8
BROADCAST NETWORKS

8.1 MORE ON MEDIUM ACCESS CONTROL

We have seen that the issue in broadcast networks is medium access rather than routing. Broadcast networks can be divided into those which use guided technology and those which are wireless. The former category includes most LANs and MANs, the latter, satellite networks, wireless LANs and cellphone networks. In each case, the MAC problem boils down to finding a fair means of giving multiple access to a shared medium. We have already studied this problem in Section 4.6, and several approaches were discussed. To summarise, the bandwidth of a shared medium can be divided up by frequency (FDMA, or broadband) or time (TDMA, or baseband). In either case, division may be static (pre-assigned) or dynamic (demand-assigned), and dynamic assignment can be managed in a centralised or distributed fashion.

Static division of a shared medium is relatively straightforward in principle, although there is a question of how the apportionment should be made. The obvious solution is to divide the total bandwidth of the medium, B (bits per second) say, into equal portions between the competing stations. If there are n such stations, each gets a dedicated slice of capacity, B/n, always available to it, and this might be done using FDM or synchronous TDM. The main problem is the inflexible nature of static assignment—some stations may waste their bandwidth allocation at certain times, while others could quite easily use more than they have been given.

A simple calculation from queuing theory verifies the advantage of dynamic distribution, under some reasonable assumptions. Suppose again that n stations are all producing packets for transmission into a shared medium of capacity B, dynamically allocated according to some arbitration algorithm (Figure 8.1a). First of all, note that

this is a queuing problem, with the packets playing the role of arriving *customers* and the medium the role of *server* (in the sense of the "thing" serving the queue). A packet may be thought of as joining the queue as soon as it is scheduled for transmission at a station. It is granted access to the server according to some mechanism that, we can assume for the moment, yields a first-come-first-served behaviour (a FIFO queue). If the packet length is L bits, it will take L/B seconds to transmit. However, what is really of interest is the waiting time, T, which an average packet experiences from the moment it joins the queue until it is sent.

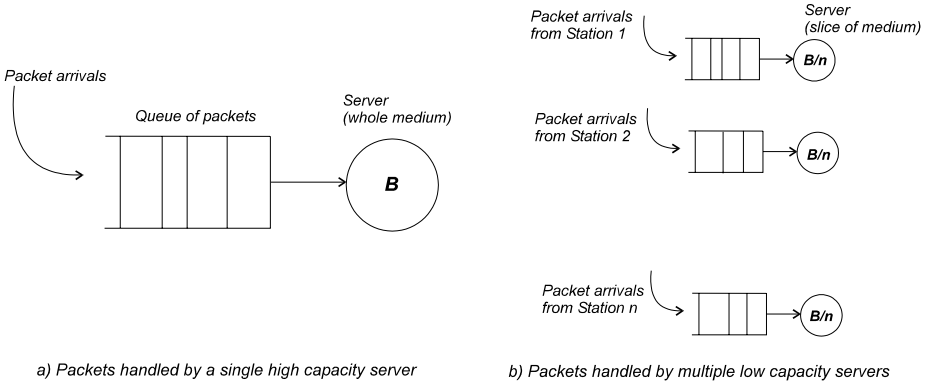


Figure 8.1 Static versus dynamic allocation

To find a value for T, we must make some approximations concerning the way in which packet arrivals and lengths are distributed. The simplest and most widely used assumptions are that arrivals obey a Poisson process, with average λ packets/sec, so that the probability of k packet arrivals in an interval of time t seconds long, is given by:

$$P(k) = \frac{(\lambda t)e^{-\lambda t}}{k!} \qquad 8.1$$

while packet lengths are *exponentially* distributed, with average L, so that the probability, $P(x)\mathrm{d}x$ of a packet having a length between x and $x+\Delta x$, is given by

$$P(x)dx = \frac{1}{L}e^{-\frac{x}{L}}dx \qquad 8.2$$

Poisson distributions arise if we assume that all arrivals are independent and that the probability of k arrivals is the identical for all time intervals of the same length. Exponential distributions are such that shorter packets are more probable than longer ones, and originate under the following idealised conditions. Suppose that, observing a station transmitting a packet, we are interested in how much of the packet is left. Suppose also that the probability of it finishing in the next s seconds is completely independent of how much of it has already been transmitted, then the length distribution is exponential.

Queues with Poisson arrivals, one server, and exponentially distributed service times are the easiest to analyse, and are known in queuing theory as *M/M/1* queues. A relatively easy result shows that the average waiting time in such a queue is given by:

$$T = \frac{1}{B/L - \lambda} \tag{8.3}$$

where λ is the average arrival time, and *B/L* the average service time (time to transmit the packet). In this queue, a packet, on reaching the head, gets the full bandwidth of the channel until it is transmitted, equivalent to a completely dynamic allocation of capacity.

However, if the channel is divided statically into *n* identical sub-channels, each with capacity *B/n*, this is equivalent to *n* separate identical queues, each with arrival rates of λ/*n,* and average service times of *B/*(*nL*) as shown in Figure 8.1b. Note that the servers in this system of queues are slower by a factor of *n*, but each receives only 1/*n* of the traffic. Using Equation 8.3 again we see that,

$$T_{shared} = \frac{1}{\frac{B}{nL} - \frac{\lambda}{n}} = nT \tag{8.4}$$

Comparing 8.3 and 8.4, the expected waiting time for an average packet is *n* times worse with static as opposed to dynamic demand allocation. Nothing in this argument has assumed FDM rather than TDM or vice versa, so the result holds in both cases. Of course it must be borne in mind that very specific assumptions have been made here. For example, it is assumed that each station can transmit at the full rate, *B* bps, of the medium, once granted access to it: this may not, in fact, be possible if *B* is very large.

Unsurprisingly, therefore, in most situations where the bandwidth of the medium is precious, it is desirable to use demand allocation. This is especially true in baseband systems, and time-division demand allocation is therefore of particular importance. This is a problem we have already looked at in some contexts, and we have seen that a medium can be allocated to competing data sources (stations) via either slotted or unslotted asynchronous TDMA. The problem of medium access control in these circumstances has many possible solutions, although some are more appropriate than others to certain circumstances. One important point is that some TDMA MAC strategies rely on the propagation delay across the medium being reasonably short. Such methods may work well in LANs, yet be inappropriate on long delay media such as those in satellite, radio or even MANs. We shall shortly conduct a brief survey of the most important MAC approaches used with TDMA, pointing out their applicability as we progress. FDMA methods often rely on in-band TDMA to subdivide the frequency bands formed by FDM, so it is natural to discuss TDMA first. We will, however, also examine FDMA later in the context of MANs and broadband wireless networks.

Broadcast networks are encountered in a number of important practical contexts, some of which have been mentioned previously in the text. One obvious circumstance is when stations can broadcast free-space radio or microwave signals, in “wireless” or satellite networks. Another is where high bandwidth cable is used to form a *serial bus*,

the commonest basic configuration for LANs and MANs. A serial bus differs from an internal system bus in two respects: firstly, there is typically only a single conducting path, as opposed to the normal internal parallel configuration; and, secondly, the distances between stations are significantly greater then those between internal subsystems. Other LAN technologies, like rings, can also be considered to behave as broadcast media even though, at the lowest level, they are in fact composed of point-to-point links. More recently, LANs and MANs based on optic fibre rings or buses have appeared. Optical buses in particular have proved difficult to implement because the energy drained from a light signal by optical receivers has been too great to allow many stations to be attached to a single fibre.

In what follows, let us assume a single uninterrupted TDMA channel, which any station can transmit into, and let us assume further that if two stations transmit at once, there will be a collision which will destroy both packets. MAC methods, as we saw in Section 4.6, can be divided into those that attempt to resolve such collisions and those that attempt to avoid them in the first place. We shall now take a close look at each of these categories.

8.1.1 Collision Resolution MAC

Collision resolution or *contention* MAC strategies rely on accepting that collisions will occur, and then coping with the consequences. There are a number of methods under this general umbrella, ranging from those which make no attempt at all to avoid collisions to those which try their best, but occasionally fail to do so.

ALOHA

The most basic approach is simply to allow any station that has a packet ready to send to transmit it, without it making any attempt to monitor activity on the medium. This technique was first used in the prototype packet radio network, *ALOHANET*, commissioned in Hawaii in1970, and has been known ever since, as *pure* or *unslotted ALOHA*. In pure ALOHA, packets carry some form of error-detecting checksum, which is verified at the receiver. If a packet is correctly received, the destination returns a short acknowledgement immediately. However, if the packet suffers a collision, it will either be destroyed or corrupted, and no acknowledgement will be received. A station that has sent a packet waits for a time equal to twice the maximum possible propagation delay on the network, and, if no acknowledgement is forthcoming, it assumes that a collision has indeed occurred. It then sends the packet again, after a random interval, and is said to *back-off*. After repeated failures it gives up.

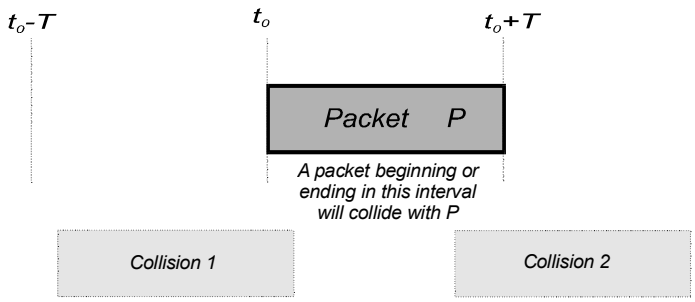


Figure 8.2 Collisions in pure ALOHA

Since transmitting stations ignore each other, a given packet, *P,* will experience a collision if some other station sends a packet whose transmission overlaps with that of *P*. The condition for this can be easily established as follows. Suppose that all packets sent in a given ALOHA network are of the same standard length, each one taking any station *T* seconds to transmit. Now suppose that a station, *S*, begins to send packet *P* at time t_0. It should be clear that any packet that begins transmission in the interval [t_0-T, t_0+T] will collide with *P* (Figure 8.2). The probability that *P* will suffer a collision is just the probability that during this interval, another packet will become ready to send at some station. We shall call this probability p_{coll}, and shall now investigate its value.

