switch, and it may optionally specify the logical links to be used among the nodes. The path outside its peer group is not specified in detail. Rather, it is abstracted as a sequence of logical group nodes to be transited. When the SETUP message arrives at a logical group node, the node is responsible for selecting a lower-level source route across the peer group that it represents, so that the SETUP message reaches the next hop destination specified in its DTL.

If a node along the path is unable to accept a set-up request, then the node *cranks it back* to a node upstream the path that is allowed to chose an alternative path. Cranckback is used when the path to the destination cannot be found, or when the requested ATM service is not supported by the switch, or when the switch cannot provide the requested bandwidth or the requested quality of service. It is also used when a DTL processing error occurs.

11.3 The PNNI signalling protocol

PNNI signalling is used to dynamically establish, maintain and clear ATM connections at the private network-network interface and at the private network node interface. The PNNI signalling protocol is based on the ATM Forum's UNI signalling, and some of its

features have been derived from the frame relay NNI signalling defined in ITU-T draft recommendation Q.2934.

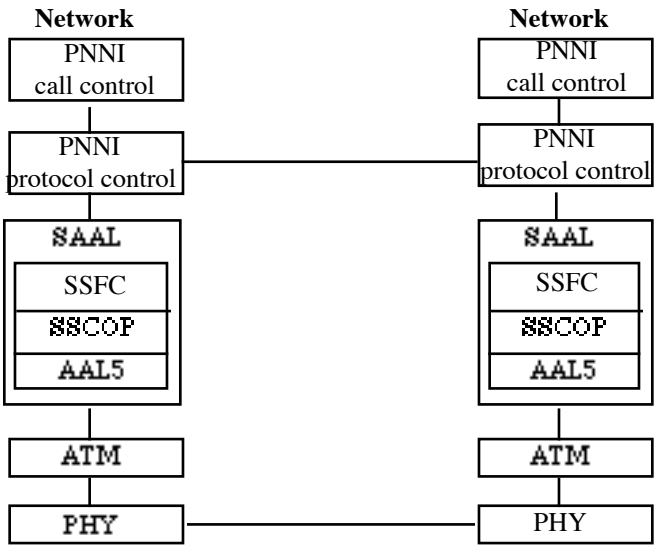


Figure 11.10: The PNNI control plane

The PNNI signalling protocol consists of two distinct entities: *PNNI call control* and *PNNI protocol control*. PNNI call control serves the upper layers for functions such as resource allocation and routing information. The PNNI protocol control entity provides services to the PNNI call control. It processes the incoming and outgoing signalling messages, and it uses SAAL, the signalling AAL, as shown in figure 11.10. For non-associate signalling, the signalling channel with VPI=0, VCI=5 is used between two nodes. For associate signalling, an available value for VCI is selected within the virtual path connection.

The signalling messages for point-to-point connections are the same as in Q.2931, except the CONNECT ACKNOWLEDGEMENT message which is not supported, and they are given in table 11.3. The signalling messages for point-to-multipoint connections are the same as in Q.2971.

Call establishment messages	ALERTING CALL PROCEEDING CONNECT SETUP
Call clearing messages	RELEASE

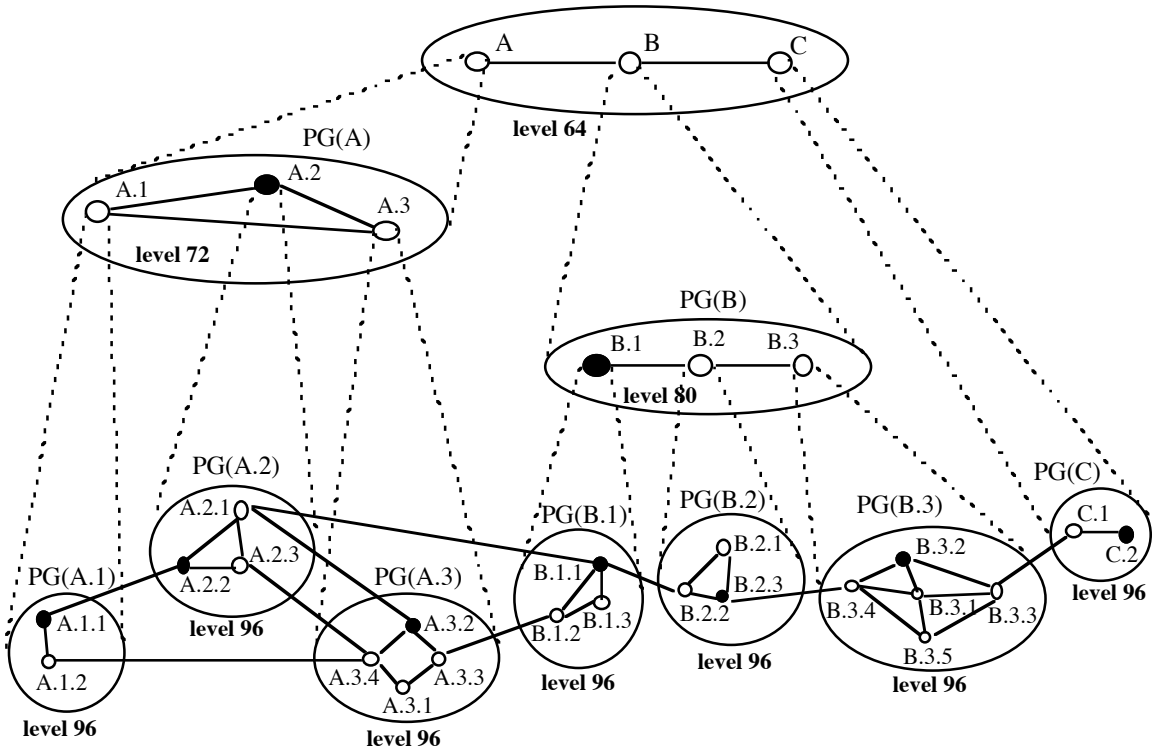
	RELEASE COMPLETE
Miscellaneous messages	NOTIFY STATUS STATUS ENQUIRY

Table 11.3: The PNNI messages for point-to-point call control

The signalling messages use the information elements described in section 10.7.1, with some modifications. In addition, the following new information elements were defined: calling party soft PVPC or PVCC, called party soft PVPC or PVCC, crankback, and designated transit list. The first two information elements are used in relation with soft permanent virtual path (soft PVP) or permanent virtual channel (soft PVC) connections. The crankback information element is used to indicate the level of the PNNI hierarchy at which the connection is being cranked back, the logical node identifier at which the connection was blocked, the logical node identifier preceding the blocking logical node on the connection's path, the logical node identifier succeeding the blocking node on the connection's path, and the reason why the call has been blocked. The designated transit list information element contains the logical nodes and logical links that a connection is to traverse through a peer group at some level of hierarchy.

Problems

1. Consider the PNNI hierarchy shown below.
   a. Identify all the border nodes
   b. Identify all the logical group nodes (The peer group leaders are marked in black).
   c. Identify all the uplinks and the induced uplinks
   d. What is the view of the PNNI hierarchy from node A.1.2?
   e. Node A.2.2 receives a request from an end-device that is attached to it to set-up a connection to an end-device which is attached to node C.1. Describe the path that it will supply in its designated transit list DTL.


A
B
C
level 64
PG(A)
A.1
A.2
A.3
level 72
PG(B)
B.1
B.2
B.3
level 80
PG(A.2)
A.2.1
A.2.3
A.2.2
level 96
PG(A.1)
A.1.1
A.1.2
level 96
PG(A.3)
A.3.2
A.3.4
A.3.3
A.3.1
level 96
PG(B.1)
B.1.1
B.1.2
B.1.3
level 96
PG(B.2)
B.2.1
B.2.3
B.2.2
level 96
PG(B.3)
B.3.2
B.3.4
B.3.1
B.3.3
B.3.5
level 96
PG(C)
C.1
C.2
level 96

List of Standards

In this section we give a list of some of the standards which are relevant to the topics presented in this book. The standards are listed per Chapter.

Chapter 4: Main Features of ATM Networks

B-ISDN General Network Aspects, ITU-T Recommendation I.311, March 1993.

B-ISDN ATM Layer Specification, ITU-T Recommendation I.361, February 1999.

Chapter 5: The ATM Adaptation Layer

Broadband ISDN – ATM Adaptation Layer for Constant Bit Rate Services Functionality and Specification, ANSI, T1/S1 92-605, November 1992.

B-ISDN ATM Adaptation Layer Type 2 Specification, ITU-T Recommendation I.362.2, November 1996.

B-ISDN ATM Adaptation Layer (AAL) Specification, ITU-T Recommendation I.363, March 1993.

Chapter 7: Congestion Control in ATM Network

Traffic Management Specification Version 4.1, ATM Forum, March 1999.

Addendum to Traffic management V4.1 for an Optional Minimum Desired Cell Rate Indication for UBR, ATM Forum, July 2000.

Chapter 8: Transporting IP Traffic Over ATM

LAN Emulation Over ATM Version 1.0, ATM Forum, January 1995.

Multiprotocol Encapsulation Over ATM Adaptation Layer 5, IETF, RFC 2684 (replaces RFC 1483), September 1999.

Classical IP and ARP Over ATM, IETF RFC 2225, April 1998.

Support for Multicast Over UNI 3.0/3.1 based ATM Networks, IETF, RFC 2022, November 1996.

Multicast Server Architectures for MARS-Based ATM multicasting, IETF, RFC 2149, May 1997.

NBMA Next Hop Resolution Protocol (NHRP), IETF, RFC 2332, April 1998.

Cisco Systems' Tag Switching Architecture Overview, IETF, TFC 2105, February 1997.

Multiprotocol Label Switching Architecture, IETF, Internet Draft, August 1999.

LDP Specification, IETF, Internet Draft, August 2000.

Chapter 9: ADSL-Based Access Networks

Data-Over Cable Service Interface Specifications – Radio Frequency Interface Specification, Cable Television Laboratories, 1999

Broadband Optical Access Systems Based on Passive Optical Networks (PON), ITU-T Recommendation G.983.1, October 1998.

Network and Customer Installation Interfaces - Asymmetrical Digital Subscriber Line (ADSL) Metallic Equipment, ANSI, T1.413 Issue 2, June 1998.

Broadband Service Architecture for Access to Legacy Data Networks Over ADSL Issue 1, ADSL Forum TR-012, June 1998.

ATM Over ADSL Recommendation, ADSL Forum, March 1999.

References and Requirements for CPE Architectures for Data Access Version 3, ADSL Forum WT-31, March 1999.

PPP Over AAL5, IETF RFC 2364, July 1998.

Layer Two Tunneling Protocol "L2TP", IETF, Internet-Draft, November 2000.

Remote Authentication Dial in User Service (RADIUS), IETF, RFC 2865, June 2000.

A Method for Transmitting PPP Over Ethernet (PPPoE), IETF, RFC 2516, February 1999.

Chapter 10: Signalling over the UNI

ATM User-Network Interface (UNI) Signalling Specification, Version 4.0, ATM Forum, July 1996

Broadband Integrated Services Digital Network (B-ISDN) – Digital Subscriber Signalling System No 2 (DSS 2) – User-Network Interface (UNI) Layer 3

Specification for Basic Call/Connection Control, ITU-T Recommendation Q.2931, 1995.