A packet in an ALOHA network can become ready to send either if it is a new arrival, or if it is a packet which has previously failed and been deferred for some random interval which has now expired. The rate at which new packets become ready to send in any network is called the *arrival rate*. The rate at which packets, including retransmissions, are injected into the network, as has been mentioned before, is the *offered load*. Obviously the offered load will be greater than the arrival rate, since some packet transmissions will fail. When a transmission fails, the station that has generated it becomes *backlogged*, i.e. its priority is to retransmit the failed packet rather than deal with new ones.

Suppose we now assume that the arrival rate is Poisson with a fixed average of λ packets/sec from all stations. For this to be valid, we also need to assume that backlogged stations do not significantly affect the arrival rate (this can be done by assuming an infinite number of stations). The offered load is of course dependent, not only on the arrival rate, but also on the retransmission strategy. It turns out that if we set the retransmission time for a failed packet to be exponential, with average τ, then the offered load turns out to be Poisson as well, with average $\lambda'=\lambda+n/\tau$ packets/sec, where *n* is the mean number of backlogged stations.

We have already seen that in any network, the throughput is the number of packets delivered successfully per unit time. It is convenient to take the unit of time as equal to the time taken to transmit a packet, *T*. We can then define the *normalised throughput*, *S*, as the number of packets delivered in *T* seconds, and the *normalised load G*, as the number of packets sent in *T* seconds. Then:

$$G = \lambda' T \qquad 8.5$$

S is just the number of packets delivered successfully in one packet time, and is, therefore, equal to the number of packets sent in that time (i.e. G) multiplied by the probability of success.

$$S = G(1 - p_{coll}) \qquad 8.6$$

The probability of a packet being successfully transmitted, $1 - p_{coll}$, is just the probability of no other packets being sent in the time interval [t_0-T, t_0+T] and is, from Equation 8.1, $e^{-\lambda' 2T} = e^{-2G}$. Thus

$$S = Ge^{-2G} \qquad 8.7$$

This gives a relationship between the throughput and offered load for the ALOHA system. It is instructive to graph Equation 8.7 (Figure 8.3). It can be seen that the throughput reaches a maximum for G=0.5 (this is easily verified by differentiating Equation 8.7). At this point S takes the maximum value $1/2e$, approximately 0.184.

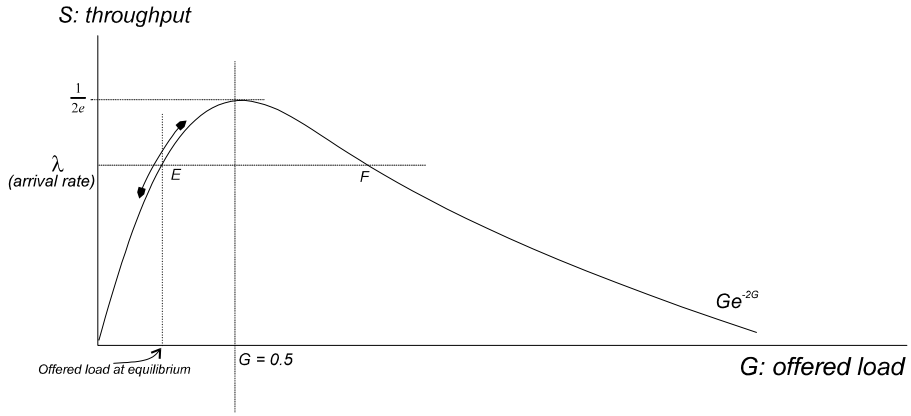


Figure 8.3 Throughput versus offered load for pure ALOHA

Let us be clear about what the graph of Figure 8.3 means. Assume for the moment that the arrival rate in the system is kept steady at λT. Even if this is the case, however, the offered load, G, is not steady, because it varies with the number of backlogged stations. Suppose we try to operate the system at the point E, indicated in Figure 8.3. As fluctuations in offered load occur, the point of operation will slide up and down the graph away from E. Fortunately, if the offered load increases slightly, from E, the shape of the graph shows that the throughput increases as well, tending to clear any increased backlog, and bring the operating point back to E again. The same applies if the offered load drops temporarily. E is called a point of *stable equilibrium*.

If we try to increase the arrival rate to move E further towards the peak of the graph, we can observe firstly that the percentage of successful packets will never exceed 18.4%. However, once we pass the peak, the system begins to behave badly. If the operating point passes G=0.5, any increase in offered load, however temporary, will immediately reduce throughput, causing a further increase in offered load and so

on. Although, if everything were entirely stable, it would be possible to operate the system in equilibrium in this region, say at point *F*, the slightest perturbation will cause a collapse of throughput as offered load shoots up. *F* is a point of *unstable equilibrium*.

Because perturbations in offered load are always occurring, it is not sensible even to try and push the stable operating point too close to the maximum, since, if it is carried past that maximum by a perturbation, a collapse will ensue. The theoretical peak of 18.4% of successful transmissions is therefore optimistic—a real ALOHA system will not achieve even this modest level.

This analysis, although useful, is rather imprecise. For example, in a real system, an increase in the number of backlogged stations will cause a decrease in the arrival rate, so the assumption of constant arrival used above is not consistent. A more detailed model using Markov chains can be used to take the varying number of retransmissions into account, although, in fact, the results, while more rigorously obtained, are similar. This is beyond the scope of the current book, although see Bertsekas and Gallager (1992) for further detail.

The best possible performance of ALOHA, at less than 18%, may seem somewhat disappointing. Remember, however, that it is a totally hit or miss undisciplined methodology. Its attraction is in its simplicity, for it requires no supporting MAC mechanism whatsoever. Perhaps unsurprisingly, pure ALOHA can be improved upon fairly easily in the right circumstances. One of the simplest ways of doing this is to divide the medium bandwidth into time slots, delimited by common clock, each long enough to accommodate one packet and its subsequent acknowledgement. In this *slotted ALOHA*, a station that has a packet to send must await the beginning of the next slot before it can begin transmission. If the acknowledgement has not been returned by the end of a slot, the station assumes that the transmission has been unsuccessful, and becomes backlogged for some random number of slots. Otherwise, the stations behave pretty much just as in pure ALOHA.

With the slotted discipline, a packet which is transmitted in a slot of length T, starting at time t_0, can only collide with another which was transmitted in the same slot as itself, i.e. in the interval $[t_0, t_0+T]$. Using the same simplified analysis as for pure ALOHA yields:

$$S = Ge^{-G} \tag{8.8}$$

as an analogue of Equation 8.7. The maximum of S, now occurs at $G = 1$, and takes a value of approximately 0.36. Hence, at most 36% of channel capacity may be used for successful data transmission, twice the performance of pure ALOHA. However, slotted ALOHA suffers from the same stability problems as pure ALOHA, and any slotted system operating at equilibrium cannot achieve this figure. Nonetheless, a performance improvement of a factor of two is very significant. The price is that we now need a global clock which transmits slot boundary instants to all stations. There are measures which can be taken to improve stability in slotted ALOHA, such as Rivest's *Pseudo-Bayesian algorithm* (see Bertsekas and Gallager, 1992), but these are not discussed further here.

These analyses do not take account of the propagation delay across the medium. This affects the mathematics slightly because, for example, in pure ALOHA, a packet,

P, transmitted at time t_o, can still collide with another packet, Q, which begins later than t_0+T but earlier than $t_0+T+\delta$, with δ the propagation delay across the medium.

Carrier Sense Multiple Access

Carrier sense multiple access or *CSMA* is like ALOHA, but additionally, each station can detect whether the medium is idle or not, and will only transmit if it finds it so. Unfortunately, this does not completely eliminate collisions, because it is still possible for two stations that have detected an idle medium to attempt to send at approximately the same time, resulting in a collision. If a collision does occur, neither station is aware of it until acknowledgements fail to turn up at the expected time. CSMA is actually a family of protocols, the most significant varieties of which we now discuss.

In *1-persistent CSMA* a station that has a packet to send listens to the medium. If it finds the medium idle it transmits at once; if, however, it finds the medium busy, it continues to listen and transmits as soon as the medium becomes idle again (Figure 8.4). If a collision occurs, the station then becomes backlogged, and waits for a random interval of time before listening again. The problem with 1-persistence is that, if two or more stations attempt to transmit during a busy period, they will all wait for the medium to become idle, whereupon they will all transmit, guaranteeing a collision.

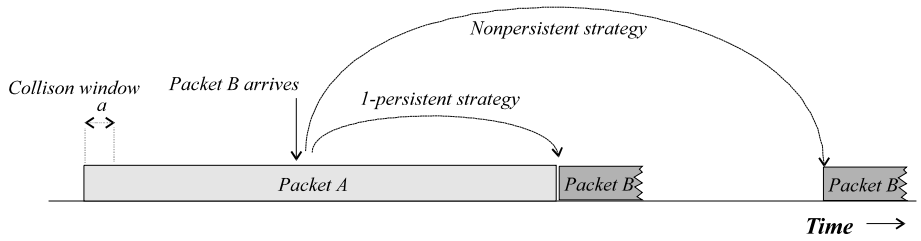


Figure 8.4 Comparison of 1-persistent and nonpersistent CSMA

The trouble with the 1-persistent strategy is that it is greedy. Any station which finds the medium busy, jumps in as soon as it is idle, with no concern as to whether there are other prospective senders or not. A more "community spirited" approach is *nonpersistent CSMA*. Here a station that has a packet to send listens to the medium, and, as before, if it finds it idle, transmits at once. If, however, the medium is busy, then, rather than continuing to listen for an opportunity to break in, the station *backs-off*, i.e. waits for a random interval, and then tries again (Figure 8.4). It seems reasonable that adopting this strategy is likely to reduce the chance of collisions in a busy network, although, of course, if only one station is waiting, the automatic back-off actually slows it down. It follows that we might expect 1-persistence to prove superior at low loading, non-persistence at high loading. In fact, this is exactly what is observed in practice as can be seen from Figure 8.5.