Broadband Integrated Services Digital Network (B-ISDN) – Digital Subscriber Signalling System No 2 (DSS 2) – User-Network Interface (UNI) Layer 3 specification for Point-to-Multipoint Call/Connection Control, ITU-T Recommendation Q.2971, October 1995

Chapter 11: The private network-network interface (PNNI)

Private Network-Network Interface Specification Version 1.0 (PNNI 1.0), ATM Forum, March 1996

Glossary of Abbreviations

AAL	ATM adaptation layer
ABR	available bit rate
ABT	ATM block transfer
ACR	allowable cell rate
ADSL	asymmetric digital subscriber line
AFI	authority and format identifier
ANP	AAL 2 negotiation procedure
APON	ATM passive optical networks
ARP	address resolution protocol
ARQ	automatic repeat request
ATM	asynchronous transfer mode
ATU-C	ADSL transceiver unit at the central office
ATU-R	ADSL transceiver unit at the remote terminal
BAS	broadband access server
BCOB-A	broadband connection oriented bearer class A
BCOB-C	broadband connection oriented bearer class C
BCOB-X	broadband connection oriented bearer class X
B-frame	bi-directional-coded frame
B-ICI	broadband inter-carrier interface
BECN	backward explicit congestion notification
BGP	border gateway protocol
BOM	beginning of message
BT	burst tolerance
BUS	broadcast and unknown server
CAC	call admission control
CBR	constant bit rate
CCITT	International Telegraph and Telephone Consultative Committee
CCR	current cell rate

CDVT	cell delay variation tolerance
CER	cell error rate
CI	connection identifier
CIDR	classless inter-domain routing
CIR	committed information rate
CLEC	competitive local exchange carrier
CLLM	consolidated link layer management
CLNAP	connectionless network access protocol
CLNIP	connectionless network interface protocol
CLP	cell loss priority bit
CLR	cell loss rate
CLS	connectionless server
CMR	cell misintertion rate
CO	central office
COM	continuation of message
CoS	class of service
CPS	common part sublayer
CRC	cyclic redundant check
CR-LDP	constraint routing-label distribution protocol
CS	convergence sublayer
CTD	cell transfer delay
DBR	deterministic bit rate
DCC	data country code
DCE	data communication equipment
DMCR	desirable minimum cell rate
DMT	discrete multi-tone
DOCSIS	data-over-cable service interim specification
DSL	digital subscriber loop
DSLAM	ADSL access multiplexer
DSP	domain-specific part
DTE	data terminal equipment

DTL	designated transit list
EFCN	explicit forward congestion notification
EOM	end of message
ER	explicit rate
ESI	end system identifier
FCS	frame check sequence
FDM	frequency division multiplexing
FEC	forwarding equivalent class
FECN	forward explicit congestion notification
FIB	forwarding information base
FRAD	frame relay access devices
FRP/DT	fast reservation protocol with delayed transmission
FTTB	fiber to the basement
FTTC	fiber to the curb
FTTCab	fiber to the cabinet
FTTH	fiber to the home
GCRA	generic cell rate algorithm
GFR	guaranteed frame rate
GSMP	general switch management protocol
HDLC	high-level data link control
HDSL	high data rate DSL
HEC	header error control
HFC	hybrid fiber coaxial
HO-DSP	high-order DSP
IBP	interrupted Bernoulli process
ICD	international code designator
ICMP	internet control message protocol
IDI	initial domain identifier
IDP	initial domain part
IDSL	ISDN DSL
IE	information elements

IFP	Interrupted fluid process
IFMP	Ipsilon's flow management protocol
I-frame	intra-coded frame
IGMP	internet group management protocol
IISP	interim interswitch signalling protocol
InATMARP	inverse ATMARP
ILEC	incumbent local exchange carrier
IP	internet protocol
IPP	interrupted Poisson process
ISO	International Organization of Standards
ISP	Internet service provider
ITU	International Telecommunication Union
IWU	interworking unit
L2TP	layer 2 tunnel protocol
LAC	L2TP access concentrator
LDP	label distribution protocol
LE	LAN emulation
LE-ARP	LAN emulation address resolution
LECID	LE client identifier
LER	label edge router
LIS	logical IP subnet
LIJ	leaf initiated join
LMDS	local multipoint distribution services
LMI	local management interface
LSP	label switched path
LSR	label switching router
LUNI	LAN emulation user to network interface
MARS	multicast address resolution server
MBS	maximum burst size
MCR	minimum cell rate
MCS	multicast servers

ME	mapping entity
MFS	maximum frame size
MMBP	Markov modulated Bernoulli process
MMPP	Markov modulated Poisson process
MPLS	multi-protocol label switching
MPOA	multi-protocol over ATM
MTU	maximum transfer unit
NAS	network access server
NBMA	non broadcast multiaccess network
NHC	next hop client
NHRP	next hop resolution protocol
NHS	next hop server
NNI	network node interface
NRT-VBR	non-real-time variable bit rate
NRT-SBR	non-real-time statistical bit rate
NSAP	network service access point
NSP	network service provider
NTR	network timing reference
OC	optical carrier
OLT	optical line terminator
ONU	optical network unit
OSI	open system interconnection reference model
OSPF	open shortest path first
PCM	pulse code modulation
PCR	peak cell rate
PDH	plesiochronous digital hierarchy
PDU	protocol data unit
P-frame	predictive-coded frame
PGL	peer group leader
PIM	protocol independent multicast
PMD	physical medium dependent sublayer

PNNI	private network-network interface or private network node interface
PON	passive optical network
PPP	point-to-point protocol
PTI	payload type Indicator
PTSE	PNNI topology state element
PTSP	PNNI topology state packet
PVC	permanent virtual connection
QAM	quadrature amplitude modulation
RADIUS	remote authentication dial in user service
RCC	routing control channel
RM	resource management
ROC	regional operations center
RSVP	resource reservation protocol
RT-VBR	real-time variable bit rate
RT-SBR	real-time statistical bit rate
SAAL	signalling AAL
SAR	segmentation-and-reassembly sublayer
SBR	statistical bit rate
SCR	sustained cell rate
SDH	synchronous digital hierarchy
SDU	service data unit
SDSL	symmetric DSL
SEL	selector
SMDS	switched multimegabit data service
SONET	synchronous optical network
SSCF	service-specific connection function
SSCOP	service-specific connection oriented protocol
SSCS	service specific convergence sublayer
SSM	single segment message
STF	start field
STM	synchronous transfer mode

STS-1	synchronous transport signal level 1
SVC	switched virtual connection
TC	transmission convergence sublayer
TDP	tag distribution protocol
TER	tag edge router
TFIB	tag forwarding information base
TSR	tag switching router
TTL	time to live
UBR	unspecified bit rate
UNI	user network interface
VCC	virtual channel connection
VCI	virtual channel identifier
VDSL	very high data rate DSL
VPI	virtual path identifier
WDM	wavelength division multiplexing
xDSL	x-type digital subscriber line

Index

25 Mbps, 152, 178

4B/5B, 73, 77

8B/10B, 73, 78

AAL 1, xii, 81, 83, 84, 85, 86, 87, 88, 97, 242
AAL 2, xii, 81, 83, 88, 89, 91, 92, 97, 171, 277
AAL 2 negotiation procedure, 91, 92, 277
AAL 3/4, xii, 81, 83, 92, 95, 96
AAL 5, xii, 81, 83, 92, 95, 96, 146, 164, 184, 194, 195, 196, 205, 206, 207, 222, 224, 231, 235, 262
AAL parameter IE, 241
acknowledged connectionless service, 28
adaptive clock method, 88
ADD PARTY, 248, 249, 252, 253
ADD PARTY ACKNOWLEDGMENT, 248, 249
ADD PARTY REJECT, 248
address resolution protocol, xi, xiv, 37, 178, 183, 184, 188, 191, 208, 273, 277
ADSL access multiplexer, 217, 222, 225, 278
ADSL super frame, viii, xiv, 220
ADSL transceiver unit at the central office, 215, 217, 219, 220, 277
ADSL transceiver unit at the remote terminal, 215, 216, 217, 219, 220, 222, 277
ALERTING, 245, 247, 248, 270
allowable cell rate, 134, 167, 168, 169, 277
American National Standards Institute, 5, 8, 46, 51, 215, 273, 274
associated signalling, 236
asynchronous balanced mode, 23
asynchronous digital subscriber line, viii, xiv, 70, 78, 101, 175, 211, 212, 213, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 274, 277, 278
asynchronous response mode, 23
Asynchronous Transfer Mode, xi, 3, 4, 55
ATM adaptation layer, vii, ix, xii, xiv, 53, 62, 63, 65, 66, 68, 76, 81, 82, 83, 84, 85, 86, 87, 88, 89, 91, 92, 95, 96, 97, 134, 146, 164, 171, 184, 188, 194, 195, 196, 205, 206, 207, 222, 224, 229, 231, 235, 241, 242, 243, 244, 245, 246, 248, 262, 270, 273, 277, 281
ATM anycast capability, xv, 253
ATM block transfer, xiii, 147, 151, 154, 155, 156, 244, 277
ATM cell, vii, xi, 4, 53, 55, 56, 57, 58, 59, 60, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72, 76, 77, 78, 79, 80, 82, 84, 85, 86, 90, 93, 96, 97, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 113, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 158, 159, 160, 162, 163, 164, 165, 167, 168, 169, 170, 171, 172, 173, 174, 195, 199, 203, 205, 206, 207, 235, 242, 243, 244, 277, 278, 279, 280, 281
ATM Forum, viii, 5, 9, 10, 11, 59, 78, 79, 133, 134, 135, 136, 143, 144, 146, 158, 170, 177, 179, 194, 208, 230, 237, 242, 250, 253, 255, 257, 269, 273, 274, 275
ATM passive optical networks, 211, 213, 214, 215, 277
ATM protocol stack, vii, xii, 55, 63, 64, 231
ATM traffic descriptor IE, 241, 242
ATM transfer capability, 147, 155, 243
ATMARP protocol, 184, 185
ATOM switch, 119, 120
authority and format identifier, 237, 238, 277
automatic repeat request, 21, 22, 24, 32, 42, 43, 47, 65, 232, 253, 277
autoregressive-moving average model, 137
available bit rate, viii, xiii, 41, 62, 67, 134, 144, 146, 147, 165, 166, 168, 170, 243, 277

backward explicit congestion notification, 45, 49, 50, 51, 52, 167, 277
bandwidth enforcement, viii, 133, 147, 149, 158
banyan, 102, 105, 106, 107, 108, 109, 110, 111, 112, 113, 122, 130
Batcher sorter, 111, 112, 113, 130
bearer channels, viii, 218, 219, 220
Bernoulli, 129, 131, 140, 172, 279, 280
B-frame, 137, 277
bit stuffing, 23
bit-slicing, 117
block-coding, 73
blocking switch, 120
border gateway protocol, 198, 204, 205, 208, 277
border node, 259, 261, 265, 271
broadband access server, 224, 225, 277
Broadband bearer capability IE, 241, 243
broadband connection oriented bearer class, 243, 277
broadband connection oriented bearer class A, 277
broadband connection oriented bearer class C, 243, 277
broadband connection oriented bearer class X, 243, 277
Broadband high-layer IE, 241
broadband inter-carrier interface, 70, 277
broadband low-layer IE, 241
Broadband repeat indicator IE, 241
broadcast and unknown server, 181, 182, 183, 277
broadcast communication networks, 13
buffered banyan network, 108, 130
buffered leaky bucket, 158, 159
burst tolerance, 134, 163, 277
burstiness, 134, 135, 136, 139, 140, 154, 159, 172