Both nonpersistence and 1-persistence can be subjected to analysis similar to that applied to ALOHA in the previous section. Even more than with ALOHA, the propagation delay across the medium is of great importance. This is no surprise, since when a station begins to transmit a packet, other stations are unaware that the transmission has begun until the signal has propagated across the whole medium. It follows that, until this propagation is complete, another station may unsuspectingly

begin to transmit a packet of its own, causing a collision. In other words, every packet transmitted into a CSMA network has a window of vulnerability to collision, which lasts until its first signal element has been detected by all stations (Figure 8.4). The greater the average propagation delay across the network, the longer this window will be on average and therefore the greater the probability of a collision occurring. From this it can be seen that CSMA works best in networks where the collision window is a small fraction of a packet length. To illustrate this, the average propagation delay, a, like S and G, is usually "normalised" to units of packet times for the purposes of analysis.

To minimise a, a CSMA network must have either short paths or long packets (in terms of the time taken to transmit them). Long packets are achieved by having lots of bits relative to the transmission rate. In any case, when a is small, analysis of nonpersistent CSMA yields the relationship shown on page 320.

8A IN DEPTH: NON-PERSISTENT AND 1-PERSISTENT CSMA

A mathematical analysis of CSMA can be attempted using the same definitions and concepts as that developed above for ALOHA. In general all analyses of this type operate by dividing the network time into *cycles*. A cycle includes the transmission of a packet (successfully or unsuccessfully) and the subsequent idle time (which, of course, can vary in length). The throughput is the average number of packets delivered in such a cycle, divided by the cycle's average length. Further assumptions include exponential back-off, giving a Poisson offered load with average rate G.

Attempting an analysis of this type for nonpersistent CSMA, we assume that a cycle consists of a packet transmission of length $1+a$, followed by an idle period of length $1/G$. A standard result from probability theory proves that the average interval between events in a Poisson distribution of average rate α, is just $1/\alpha$. The "a" term, comes from the average time after a station has finished transmitting until the medium is detected idle by all other stations. We assume that back-off occurs from this point and the average wait until some station decides to transmit is $1/G$.

The probability of a successful transmission in a cycle is just the probability that no second transmission will begin within time a of the start of the cycle. Using Equation 8.1 for the Poisson distribution, this is just e^{-aG}. If the transmission is successful, one packet gets transmitted; if not, no packet gets sent. Hence the average number of packets in a cycle is just:

$$e^{-aG} \cdot 1 + (1 - e^{-aG}) \cdot 0 = e^{-aG}$$

Thus :

$$S = \frac{expected\ number\ of\ packets\ sent}{expected\ length\ of\ cycle} = \frac{e^{-aG}}{(1+a)+\frac{1}{G}} \qquad 8A.1$$

Letting $a \to 0$, yields Equation 8.9.

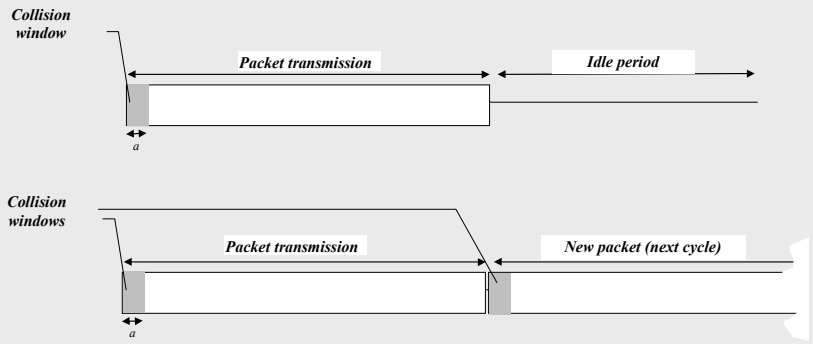


Figure 8A.1 Cycles in analysis of 1-persistent CSMA

Analysis of 1-persistent CSMA is somewhat harder, so let us assume from the outset that $a = 0$, to make the calculation a little easier. In 1-persistence, there are actually two different types of cycle, as shown in Figure 8A.1. One involves a transmission followed by an idle period (no other packet arrived and got blocked during the transmission); the other involves a transmission followed by no idle period (one or more packets arrived during the transmission, got blocked, and were ready waiting for the end of the transmission). The probability that no arrivals occur during a transmission is given by e^{-G}, so the expected length of a cycle is:

$$e^{-G}(1+\frac{1}{G})+(1-e^{-G})\cdot 1=\frac{G+e^{-G}}{G}$$

What is the probability of a successful packet transmission? Interestingly, it depends on what happened in the previous cycle. If the previous cycle was of the first type, and assuming a=0, there will be no collision. The probability of this is just that there are no arrivals in the transmission phase of the previous cycle: again this is e^{-G}. If the previous cycle was of the second type, however, the packet now being transmitted was blocked during it. In this case, the transmission will be successful, if exactly one packet got blocked—any more and a collision is guaranteed. The probability that exactly one packet will arrive during the transmission phase of the previous cycle is by Equation 8.1, Ge^{-G}.

Putting these possibilities together, the total probability of a successful transmission in the current cycle is:

$$Ge^{-G}+e^{-G}$$

Applying the same arguments as for non-persistence, and using Equation 8A.1, gives:

$$S=\frac{Ge^{-G}+e^{-G}}{\left(\frac{G+e^{-G}}{G}\right)}=\frac{Ge^{-G}(G+1)}{G+e^{-G}}$$

which is just Equation 8.10.

$$S=\frac{G}{1+G} \qquad 8.9$$

between normalised throughput, S, and offered load, G (strictly this analysis is for an exponentially distributed retransmission strategy). Likewise, in the case of 1-persistent CSMA,

$$S = \frac{Ge^{-G}(1+G)}{G+e^{-G}} \qquad 8.10$$

The derivation of these results is discussed in Panel 8A. The graphs of 8.9 and 8.10 are shown, together with some others, in Figure 8.5. As this diagram shows, even 1-persistent CSMA is a considerable improvement on slotted ALOHA, with a maximum throughput in excess of 50%. Nonpersistent CSMA is even better at all but low-levels of G, and unlike ALOHA or 1-persistence, does not suffer a drop in throughput as load increases. In fact, this last rather optimistic result relies on an assumption that $a = 0$ (see Panel 8A), which, of course, can never be true in practice. Incidentally, both non-persistence and 1-persistence can be run in slotted form, but analysis shows that, unlike ALOHA, there is nothing to be gained from this.

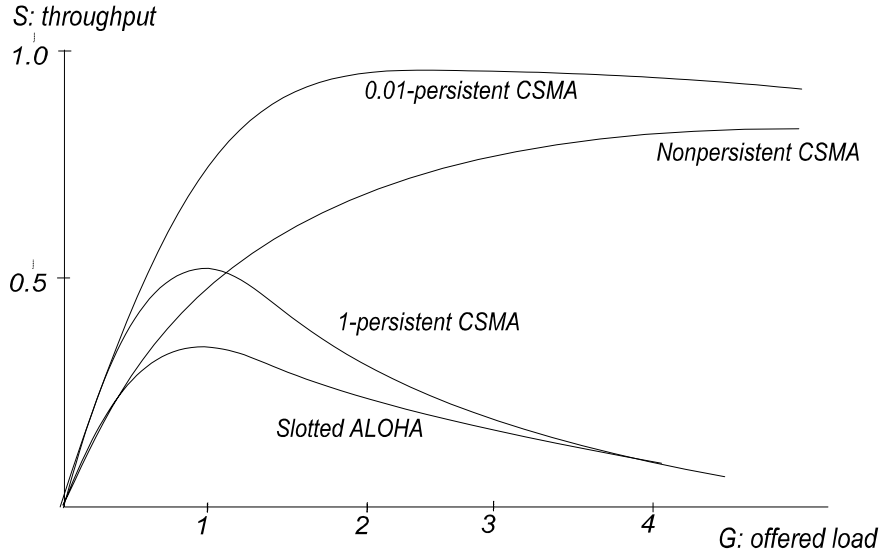


Figure 8.5 Comparison of CSMA strategies

In an effort to retain the efficiency of 1-persistence at low loading, and the performance of nonpersistence at high load, a third approach called *p-persistent* CSMA has been proposed. Unlike nonpersistence and 1-persistence, *p*-persistence is a slotted strategy with a slot length being set to the maximum propagation delay of the medium. A station wishing to transmit a packet, as is usual in CSMA, first listens to the medium. If the medium is busy, it waits for one slot and tries to begin again. If the medium is idle, the station will either transmit, with probability *p,* or wait for one slot, with probability 1- *p*. This is repeated with each idle slot until either the packet is sent or the medium is detected busy since some other station has begun to transmit. In the latter case, the station behaves as if there had been a collision and backs-off for a random interval before starting all over again. Clearly, like nonpersistent CSMA this is a very unselfish approach. It can be shown that if *p* is set very small (say *p*=0.01), the performance of *p*-persistence can exceed that of nonpersistence at all loads and

will even equal 1-persistence at low loading. However, *p*-persistence is little used in practice. It would require greater hardware complexity than the other forms, and, in any case, as we are about to see, a minor enhancement to CSMA yields such obvious benefits that it effectively renders the other strategies redundant. That enhancement is *collision detection*

CSMA with Collision Detect (CSMA/CD)

CSMA still behaves wastefully when a collision occurs, since each of the stations involved continues to transmit an entire packet, even though all such packets are doomed to be unsuccessful from the moment the collision starts. *CSMA with collision detect* (*CSMA/CD*) is a modification of CSMA wherein stations are equipped with hardware capable of detecting collisions as soon as they occur. A collision will only take place at the beginning of a packet transmission. If the collision is detected, the transmission can be aborted, and the channel released immediately, saving the bandwidth that would otherwise be wasted in sending the rest of the doomed packet.

Collision detection may be achieved in different ways. One approach involves comparing the signal that a station is trying to transmit with the signal actually present in the medium. Clearly, if there is no collision these should be the same, since the transmitting station will have sole access to the medium. If there is a collision, however, the signals will be different. A simple circuit to achieve this is shown in Figure 8.6, where the exclusive-OR gate is used to effect the comparison. A high on the *Collision(H)* line indicates to the station that a collision has occurred.

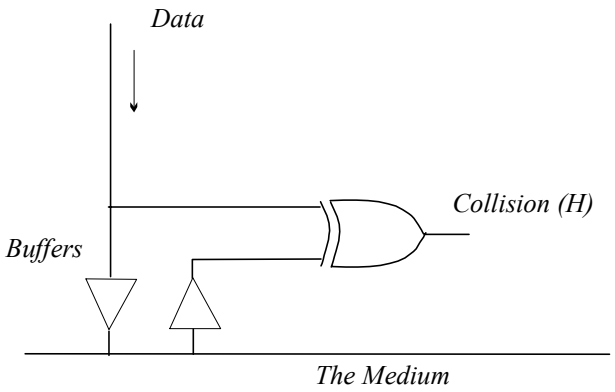


Figure 8.6 Collision detect circuitry for CSMA/CD

Another method relies on looking for signal amplitudes in the medium, greater than those used for actual transmission. These will occur when two high or two low signals superpose on each other. This is easy to implement, and will detect a collision if both signals have the same polarity, but will fail if there is so much attenuation in the medium that the superposition is within the amplitude bounds of a normal transmitter.

On detection of a collision, a transmitting station ceases transmission and sends a *jamming signal*, to ensure all stations are aware that a collision has occurred. The station then waits a random period (*back-off)* before trying again. Remember that in

standard CSMA, a station will not realise its transmission has failed until the expected acknowledgement fails to turn up, so this gives much earlier notice.

8B IN DEPTH: ANALYSIS OF CSMA/CD

We can analyse CSMA/CD, using the same techniques as for standard CSMA. Again, the idea is to find the average length of a transmission plus idle period, and divide that into the expected number of packets transmitted. We will only perform a rough calculation here, using the more easily analysed nonpersistent CSMA/CD, in order to obtain the bound of Equation 8.11.

There are two types of cycle, each beginning when at least one station starts to send a packet after an idle period. The first begins with a successful transmission, followed by another idle period. The average worst length of such a cycle is $1+\frac{1}{G}+\beta$, since it takes β in the worst case, for all stations to realise that an idle period has begun.

The second type of cycle begins with a collision followed by an idle period. From the beginning of the transmission, it takes a worst case of 2β, for all nodes to be aware of the collision. It takes another β for the collision to be revoked (we ignore the jamming signal time here). This type of cycle therefore has length $2\beta+\beta+\frac{1}{G}$.

The first type of cycle has probability $e^{-\beta G}$; the second has probability $1-e^{-\beta G}$. The worst case average cycle length, will therefore be:

$$\beta+\frac{1}{G}+e^{-\beta G}+2\beta(1-e^{-\beta G})$$

The expected number of packets transmitted is just $e^{-\beta G}$. It follows that:

$$S>\frac{e^{-\beta G}}{\beta+\frac{1}{G}+2\beta(1-e^{-\beta G})}=\frac{Ge^{-\beta G}}{\beta G+1+2\beta G(1-e^{-\beta G})}$$

If we find the maximum of the right hand side of this, Equation 8.11 is obtained. This gives a worst case maximum throughput for a CSMA/CD system.

There are a couple of points to note before moving on. Firstly, although this derivation is for the nonpersistent case, a very similar result holds for 1-persistence. Secondly, if the jamming signal duration is taken into account it will worsen the result slightly, since the collision length will be extended by this amount. However, jamming signals are only transmitted for sufficient time to ensure that all stations have noticed the collision, and the duration is therefore short (comparable with β).

CSMA/CD can be analysed by similar methods to those we have used above. Although an accurate analysis is rather involved, it is not difficult to show that the maximum throughput is at least as good as:

$$S_{\max} > \frac{1}{1 + 6.2\beta} \qquad 8.11$$

for a medium with end-end propagation delay β (in packet times). We can express β, in terms of τ, the maximum propagation delay across the network in seconds, L the packet length and R, the transmission rate. Thus:

$$\beta = \frac{\tau R}{L} \qquad 8.12$$

If τ is small then CSMA/CD can achieve maximum throughput getting close to 100%.

A station that has begun to transmit can be sure it has gained the medium after a maximum of 2τ seconds. After this point, the transmitter can be confident there will be no collision involving its packet. After a successful transmission, the network enters a *contention phase*, which may involve several collisions and retries. CSMA/CD can be combined with nonpersistence or 1-persistence, if a station wishing to transmit finds the medium busy. The 1-persistent strategy is much more effective than in standard CSMA, because, although it makes collisions more likely, the penalty for each is greatly reduced. 1-persistent CSMA/CD is in fact the MAC approach used in *Ethernet,* which we shall look at in greater detail later.

8.1.2 Collision Avoidance MAC

Unlike MAC strategies which rely on resolving collisions when they occur, either by aborting transmission, or repeating damaged packets, collision avoidance aims to prevent collisions occurring in the first place. Methods in this class rely on contending stations working out who has permission to send, prior to actually transmitting. There are a number of common implementations, the most important of which we now discuss.

Polling

One of the oldest and most primitive avoidance techniques is *polling*. Unlike the collision resolution systems, polling requires that one of the stations, the *primary*, plays the roll of a central controller, initiating the polling sequence, while the other stations, the *secondaries*, are asked in some predefined *polling sequence* whether or not they have a packet to transmit.

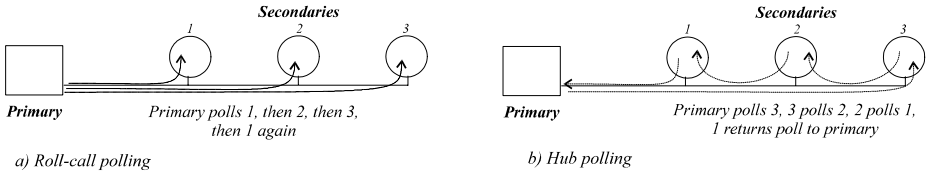


Figure 8.7 Polling

In the simplest form, the primary explicitly polls each secondary in turn, using a special *poll packet* or *signal* directed to that secondary. A secondary station wishing to transmit may do so when polled, although it is often permitted only to send directly to the primary. The technique is called *roll-call polling* and is straightforward but

inefficient and can be slow (Figure 8.7a). A station that wishes to transmit must wait until it is polled, even if no other station wants to use the medium. Also a poll operation needs two propagation delays since the poll request must be sent to the secondary and its response returned.

An enhancement is *hub polling* (Figure 8.7b). Here, the primary polls the farthest secondary. If this does not wish to transmit, or when it finishes, it returns the poll to the next secondary along, and so on. This is more complex than the roll-call method, in that a secondary must be able to pass the poll request on to its neighbour. It is also more efficient, since a poll no longer needs to propagate across the medium twice, but it still requires central control. Also, in roll-call polling, if the primary itself wants to send a packet, it has an opportunity to do so after each secondary is finished with the medium, whereas in hub polling, it gets an opportunity only each time all secondaries are finished. If the primary needs more access than this, a second data path must be included for primary-to-secondary transfers. This can either be done by splitting a single medium using, for example, FDM, or by introducing a second physical pathway.

Polling is useful in asymmetrical situations such as where a number of terminals are connected to a controller by a single multidrop link. In this configuration a protocol like HDLC-NRM might be used, with the controller acting as primary and the terminals as the secondaries. In this HDLC mode a primary employs roll-call polling to establish communication with the secondaries (see Section **Error! Reference source not found.**).

Token Passing

Hub polling, by definition, relies on the existence of a special primary station. This is not an appropriate configuration in a situation where no one station is inherently different to any other. However, the hub polling idea can be extended to remove the primary completely. The stations are assigned positions in a logical circular sequence, which may or may not correspond to their physical positions, and a control packet or *token* controls access to the medium (see Figure 8.8). A station that has possession of the token may transmit a packet, or, in some cases, require another station to transmit a packet. Packet sizes may be fixed or variable, and there will be some rule as to how long a station can retain possession of the token before passing it on to the next in sequence.

Token passing avoids the requirement for a central polling controller, but one station is required to take responsibility for token management such as regenerating the token if it is lost. To avoid dependence on any particular station, all must be capable of assuming this management role, and the station that actually does so is often simply the first on-line during a particular network session.

The token passing system is particularly simple conceptually, although the actual hardware involved in implementation tends to be considerably more complex than that required for contention systems like CSMA/CD, particularly in view of the requirement that any station be able to become the token manager. Its main practical drawback, which it shares with hub polling, is that, at light loading, a token may need to be passed around many idle stations before reaching one that is waiting to send. We

can compute an average time, γ, for transfer of the token from one station to the next as:

$$\gamma = \bar{\tau} + \frac{T}{R} + r \qquad 8.13$$

where $\bar{\tau}$ is the average propagation delay between two stations, T is the length of the token in bits, R, the transmission rate, and r the time taken for a token to be detected. Obviously $\bar{\tau}$ depends not only on the physical spacing of the stations, but also the order in which they are placed in the logical token-passing sequence. In some stations this is the same as the physical order, but in others stations can be rearranged to make different sequences.

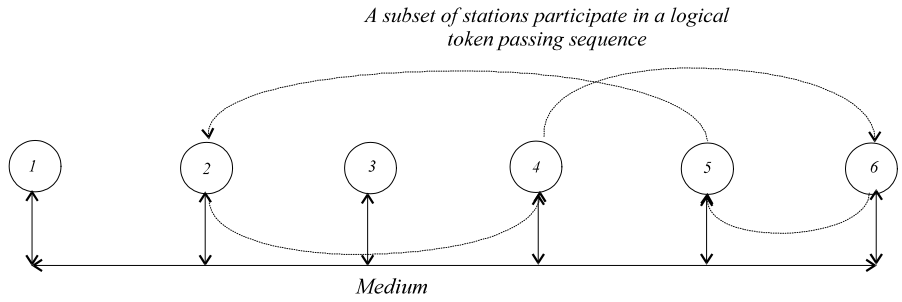


Figure 8.8 Token passing

One problem with token passing occurs when a new station wishes to join the logical sequence. Some mechanism must be included to allow this, for example by ensuring that, periodically, stations within the ring invite others to join (this mechanism is used in the *IEEE 802.4 token bus* LAN). Of course a token passing network may include stations which cannot participate in the token exchange and can then only respond to, but not initiate, communication.