cable modem, 70, 212
call admission control, viii, xiii, 61, 68, 133, 141, 144, 147, 148, 149, 150, 269, 277
call clearing, 244, 245, 247, 270
call establishment, 244, 245, 246, 270
CALL PROCEEDING, 245, 246, 247, 249, 270
carrier, 26, 76, 278, 279, 280
Cause IE, 242
cell delay variation tolerance, 134, 136, 137, 142, 144, 145, 146, 159, 160, 163, 171, 277
cell delineation, 71, 72, 76, 78
cell error rate, 143, 277
cell loss priority bit, xii, 48, 62, 71, 164, 242, 243, 278
cell loss rate, 66, 67, 136, 141, 143, 144, 145, 146, 148, 150, 151, 153, 154, 165, 171, 172, 173, 243, 244, 278
cell misinsertion rate, 143, 144, 278
cell transfer delay, 141, 142, 143, 151, 171, 278
central office, 70, 83, 211, 215, 217, 277, 278
child peer group, 260, 263, 268
circuit emulation service, 83, 87
circuit-switched networks, 13
classical IP and ARP over ATM, viii, 175, 177, 184, 191, 192
Classless inter-domain routing, 35, 36, 37, 277
class-of-service, 199, 203, 278
clear channel, 78
Clos network, xii, 105, 113, 114
cluster, 187, 188, 189, 190
ClusterControlVC, 188, 189, 190
committed information rate, 47, 52, 277
common part sublayer, 81, 89, 90, 91, 92, 93, 94, 95, 96, 97, 171, 278
competitive local exchange carrier, 221, 278
complex node, 262
CONNECT, 244, 245, 247, 249, 269, 270
CONNECT ACKNOWLEDGEMENT, 245, 270
connection identifier, 45, 46, 57, 59, 60, 167, 168, 169, 170, 245, 246, 247, 277
Connection identifier IE, 242
connectionless network access protocol, 278
connectionless network interface protocol, 278
connectionless server, 278
connectionless service, 28, 32, 83, 95, 179
connection-mode service, 28
consolidated link layer management, 51, 278
constant bit rate, 67, 82, 83, 84, 86, 124, 125, 128, 131, 132, 138, 139, 144, 145, 146, 149, 171, 243, 277
constraint routing, 207, 208, 278
constraint routing LDP, 208, 278
continuous-state leaky bucket algorithm, 160

control direct VCC, 182, 183
control distribute VCC, 182, 183
control plane, 231, 270
convergence sublayer, xii, 68, 81, 84, 85, 86, 87, 89, 92, 95, 190, 231, 278, 282
correlation, 134, 136, 140
CPS-packet, 89, 90, 91, 97, 171
CPS-PDU, 89, 90, 91, 92, 93, 94, 95, 96, 97, 171
crankback, 271
cross-bar, xii, 102, 104, 105, 107, 108, 115, 120
current cell rate, 167, 169, 277
customer premises equipment, 6, 274
cyclic redundant check, 20, 21, 24, 50, 63, 72, 80, 85, 91, 94, 97, 220, 278

data communication equipment, 7, 29, 30, 31, 278
data country code, 237, 278
data direct VCC, 182, 183
data plane, 231
data terminal equipment, 7, 29, 30, 31, 38, 278
datagrams, 14, 15, 32, 37, 222
data-over-cable service interim specification, 212, 278
designated transit list, 269, 271, 278
desirable minimum cell rate, 145, 278
destination-based routing., 204, 205, 207
deterministic bit rate, 146, 278
digital subscriber loop, 212, 278, 279, 281, 282
discrete multi-tone, viii, xiv, 217, 218, 219, 220, 278
DMT symbol, 219, 220
domain-specific part, 236, 237, 238, 278, 279
dotted decimal notation, 34
downstream tag allocation, 203, 204
downstream tag allocation on demand, 203
DROP PARTY, 248, 250
DROP PARTY ACKNOWLEDGMENT, 248, 250

early packet discard, 164, 206
echo cancellation, 218
Electronics Industries Association, 8
end system identifier, 238, 239, 278
End-to-end transit delay IE, 242, 244
equivalent bandwidth, 151, 152, 153, 154, 171
error control, xii, 15, 17, 27, 28, 43, 56, 63, 65, 79, 91, 279
error detection, 17, 19, 20, 21, 84, 85, 232
Ethernet, 4, 13, 16, 27, 101, 126, 127, 178, 179, 180, 199, 205, 207, 274
Exchange Carriers Standards Association, 8
explicit forward congestion notification, 49, 50, 62, 168, 169, 170, 278
explicit rate, 167, 168, 169, 170, 203, 206, 278
explicit routing, 207, 209
Extended QoS parameters IE, 242, 244
external blocking, 102, 104, 107, 109, 114, 120, 121, 130

fast path, 219, 220, 225
fast reservation protocol, 155, 279
FDDI, 4, 73, 75, 77, 101
fiber channel, 78
fiber to the basement, 214, 279
fiber to the cabinet, 214, 279
flow attribute notification protocol, 194
folded switch, 100
foreign address, 267, 268
forward equivalent class, 199, 200, 201, 202, 203, 204, 205, 207, 220, 278
forward explicit congestion notification, 45, 49, 50, 51, 52, 62, 278
forwarding information base, 198, 199, 200, 201, 203, 204, 278, 282
frame check sequence, 20, 24, 28, 38, 46, 63, 72, 85, 91, 94, 96, 278
frame relay, vii, xi, 1, 4, 6, 10, 32, 41, 42, 43, 44, 45, 46, 47, 49, 52, 56, 62, 67, 68, 70, 83, 138, 192, 199, 222, 224, 225, 243, 269, 279
frame relay access devices, 45, 279
Frame Relay Forum, 5, 10
frame relay network device, 45, 46
frame relay UNI, vii, xi, 41, 44, 45
frequency division multiplexing, 218, 278
full service access network, 213
FUNI, 76

general switch management protocol, 196, 198, 279
generic cell rate algorithm, viii, xiii, 137, 142, 158, 159, 160, 163, 166, 279
generic flow control, xii, 59
go-back-n ARQ, 21
guaranteed frame rate, 67, 144, 146, 147, 279

header conversion, 57, 60, 101, 107, 116, 130
header error control, xii, 56, 63, 71, 72, 79, 85, 91, 279
head-of-line blocking, 105, 108, 120, 121
HEC cell generation, 71
hello packet, 258, 262, 266
hello protocol, 258, 261
high data rate DSL, 212, 213, 279
high-level data link control, xi, 13, 22, 23, 24, 28, 29, 38, 44, 45, 223, 279
high-order DSP, 238, 258, 279
Hitachi, 116, 117
hop-by-hop routing, 42, 43, 207, 209, 269
host map, 188, 189, 190, 191, 251
hybrid fiber coaxial, 212, 279