CSMA/CA

One of the problems with token passing is the length of the token itself, and consequently, the time it takes to transmit (the second term in Equation 8.13). In certain circumstances this can be reduced by using an "implicit token" in the form of the channel idle period between transmissions. Basically this works as follows. Stations are arranged in a logical ring, just as in token passing, but now a station waits for a break in transmission of its predecessor to determine its own right to transmit.

Once a station, say number k in the sequence, is finished transmitting, there will be a delay τ, equal to the maximum propagation delay across the medium, before all other stations, including k+1, have seen the transmission stop. If station k+1 wishes to transmit it may now do so. However, if it has not transmitted in some fixed time r, it relinquishes its right. This latter time is set to give the station a chance to respond to the detection of the idle medium, should it so wish. If station k+1 does not wish to transmit, the medium will still be idle, as seen at station k+1 itself, after $\tau + r$. The idle state will be visible with certainty across the entire medium after another τ, and so on.

If subsequent stations also decline to transmit, each must wait an additional $\tau + r$ for its turn. The transfer time between stations is thus:

$$\gamma = \tau + r$$

replacing Equation 8.13 for token passing. Note that the propagation delay here is the maximum across the entire network, as opposed to the average delay between stations. The technique requires a station to listen to the medium and is therefore considered to be a variant of CSMA, known as *CSMA/CA* (*CSMA with collision avoidance*).

Reservation Protocols

All the avoidance algorithms discussed so far work by letting contending stations determine who sends next, permitting the winner to transmit, and then entering another decision phase. In a *reservation* MAC protocol, advance decisions are made for stations that wish to transmit, during a preliminary *reservation phase*. All stations that have acquired a right to transmit during the reservation phase then do so, in precisely controlled order. Once this is complete a new reservation phase is entered.

There are several different implementations of the reservation concept, but typically the reservation phase consists of a series of short *minislots*, which stations can use to reserve a right to transmit. One way of doing this is to provide a 1-bit minislot for every station. If station *k* wants to transmit a packet it simply sends a "1" during its minislot. At the end of the reservation phase, those stations that have booked in this way will each send a packet, in the same order as their minislots (Figure 8.9).

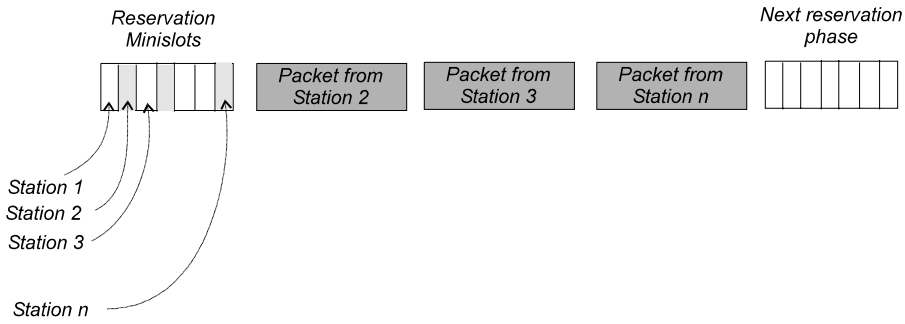


Figure 8.9 Bit-map reservation protocol

This requires a reservation minislot for every station, which, in a large network, may become quite an overhead. A variation limits the number of reservation minislots, does not allocate any one to a particular station, and uses a contention protocol like slotted ALOHA to decide between competitors during the reservation phase. Each station that succeeds in acquiring a minislot then transmits a packet, in the appropriate order, once the reservation phase is over. Obviously the use of a single bit for a minislot is no longer adequate, since a successful station must identify itself explicitly to the others. Strictly, this is not a collision avoidance algorithm, however, since there is contention for minislots.

8.1.3 Pipelined Media and Rings

In a typical broadcast network, when one station sends a message it propagates across the entire medium at a speed characteristic of the signal. After a certain propagation delay, any other attached station can detect the message. There are, however, some situations, where attenuation in the medium is so great that distant stations cannot detect the signal—as is the case, for example, in some wireless networks and in situations where cables are very long. In networks relying on guided signals (cable, optic fibres etc) *repeaters* can be helpful in extending the range. As described in Section 3.4, a repeater is basically a device that regenerates guided signals (Figure 8.10). To do this it needs little more than a line receiver, a circuit to clean up digital levels (low, high etc), and a line driver to retransmit the signal on the next leg of its journey. Sometimes the repeater may include a store one or two bits, which can not only facilitate the regeneration of a new bit stream, but also gives the repeater control over the exact timing of the retransmission. Simple repeaters do not normally analyse bit-stream passing through and do not perform switching or routing, but they do fundamentally separate the medium physically into two disjoint sections, which rely on the repeater to relay signals from one to the other. Fundamentally, the repeater function is unidirectional, at any given time, although it is not difficult to design half-duplex repeaters, or indeed fully bidirectional models that repeat FDM signals.

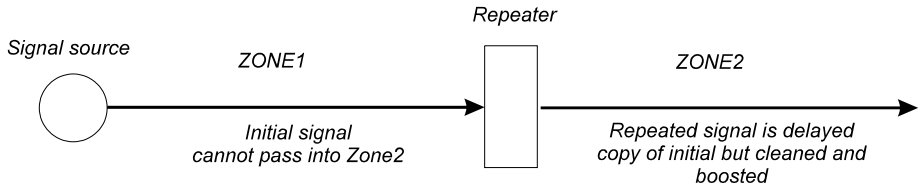


Figure 8.10 Action of a simple repeater

While simple repeaters are useful ways of extending cable- or fibre-based broadcast networks, augmented repeaters can also be used to allow stations to access a guided medium in a manner somewhat different to the simple signal broadcast model. To see how this works, consider the common example of the serial cable bus encountered in many LANs, where a signal injected by one station propagates in both directions along the cable, until it is received by all (Figure 8.11). If two signals are transmitted onto the cable at the same time they will superpose, and, if they are not frequency-division multiplexed, a collision will result.

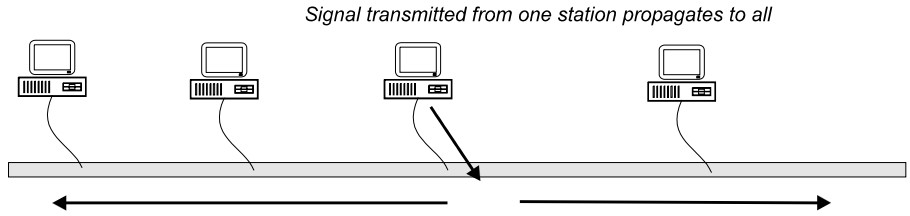


Figure 8.11 Serial cable bus without repeaters

However, suppose now that the cable is divided into multiple segments by clocked unidirectional repeater/switches, each of which also provides switched access to a single station (Figure 8.12). Without loss of generality, suppose signals move along the cable from left to right. The repeater/switch performs three functions.

a) The *repeat function* retransmits on the cable to the right, each bit received from the cable on the left, with a small delay, which should, ideally, be one bit time.

c) The *listen function* copies each bit received from the cable along the access line to the attached station, so that the station sees incoming data.

b) The *transmit function* is used only when a repeater's station has data to transmit, whereupon the repeater switches data from its own station onto the cable.

Obviously the repeater cannot repeat incoming bits in the same band as it transmits bits from its local station, so unless the system uses FDM it cannot repeat and transmit at the same time. This, of course, is just what MAC procedures are about, and some medium access algorithm is required to ensure that transmit and repeat functions are performed in an orderly way.

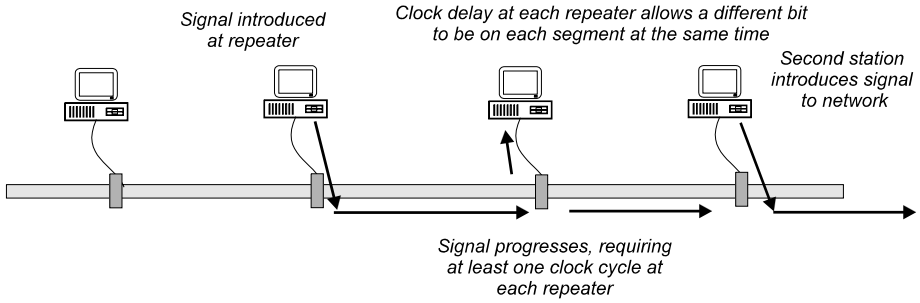


Figure 8.12 Pipelining signal transmission on a cable

The key point about this arrangement is that, if the repeaters transmit at the same rate, different bits can be sent on different segments at the same time. The hardware does have to be arranged carefully to ensure that this works, but the result is a kind of pipelining effect. For example, if each repeater can store one bit, a given repeater can be passing on bit k of some packet, while the immediate predecessor to its left is passing on bit k+1.

What are the advantages of the pipelined over the non-pipelined serial cable bus? Primarily, the medium now consists of a series of point-to-point links, and, in both electrical and optical systems, these are easier to engineer than multidrop connections, especially where very high baud rates are to be used. As we saw in Section 2.8, electrical digital signals can be distorted by *reflections* off badly terminated lines, stubs (which are short offshoots from a main cable), or indeed any physical discontinuities in line impedance. To reduce this effect, multidrop LANs must be terminated at each end of the cable, and, if lines between repeaters are split in any way, this must be done very carefully, by, for example, keeping stubs as short as possible. These factors place quite stringent constraints on the design of serial cable bus systems. Similar difficulties arise with optics where point-to-point transmission is

much better developed than technologies that permit one-to-many physical signalling. In essence, any system using one-to-many signalling must take account of the fact that each physical receiver drains some energy from the signal passing it, as well as reflecting some back in the wrong direction. If these effects are severe enough, signals in a multidrop medium will become so attenuated and distorted as to be impossible to detect correctly. A second advantage is the obvious one that the repeaters give regular regenerative boosts to signals, making transmission over distances at speed more reliable.

However, there are drawbacks as well. Firstly, since repeaters are unidirectional, there is no immediate mechanism to allow stations to transmit to those upstream (all signals move from left to right). One way round this is to use two cables, with one carrying bits left to right, the other right to left. Another solution is to connect the cable and repeaters into a *ring*. In fact, this is the approach most commonly adopted in those local area networks that use pipelined cable, and ring topologies have been popular for many years in this arena. In fact, as we shall see shortly, networks based on serial buses, and those based on rings are, traditionally, the two great contending classes of LAN—although more recently switched network technology has also made an appearance in this market. Figure 8.13 shows the layout of a typical unidirectional ring. It is perfectly possible, of course, to combine the dual cable and ring solutions to create a bidirectional ring as well.

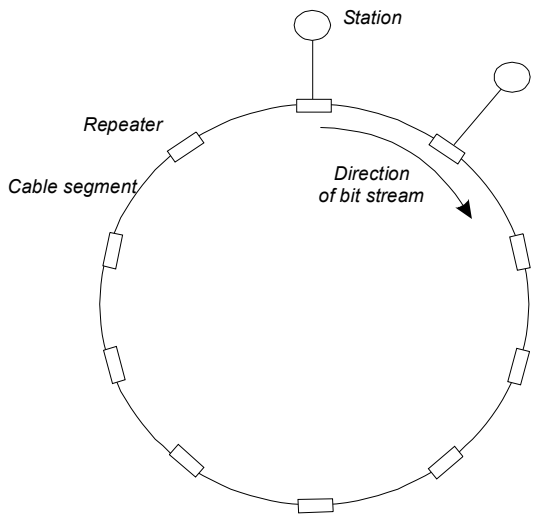


Figure 8.13 The unidirectional ring topology

In summary, then, a (unidirectional) ring network consists of *repeaters* connected by point-to-point links to form a closed loop, with bits circulating continuously in one direction. The repeaters are active devices that regenerate bits but also perform the medium access function, and always introduce a delay of at least one bit per repeater. The ring behaves in many ways like a large constantly circulating clocked circular

shift register, and an important characteristic for any ring is the number of bits that can be circulating at one time. To evaluate this, note first of all that the ring must have a clock rate—all bits at all repeaters are clocked steadily at this rate, so that as one bit arrives at a repeater, another leaves in synchronism. At any instant, some bits on the ring are held temporarily at repeaters, the number involved depending on how many bit delays each repeater introduces (and hence how many bits a repeater can hold). Other bits, however, are actually being transmitted from one repeater to another, and in fact, some of these may exist at a given instant, only as a moving signal on the cable.

For example, suppose a ring has a clock rate of 10MHz, so, assuming a bit stream is passing through it, any repeater must transmit a new bit every 100μs. If the signal travels at a typical speed of 2 x 10^8 m/s (0.67 of the speed of light), the wavefront representing the start of a new bit will travel $\frac{2\times10^8}{10^7}$, or 20 metres, before the transmitter gets its next clock edge and begins to send the next bit. If a link is, say 100m long, five bits can be in flight along it at one time. These bits will have left one repeater, but not yet arrived at the next. The number of bits that can be fitted onto a given link is called the *bit length* of the link, and clearly depends on the actual length, the clock rate and the signal speed.

Ring links may be twisted pair, coaxial or optic fibre. As it is quite difficult to implement a multidrop optic fibre link most optical LAN implementations are rings. A frame or packet structure is imposed on the bits streaming round the ring, with each frame carrying the destination address of the station at which it is targeted. In most rings the only efficient way to carry out many protocol functions on such a frame is to build a capability directly into the repeaters. Therefore, in most ring implementations repeaters can identify frames and may modify certain control bits before passing them on.

Physically, pipelined cable systems, including rings, are not really broadcast media, because it is not possible for a sender to transmit a signal across the whole medium without it being relayed by what are, effectively, switches. However, to all intents and purposes this distinction is irrelevant, because the repeaters operate at a bit level, and frames transmitted by a station are immediately repeated across the whole network. Looking at things from a link-level point of view, therefore, a ring is a broadcast medium in just the same way as a bus is. Of course, because medium access is so tied in to the hardware, the algorithms used in rings are a little different to those we have encountered so far. In one approach, called the *slotted ring*, bits circulate round the ring at all times, whether or not there is data to transport. The ring is viewed as a series of *minislots*, which have a frame structure, and can be marked empty or full. A station wishing to transmit, must find an empty slot, marking it full, and fill it with data, all in synchronism with the ring clock. Other rings use the more familiar token passing principle to govern access.

8.2 LOCAL AREA NETWORKS

8.2.1 Introduction and Standards

The most important application of broadcast technology is in local area networks. LANs are typically privately owned and administered. Because they operate over short distances they can use customised high-performance links, and, consequently, compared to wide area networks, they have high bit rates and low error rates. Probably the commonest and least exacting motivation for the introduction of a LAN is the sharing of communal resources, such as printers or disks, between many personal computers. In this environment, the communications system has to be simple enough to function when hosted by a single-process personal operating system such as *Microsoft's MS-DOS* or *Apple's MacOS*. The network organisation is typically, although not invariably, client-server, and allows some machines, file servers, to export their file systems, which can then be mounted as extensions to the local file systems of their clients. Other organisational environments require more than just resource sharing, and must have a communications system able to deliver inter-process messaging and to support true distributed applications. Still others require high-speed connections for a limited number of powerful devices, often including a high-end mainframe or supercomputer.

Initially, LAN technologies were developed by individual manufacturers, but it soon became apparent that a standardisation process would be beneficial not only to users but also to equipment and software suppliers. The first really important standards work was *IEEE 802*, which attempted to take the most popular types of LAN, and lay down precise definitions of the physical technologies and MAC mechanisms (which are inextricably intertwined). It also attempted to provide a framework for how these could fit into a more general subnet access model. In fact, IEEE 802 was very successful, and was adopted pretty much in its original form as *ISO 8802*. It attempts to bring LAN technologies into the OSI framework, and, as we saw in Section 6.2, the variety of these technologies and associated MAC methods is accommodated by splitting the data-link layer into two sublayers, MAC and LLC.

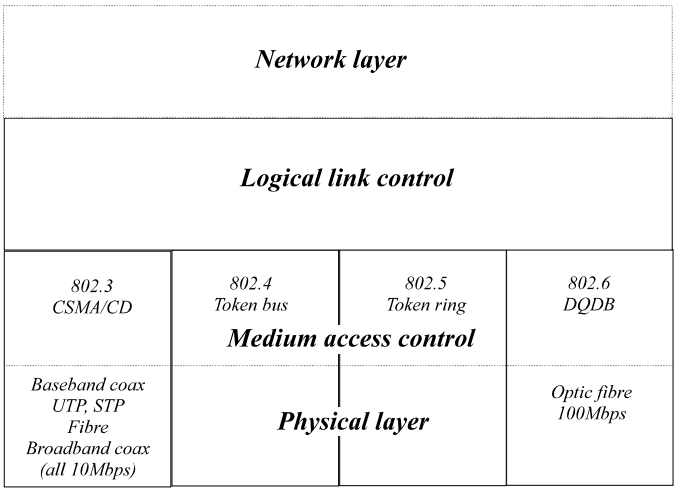


Figure 8.14 Structure of IEEE 802

The standard is actually divided into a number of documents as follows (see also Figure 8.14). The first, *IEEE 802.1* is an "Introduction and Service Definition", while *IEEE 802.2* defines the *logic link control* or LLC protocol. IEEE 802.3, 802.4 and 802.5, standardise MAC for what IEEE saw as the three most important LAN technologies, respectively the CSMA/CD serial bus (effectively Ethernet), *the token bus*, and the *token ring*. Of these three choices, it would be difficult to argue with the importance of Ethernet and even token ring, but token bus has proved less popular, largely because its attempts to fully decentralise token management have made it very complex. It is also notable that other technologies such as the slotted ring were not included. IEEE 802 did, however, also specify a standard for MANs, namely the *distributed queue dual bus* (*DQDB*) defined in *IEEE 802.6*. Other documents have been added since, covering areas such as wireless LANs (*IEEE 802.11*).

Although IEEE 802 is structured formally in the style of OSI, the basic technologies and protocols it encompasses are widely used in other architectures as well. The MAC sublayer is highly technology-dependent, but, for reasons that will become apparent shortly, provides only a semi-reliable connectionless service to higher levels. In the following sections we will first examine the major LAN technologies, then discuss briefly how these are incorporated into communications architectures.

8.2.2 Ethernet.

The first Ethernet was developed at Xerox in 1976 by Metcalfe and Boggs. It was based on a 3Mbps serial cable bus, called the *Ether*, and used 1-persistent CSMA/CD for medium access. In 1980 a consortium consisting of Xerox, DEC and Intel produced the *Ethernet Specification*, an updated version whose primary goal was to stabilise the standard at a higher speed of 10Mbps. The Ethernet Specification, defined what was called *Ethernet II*, and was used as the basis for IEEE 802.3, which is actually a generalisation covering a family of similar systems. In fact there are differences between the MAC protocol used in Ethernet II and IEEE 802.3. IEEE have

provided a mechanism to allow for backward compatibility in the form of the *SNAP* extension to LLC, discussed briefly at the end of Section 8.2.5.