IEEE Standards Association, 8
I-frame, 24, 137, 173, 279
implementation agreements, 10
InATMARP, 184, 185, 186, 279
incumbent local exchange carrier, 221, 279
indication primitive, 235
induced uplink, 265, 271
information element, 240, 241, 242, 243, 244, 245, 246, 247, 248, 251, 252, 253, 254, 269, 271, 279
information frame, 24
initial domain identifier, 237, 279
initial domain part, 236, 237, 238, 258, 279
input buffering switch, 100, 105, 120
Institute of Electrical and Electronics Engineering, 5, 8, 27, 180, 184, 238, 239
interim interswitch signalling protocol, 255, 279
interleaved path, viii, 219, 220, 225
internal blocking, 102, 104, 107, 110, 112
international code designator, 237, 238, 279
International Electronical Commission, 7, 241
International Organization for Standardization, 5, 7, 8, 16, 22, 29, 137, 237, 241, 279
International Telecommunication Union, 5, 6, 279
International Telegraph and Telephone Consultative Committee, 6, 277
Internet Architecture Board, 8
Internet Assigned Numbers Authority, 9
internet control message protocol, xi, 37, 279
Internet Engineering Steering Group, 8, 9
Internet Engineering Task Force, viii, 5, 8, 9, 11, 177, 178, 183, 192, 194, 198, 206, 208, 222, 273, 274
internet group management protocol, 186, 279
internet protocol, xi, 9, 13, 32, 279
Internet service provider, 4, 36, 221, 279
Internet Society, 8, 9
Internet-draft., 9
interrupted Bernoulli process, 140, 172, 173, 279
interrupted fluid process, 141, 152, 279
interrupted Poisson process, 140, 141, 279
interworking function, 87
interworking unit, 69, 279
IP, viii, xi, xiii, xiv, 4, 9, 10, 15, 20, 32, 33, 34, 35, 36, 37, 38, 39, 42, 43, 55, 65, 66, 67, 138, 175, 177, 178, 179, 183, 184, 185, 186, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 212, 217, 222, 223, 224, 225, 238, 251, 269, 273, 279, 280
IP flow, 195, 196, 197, 198, 209
IP switching, viii, xiv, 175, 177, 178, 194, 195, 196, 197, 199, 203, 207, 209
IP version 4, 32
IP version 6, xi, 38, 199, 206
Ipsilon's flow management protocol, 196, 197, 198, 279
ISDN DSL, 212, 279
ITU Development Sector, 6
ITU Radiocommunications Sector, 6
ITU Telecommunications Standardization Sector, ix, 4, 5, 6, 8, 9, 11, 26, 29, 41, 44, 46, 51, 55, 56, 57, 58, 59, 72, 74, 76, 134, 136, 146, 158, 213, 215, 216, 229, 230, 241, 243, 269, 273, 274

jitter, 129, 141, 142, 171
Joint Technical Committee 1, 7

L2TP access aggregation, xiv, 221, 222, 223
L2TP access concentrator, 222, 223, 224, 279

label, viii, xiv, 60, 61, 79, 80, 155, 157, 175, 177, 178, 194, 197, 198, 199, 206, 207, 208, 209, 280
label distribution protocol, 208, 274, 280
label edge router, 206, 207, 280
label stack, 207, 209
label switched path, 206, 207, 208, 280
label switching routing, 206, 207, 280
LAN emulation, viii, xiii, 175, 177, 178, 179, 180, 181, 182, 183, 188, 194, 208, 253, 280
LAN emulation address resolution, 182, 183, 280
LAN emulation client, viii, 175, 177, 179, 180, 181, 182, 183, 253, 280
LAN emulation server, 180, 181, 182, 183, 253
LAN emulation user to network interface, 180, 181, 280
layer 2 tunnel protocol, xiv, 221, 222, 223, 224, 225, 274, 279
LDP peers, 208
LDP session, 208
LE client, 180, 181, 182, 183, 253, 280
LE client identifier, 280
Leaf initiated join (LIJ) parameters IE, 251
Leaf initiated join call identifier IE, 251
leaf initiated join capability, xiv, 229, 230, 250, 251, 252, 253, 254, 280
Leaf sequence number IE, 251
LEAF SETUP FAILURE, 251
LEAF SETUP REQUEST, 251, 252, 253
leaf-prompted join without root notification, 250
level indicator, 258, 268
link aggregation, 262
local management interface, 280
local multipoint distribution services, 211, 280
logical group node, 260, 261, 262, 264, 265, 266, 268, 269, 271
logical link, xi, 8, 13, 27, 179, 260, 261, 262, 269, 271
logical link control, xi, 8, 13, 27, 28, 179, 180, 183, 184, 188, 196, 199, 222
longitudinal redundancy check, 19
loose routing, 208

mapping entity, 280
Markov modulated Bernoulli process, 140, 280
Markov modulated Poisson process, 141, 280
maximum burst size, 134, 135, 136, 145, 146, 149, 163, 171, 173, 174, 242, 280
maximum cell transfer delay, 141, 142, 143, 144, 145, 151, 244
maximum frame size, 146, 280
maximum transfer unit, 37, 184, 189, 280
medium access control, 27
minimum cell rate, 146, 165, 168, 169, 278, 280
modified time-division multiplexing algorithm, 119, 125, 126
MPEG, 137, 173
multicast address resolution server, 188, 189, 190, 191, 208, 251, 280
multicast forward VCC, 182
multicast send VCC, 182, 183
multicast server, 181, 182, 187, 188, 190, 191, 251, 280
multicasting, xiv, 34, 121, 122, 123, 181, 186, 187, 188, 192, 251, 273
multihomed, 35
multi-protocol label swapping, viii, xiv, 175, 177, 178, 194, 198, 206, 207, 209, 280
multi-protocol over ATM, 194, 280
multi-stage interconnection network, 101, 105, 111, 113, 120