Strictly "Ethernet" refers to a specific product, which predated the general standard 802.3. However, conventional usage has blurred this distinction, and any 802.3 complaint network is now commonly referred to as an "Ethernet". In fact the term, "Ethernet", is now also used to describe any network which uses a MAC sublayer similar to that specified in 802.3, even when other physical media or enhancements to the MAC protocol have been introduced. In the original Ethernet the broadcast medium was coaxial cable, forming a single-channel baseband serial bus operating at 3Mbps or 10Mbps. 802.3 also permits the use of twisted-pair and optical fibre "Ethernets", while the evolution of contention buses in the market-place has seen the introduction of "fast Ethernets", capable of 100Mbps operation, and "switched Ethernets" based on switches rather than conventional broadcast media,

Coaxial cable is still used for Ethernet implementations, although, as we shall see, it is increasingly being replaced with twisted pair technology. Two types of cable are employed: the 10mm *10Base5*, known as *thick Ethernet*, which allows up to 100 nodes over a maximum length of 500m (in a single segment); and the 5mm *10Base2*, or *thin Ethernet*, which can support 30 nodes over 185m. The two technologies differ significantly in the way that stations are connected to them. The thick Ethernet cable is expected only to pass close to stations, and need not attach to them directly. Each station is linked by a *drop-cable* that can be up to 50m long, to an associated *attachment unit interface* or *AUI* (Figure 8.15). This consists of a so-called *vampire tap*, which uses a metal screw to establish direct electrical contact with the core of the Ether, and an adjacent *transceiver*, which performs carrier and collision detection as well as transmission and reception of signals. The drop cable carries five twisted pairs: *data*, *output enable* and *power* from station to transceiver; *data* and *collision indication* from transceiver to station. Thin Ethernet cable, on the other hand, is more flexible, and is intended to attach directly to the host station using standard *BNC T-piece* connectors. In this case, the transceiver electronics are in the station itself.

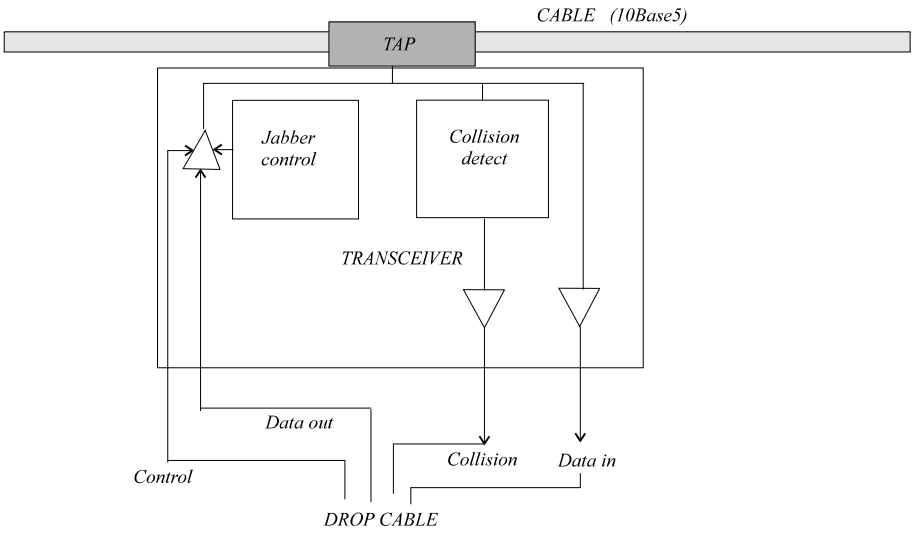


Figure 8.15 Connecting to 10Base5 Ethernet

A station or transceiver that develops a fault and will not stop transmitting can bring down the whole network, so all transceivers include a *jabber control* function, which disables the data output if, for example, the stipulated maximum frame length is exceeded. The specifications for thick and thin Ethernet include further topological constraints. For example, it is allowable to extend a thick Ethernet by using repeaters to connect multiple segments, so long as, firstly, there is only one path between any two stations, and, secondly, each such path is no more than 2.5km long with at most four repeaters.

More recently a new Ethernet configuration has become popular. This is *10Base-T,* which replaces the coaxial cable bus with a backplane-based box called a *hub* to which each station is attached via two twisted pair data cables, one for each direction (Figure 8.16). The hub collapses the main functionality of the Ether into a single small unit, and all long-distance signal propagation is over the point-to-point twisted pair links. The hub repeats any signal received on an input to all its outputs, emulating the broadcast feature of the cable bus. A hub can detect a collision very simply by watching for simultaneous activity on more than one of its inputs, and can then warn all stations by sending them a special *collision presence* signal. However, as in thin Ethernet, the jabber control and other interface functions are located in the station itself. A hub is usually organised as a backplane bus into which a number of interface cards are inserted, each containing all the circuitry necessary to handle the hub end of one or more twisted pair links (see Figure 8.16).

Hubs have several advantages over extended cable buses. Firstly, they allow for centralised monitoring, maintenance and upgrading of the network infrastructure. Secondly, they can use Category 3 (or better) unshielded twisted pair (UTP) wiring already present in many modern office buildings for telephone distribution. On the debit side, UTP runs must usually be less than 100m in length (150m with Category 5 UTP), which is somewhat more restrictive than in thick Ethernet. Physically speaking,

of course, a hub configuration is a *star* rather than a bus topology, but from the MAC point of view this makes no real difference, as the hub is also using CSMA/CD. Like the cable it replaces, the total bandwidth available to all stations in a hub is 10Mbps.

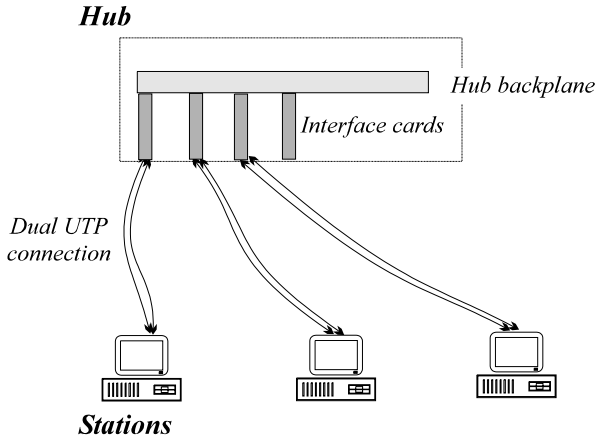


Figure 8.16 Typical 10Base-T configuration

However, the performance of Ethernet hubs can be enhanced if their design is modified as follows. Instead of relying on a single shared backplane, multiple paths can be introduced through the device, allowing more than one transfer to occur simultaneously, so long as to different destinations. This breaks away from the basic CSMA/CD principle, and effectively turns the LAN into a switched network. However, the enhanced switching hub is interfaced to the stations by the same UTP connections as 10Base-T, and therefore, to these stations, looks just like a 10Base-T hub, but with superior performance. "Collisions" of a sort are still possible if, for example, two stations try to transmit to the same target or, in some switches, if two stations that are attached to the same interface card try to send at once. Such a hub is called an *Ethernet switch*. Obviously the total bandwidth provided by an Ethernet switch must be much greater than that of a hub, and capabilities are often over 1Gbit/sec. The cost of a switch is unsurprisingly, therefore, correspondingly higher. With a switch, it also becomes possible to use the two twisted pairs connecting to the station as full duplex 10Mbps connections. It is also possible to use the switch to connect segments of 10Base-T Ethernet together, each based on its own hub. The switch then routes between these segments.

A "collision" in a switch is really just a resource conflict, which different switches can deal with in different ways. If internal buffering is provided, some collisions can be deflected, just by storing all but one of the conflicting packets temporarily. Of course this solution only defers the point where, ultimately, the switch runs out of buffer space. When this happens, it can either throw away new packets, relying on upper layers for recovery, or it can give up its collision avoidance efforts, and simply signal a collision back to the sending station. It may well strike you that this Ethernet switching bears little resemblance to the original CSMA/CD concept and has, in fact, turned Ethernet into a switched network, relying on standard switching principles.

This impression is quite correct. However, the network is still considered to be an "Ethernet", because it behaves, from the point of view of the attached stations, just like an Ethernet, although its performance is superior. Full duplex 10Mbps switched Ethernet is not part of the original Ethernet or even 802.3, but an unofficial standard, *full duplex switched Ethernet* or *FDSE*, has been developed which is supported by a number of manufacturers.

In any of the implementations, 10Base5, 10Base2 or 10Base-T, the functions required at a station can nowadays easily be combined onto a small number of integrated circuits. These are often mounted on an *Ethernet interface card*, which can plug into a standard bus on a PC or printer, and can often offer support for two, or even all three, interface types. In the case of 10Base2, the card connects directly to the thin cable; in 10Base5, to the station's AUI transceiver, via a drop cable; and in 10Base-T, to a hub or switch interface card via Category 3, or better, UTP.

The 802.3 standard also allows for fibre optic and broadband implementations. The fibre optic specifications are collectively known as *10Base-F*, although there are in fact three alternatives. *10Base-FL* and *10Base-FB* are point-to-point links for connecting repeaters to each other or to individual stations. *10Base-FP*, on the other hand, defines a *passive star* topology using a passive optical device called a *star coupler* to which up to 33 stations can be attached using optic fibres. Fibre technology can also be used to connect 10Base-T stations to hubs over distances longer than possible with UTP. The broadband specification, *10Broad36*, allows a data rate no higher than baseband Ethernet, but, by transmitting over 75Ω CATV coax, using DPSK, segment lengths of 3600m are permissible.

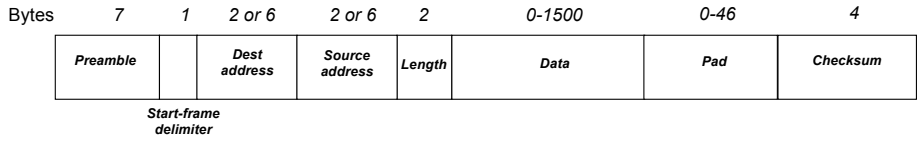


Figure 8.17 IEEE 802.3 frame structure

Ethernet and 802.3 use Manchester encoding to ensure that clocking information is transmitted along with the data. The electrical variants use voltage levels of ±0.85V. The frame (MAC PDU) structure, shown in Figure 8.17, is really very simple, with the following fields defined.

- The *Preamble* consists of seven bytes all of the form 10101010, and is used by the receiver to allow it to establish bit synchronisation (recall that there is no clocking information on the Ether when nothing is being sent).
- The *Start frame delimiter* is a single byte, 10101011, which is a frame flag, indicating the start of a frame.
- The addresses used in baseband 802.3 are always 48 bits long, although older versions of Ethernet used 16 bits. Individual addresses have a most significant bit of 0, multicast addresses a most significant bit of 1. An address of 48 "1s" is a broadcast to all stations on the local network. An interesting feature is that individual addresses may be local or global, with, respectively, a second most significant bit of 0 or 1. Local addresses have no significance except on the local

Ethernet installation, but global addresses are unique: every system with an Ethernet interface has a unique global address hardwired into that interface. In principle, any station can address any other, anywhere in the world, but of course this would rely on a rather impractical internet layer to identify the target, and perform the routing. On a single Ethernet there is no such problem because a frame is seen by all stations and each can recognise its own address.