N^2 disjoint paths, xii, 114, 121
narrowband ISDN, 41
native address, 267, 268
network access server, 222, 224, 225, 280
network LIJ connection, 250, 251, 252
network node interface, 58, 59, 69, 255, 256, 269, 280, 281
network service access point, 67, 236, 237, 238, 239, 253, 256, 258, 267, 268, 280
network service provider, viii, xiv, 175, 211, 221, 222, 224, 225, 280
network timing reference, 220, 280
next hop client, 192, 280
next hop routing protocol, viii, xiv, 175, 177, 178, 191, 192, 193, 194, 209, 273, 280
next hop server, 192, 280
nodal aggregation, 262
non broadcast multiaccess network, 192, 193, 273, 280
non-associated signalling, 236

non-blocking switch, 118, 120, 121, 123, 149, 150, 151, 152, 165, 170, 171
non-real-time statistical bit rate, 147, 280
non-real-time variable bit rate, 67, 124, 125, 131, 144, 145, 147, 280
non-statistical allocation, 149
NOTIFY, 245, 246, 270
Nx64 Kbps, 76, 83, 87

on/off model, 139, 173
open shortest path first, 198, 204, 205, 281
operations, administration, maintenance cell, 62
optical carrier, 8, 75, 280
optical line terminator, 213, 214, 280
optical network unit, 212, 213, 214, 281
OSI reference model, 13, 16, 27, 64
output buffering switch, 100, 120, 123, 130

packet-switched networks, 3, 13, 15, 29
parent peer group, 260, 262, 264
parity check, 19
partial packet discard, 164, 206
PARTY ALERTING, 248
passive optical network, 5, 213, 214, 274, 277, 281
payload type indicator, xii, 62, 71, 96, 97, 165, 168, 243, 281
peak cell rate, 134, 135, 136, 138, 139, 144, 145, 146, 150, 158, 159, 160, 163, 165, 169, 171, 172, 173, 242, 281
peak-to-peak cell delay variation, 141, 142, 143, 144, 145, 151, 159, 244
peer group identifier, 258, 262, 265, 268
peer group leader, 260, 261, 262, 263, 264, 265, 271, 281
peer groups, xv, 257, 259, 260, 261, 262, 263, 264, 265, 268
permanent virtual circuit, 30, 46, 61, 65, 147, 186, 194, 195, 196, 197, 222, 224, 225, 229, 281
P-frame, 137, 281
physical medium dependent, xii, 64, 71, 73, 79, 281
plesiochronous digital hierarchy, 26, 55, 74, 76, 281
PNNI call control, 270
PNNI protocol control, 270
PNNI routing protocol, ix, xv, 69, 255, 256, 262, 269
PNNI signalling protocol., ix, 255
PNNI topology state elements, 258, 265, 281
PNNI topology state packet, 258, 281
point-to-point protocol, xiv, 199, 207, 221, 222, 223, 224, 225, 274, 281
Poisson, 129, 140, 141, 279, 280
PPP terminated aggregation, xiv, 221, 224
preventive congestion control, viii, 68, 133, 147, 164
private network-network interface, ix, xv, 69, 205, 227, 231, 233, 239, 247, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 275, 281
protocol independent multicast, 198, 205, 208, 281
public UNI, 70
pulse code modulation, 25, 26, 281
push-out scheme, 164

Q.2931, ix, xiv, 6, 205, 229, 230, 231, 233, 239, 241, 244, 245, 246, 247, 248, 270, 274
Q.2971, ix, xiv, 7, 229, 230, 231, 233, 239, 247, 248, 250, 251, 253, 254, 270, 274
quadrature amplitude modulation, 218, 281
quality of service, viii, xiii, 55, 61, 66, 67, 77, 123, 124, 125, 131, 132, 133, 141, 142, 143, 144, 145, 148, 150, 158, 173, 178, 192, 203, 242, 244, 246, 254, 268

raw cell, 83
reachability, 255, 257, 262, 266
reactive congestion control, viii, 68, 133, 147, 165
real-time statistical bit rate, 147, 281
real-time variable bit rate, 67, 124, 125, 131, 132, 144, 145, 146, 281
regional operations center, 221, 281
RELEASE, 245, 246, 247, 249, 270
RELEASE COMPLETE, 245, 246, 247, 270
remote authentication dial in user service, 224, 225, 274, 281
request for comments, 9, 273, 274
resource management, 62, 165, 166, 167, 168, 169, 170, 281
resource reservation protocol, 205, 208, 281
reverse address resolution protocol, 37, 184

RM cell, 62, 165, 166, 167, 168, 169, 170
root LIJ connection, 251, 252
root-prompted join, 250
round-robin scheduler, 125
routing control channel, 262, 281

segmentation-and-reassembly sublayer, xii, 68, 81, 84, 85, 89, 92, 93, 94, 95, 96, 97, 231, 281
selective-reject ARQ, 21, 22, 32, 253
selector, 238, 239, 256, 260, 281
ServerControlVC, 190
service access point, 28, 92, 236, 280
service data unit, 62, 96, 281
service specific convergence sublayer, 81, 89, 90, 92, 95, 96, 97, 171, 231, 242, 282
service-specific connection function, 231, 234, 282
service-specific connection oriented protocol, xiv, 66, 229, 231, 232, 233, 234, 235, 253, 282
SETUP, 61, 148, 244, 245, 246, 248, 249, 251, 252, 253, 256, 268, 269, 270
S-frame, 24
shared medium switch, viii, 99, 119, 120, 121, 122, 130
shared memory switch, viii, 99, 101, 115, 116, 117, 118, 121, 122, 126, 130
shim label header, 207
shim tag header, 199, 202, 205, 207
signalling AAL, ix, xiv, 81, 83, 95, 229, 231, 233, 234, 235, 270, 281
signalling channel, xiv, 236, 270
signalling protocol, viii, xiv, xv, 46, 61, 64, 65, 66, 68, 70, 81, 83, 205, 227, 229, 230, 231, 233, 234, 235, 236, 239, 241, 247, 251, 253, 255, 256, 269, 270, 279
signalling protocol stack, xiv, 230, 231
single logical IP subnet, 183, 184, 185, 191, 193, 280
sliding window-flow control, 18, 21
soft PVC, 61, 271
SONET, 73, 74, 75, 76, 88, 281
source routing, 34, 181, 204, 269
space-division switch, viii, 99
splitterless ADSL, 216, 225
Standard RFC, 9
standards committees, vii, 3
statistical allocation, 149, 150
statistical bit rate, 147, 280, 281
STATUS, 232, 245, 246, 270
STATUS ENQUIRY, 245, 246, 270
stop-and-wait, 17, 18, 21
stop-and-wait ARQ, 21
STP, 75
strict routing, 208
structured data transfer, 242
subnet mask, 35, 39, 178, 183
subnetting, 35, 36
summary address, 267, 268
supernetting, 36
supervisory frame, 24
sustained cell rate, 134, 135, 136, 138, 145, 150, 151, 158, 160, 163, 171, 173, 174, 242, 281
switch fabric, 99, 100, 103, 104, 108, 112, 114, 130
switched communication network, 13
switched multimegabit data service, 4, 70, 92, 192, 281
switched virtual circuit, viii, 30, 46, 61, 65, 147, 148, 150, 181, 186, 227, 229, 231, 241, 255, 260, 261, 262, 265, 282
symmetric DSL, 212, 213, 281
synchronous data link control, 22
synchronous digital hierarchy, 74, 75, 76, 281
synchronous residual time stamp, 88
synchronous time division multiplexing, 13
Synchronous Transfer Mode, 55, 215, 219, 220, 282
synchronous transport signal level 1, 74, 75, 76, 282

tag distribution protocol, 203, 282
tag edge router, 199, 200, 201, 203, 206, 282
tag forward information base, 200, 201, 202, 203, 204, 282
tag stack, 199, 204, 205
tag switched path, 202, 206
tag switching, viii, 175, 177, 178, 194, 197, 198, 199, 200, 203, 204, 205, 206, 207, 282
tag switching router, 199, 200, 201, 202, 203, 204, 205, 206, 282
TAXI, 75, 77, 178
TCP, 9, 13, 15, 20, 22, 32, 33, 37, 42, 44, 50, 66, 67, 68, 83, 96, 146, 164, 208
TCP/IP, 13, 15, 20, 32, 37, 44, 66, 67, 68, 83
the PNNI signalling protocol, ix, 69, 255
threshold scheme, 164

time division multiplexing, xi, 13, 24, 25
time-to-live, 199, 282
token ring, 8, 27, 77, 101, 179, 180, 199, 205, 207
traffic contract, 158
Transit network selection IE, 242
transmission convergence, xii, 7, 64, 71, 73, 79, 282
transparent VP service, 243

unacknowledged connectionless service, 28
unbuffered leaky bucket, 158
unfolded switch, 100
unnumbered frame, 24
unspecified bit rate, 67, 124, 125, 128, 131, 132, 144, 145, 147, 243, 273, 282
unstructured data transfer, 84, 86, 87, 97
uplink, 261, 265
upnode, 261, 265
upstream tag allocation, 203
user network interface, ix, xiv, 7, 45, 46, 58, 59, 69, 70, 76, 142, 155, 157, 158, 159, 160, 163, 181, 227, 229, 231, 233, 234, 241, 242, 246, 247, 250, 251, 252, 253, 255, 269, 273, 274, 282
UTOPIA, xii, 78, 79
UTP, 75

VC merging, 205, 206
VC mesh, 187, 188, 190, 191, 251
very high data rate DSL, 212, 213, 214, 282
violation tagging, 163, 243
virtual channel connection, 59, 65, 97, 157, 181, 182, 183, 196, 282
virtual channel identification, xii, 59, 60, 61, 65, 71, 101, 155, 157, 196, 197, 198, 199, 202, 203, 205, 207, 236, 240, 242, 243, 262, 269, 270, 282
virtual circuits, 14, 15, 30
virtual path identification, xii, 59, 60, 61, 65, 71, 101, 155, 157, 196, 197, 198, 199, 202, 203, 205, 207, 236, 240, 242, 262, 269, 270, 282
virtual paths, 59, 236
virtual scheduling algorithm, 160, 162, 171

wavelength division multiplexing, 214, 282
window-flow control, 17, 18, 21
WIRE, xii, 78, 79

X.25, xi, 7, 13, 15, 28, 29, 30, 31, 32, 38, 42, 43, 44, 46, 192
X3S3, 8
x-type digital subscriber line, 70, 78, 211, 212, 213, 282

About the Author

Harry G. Perros received the B.Sc. degree in Mathematics in 1970 from Athens University, Greece, the M.Sc. degree in Operational Research with Computing from Leeds University, England, in 1971, and the Ph.D. degree in Operations Research from Trinity College Dublin, Ireland, in 1975.

From 1976 to 1982 he was an Assistant Professor in the Department of Quantitative Methods, University of Illinois at Chicago. In 1982 he joined the Department of Computer Science, North Carolina State University, as an Associate Professor, and since 1988 he is a Professor. He has spent sabbaticals at INRIA, Rocquencourt, France, University of Paris 6, France, and NORTEL, Research Triangle Park, North Carolina.

He has published extensively in the area of performance modelling of computer and communication systems, and he has organized several national and international conferences. He has also published a monograph entitled "Queueing networks with blocking: exact and approximate solutions", Oxford Press. He is the chairman of the IFIP Working Group 6.3 on the Performance of Communication Systems. In his free time, he likes to sail on board the *Aegean*, a Pearson 31!