- The *Length* field indicates the number of bytes of data in the frame's payload, and can be anything from 0 to 1500 bytes. Interestingly in 803.2 frames must be at least 64 bytes long, not including the preamble, so, if the data field is shorter than 46 bytes, it must be compensated by the *Pad* field. The reason for specifying a minimum length lies with the collision-detect mechanism. In CSMA/CD a station must never be allowed to believe it has transmitted a frame successfully if that frame has, in fact, experienced a collision. In the worst case it takes twice the maximum propagation delay across the network before a station can be sure that a transmission has been successful. If a station sends a really short frame, it may actually finish sending and release the Ether without realising that a collision has occurred. The 802.3 design rules specify an upper limit on the maximum propagation delay in any Ethernet installation, and the minimum frame size is set to be more than twice this figure (64 bytes takes 51.2μs to send at 10Mbps).
- Finally the *Checksum* field uses a CRC-32 polynomial code as discussed in Section 6.4.3.

To send a frame, a station on an 802.3 network first listens to the Ether (carrier sense function). If the Ether is busy, the station defers, but, after the current activity stops, it uses a 1-persistent strategy and will wait only for a short, fixed delay, the *inter-frame gap*, before beginning to transmit. If there is no collision, the transmission will complete successfully. If, however, a collision is detected, the frame transmission stops and the station begins to send a *jamming signal* to make sure that all other stations realise what has happened. The station then backs off for a random time interval before trying again. The back-off interval is computed using an algorithm called *truncated binary exponential backoff*, which works as follows.

The station always waits for some multiple of a 51.2μs time interval, known as a *slot*. The station chooses a number randomly from the set {0,1} and waits for that number of slots. If there is another collision it waits again, but this time for a number chosen from {0,1,2,3}. After k collisions on the same transmission it chooses its number randomly from $\{0, \ldots, 2^k\text{-}1\}$, until $k = 10$, when the set is frozen. After $k = 16$, the so-called *attempt limit*, the MAC unit gives up and reports a failure to the layer above.

The collision does, however, typically result in a short malformed packet called a *runt*. The runt is just the part of a packet that the first sender managed to transmit before the collision occurred. If the Ethernet construction rules are adhered to, the runt will be easily identified because it will be shorter than the Ethernet minimum (64 bytes). Its CRC will also be incorrect. Runts are common and cause little difficulty on individual Ethernet segments, but it is important that they should be identified and filtered by a switch to prevent them being forwarded unnecessarily onto other segments.

8.2.3 Rings

Without question, Ethernet is the dominant LAN technology in use today, and, in the typical spirit of market-driven innovation, it has, somewhat like the IBM PC architecture, exhibited an ability to evolve and adapt, beyond the predictions of many who have criticised its shortcomings. Once the market accepts a technology completely, it will often ingeniously circumvent inadequacies by a very rapid evolutionary process that can leave inherently superior competitors sidelined. This is a phenomenon that should be borne in mind by any aspiring prophets of computer technology: the best alternative on paper frequently loses out to the one that delivers what the customers want quickly and cheaply. Recent developments in Ethernet such as the switching discussed above and the 100Mbit/sec *Fast Ethernet* we will examine in Section 8.3.5, have greatly enhanced the prospects of the technology, bringing it to the point where, in some LAN situations, it can now compete with high-end solutions like ATM.

However, from the beginning CSMA/CD attracted two important criticisms. The first was its indeterminacy, in the sense that even when a packet has begun transmission, there is no guarantee as to how long it will take to reach its destination. This, as we have seen, can be problematical for isochronous data such as voice and video. The second criticism concerns the lack of packet priorities in Ethernet. This is fundamental, because, of course, all Ethernet frames are treated identically in the Ether, and it is not possible to give one an advantage in the ability to avoid collisions for example. For these reasons some people felt that alternative MAC systems should be developed for use in LANs.

Rings, in particular, were favoured by many as they avoid the electrical problems of buses, make the use of fibre optics easier, and do not have the collision indeterminacies associated with Ethernet. A number of MAC algorithms for ring access were developed, but the one which proved most widely successful was the *token ring* adopted initially by IBM, and standardised as 802.5.

The ring structure consists of unidirectional point-to-point links connecting active repeaters, which can be placed into a relay-operated *bypass mode* if a station is damaged or inactive. As many as 250 repeaters are allowed, each attached to a station via a drop cable, although, physically, groups of these repeaters may be amalgamated into a single *concentrator* (to make it convenient to attach a cluster of machines to the main ring route). Every repeater has a hardware address that conforms to the same formatting as those used in 802.3. Links may be shielded or unshielded twisted pair and rings may be slow (4Mbits/sec) or fast (16Mbits/sec). The line code used is differential Manchester, and access to the ring is, of course, via token passing.

The 802.5 standard requires only two frame formats: *data frames*, which may contain actual data or MAC information, and *tokens* (Figure 8.18a and b respectively) A station which has acquired a token may transmit a data frame, which then passes round the ring to the target destination. Repeaters can interpret the frame structure, and each one which is not bypassed repeats frames to its output link with a delay of one bit, examining the destination address field for a match with its own address. When a match is found, or a broadcast address is recognised, the frame contents are copied along the drop cable to the station. However, the frame continues round the ring back to its source, where it is removed. As the destination repeater passes on the

frame, it sets acknowledgement bits in the frame status (FS) byte. The format of the FS byte is shown in Figure 8.19. Note that the two halves of the byte are identical. This is because it is placed after the frame check sequence in the frame structure and, as it is not protected by the latter from corruption, uses repetition as a safeguard. Only two bits are of significance: *address recognised* (*A*) and *frame copied* (C). The receiving repeater sets the former when it recognises its own address and the latter when the frame is copied to the destination station. A full acknowledgement requires *A*=1 and *C*=1. Note how the natural closure of the ring assists the acknowledgement process.

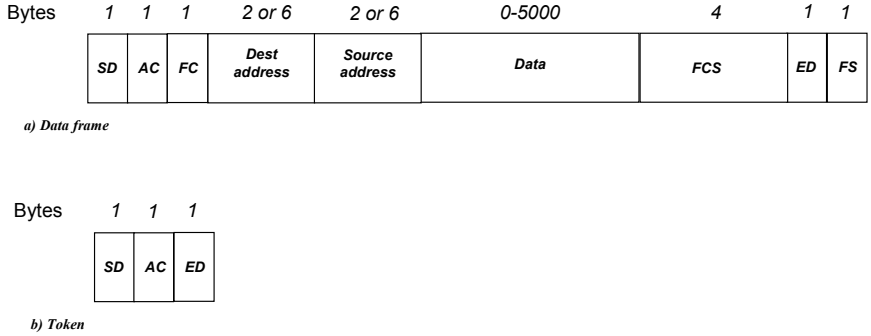


Figure 8.18 Frame and token formats in token ring

Each frame begins with a *start delimiter* (*SD*), which, as shown in Figure 8.19, includes signal elements, J (high for one bit time) and K (low for one bit time), not used in ordinary Manchester encoding The *end delimiter* (*ED*) also includes these elements, but in a different configuration. It is therefore easy for repeaters to identify frame limits. The end delimiter additionally includes an *error bit* (*E*), which the destination sets if it detects a CRC error, and an *intermediate bit* (*I*) which is used to indicate whether a frame is the last in a sequence (in which case it is reset).

The *access control (AC*) byte includes a three-bit *priority* and a three-bit *reservation* field, which are used for operating the frame priority system, discussed further later in this section. The *monitor bit* (*M*) is used to prevent a frame circulating repeatedly around the ring if the transmitter fails to remove it. The *token bit* (*T*) simply differentiates a data frame (*T*=1) from a token (*T*=0).

As for the other control bytes, the *frame control* (*FC*) differentiates true data frames from those used for MAC purposes—the data field can carry true (higher level) data, or MAC information, with a maximum size determined by the longest time a station is allowed to hold the token. For a 4Mbps ring and a typical maximum token holding time of 10ms, the size limit is about 5000 bytes. Finally, the *FCS* is just a 32-bit CRC as in 802.3.

A token frame is 3 bytes long and includes only the SD, AC and ED bytes (*T*=0). A token is generated by a station that has just finished sending data and travels round the ring until it is captured by another station which wishes to transmit. “Capture” actually involves altering the *T* bit, but passing on the SD and altered AC bytes as the leading bytes of the data frame being sent (the ED byte of the token gets overwritten). However, a prospective transmitter is constrained by the priority mechanism, in that it

is not allowed to capture tokens whose priority is too high (notice that a token has an AC byte and therefore a priority). A station may hold the token for a maximum *token-holding* time, which is usually set at a default of 10ms. In a 4Mbps ring, a station must wait for its last transmitted frame to return before generating a new token. In the faster 16Mbps ring, however, it is permissible to generate a new token as soon as the last bit of the last frame has been sent: this is called *early token release*.



Figure 8.19 Format of control bytes in token ring frames

So how does the priority system work? Basically a station decides before it transmits what priority a new frame should have. It can only capture the token if the token's own priority is no greater than that of the waiting frame. However, even if it cannot transmit it can try to reserve the next instance of the token, using the reservation bits. If the reservation bits on the token or on a passing data frame are lower than the priority of the waiting frame, the prospective transmitter raises the reservation bits to the latter value. The next token generator sends a token at the priority of the highest reservation value it sees, setting the token reservation field to zero. However, what goes up must come down, or lower priority frames would never again get transmitted. The protocol handles this by getting the station which has just raised the token priority to record the new value, *P*, say, and so becoming what is called a *stacking station*. The stacking station has the responsibility for lowering the token priority again when there are no remaining frames of priority of *P* or greater waiting to enter the ring.

Another issue in the token ring is management, particularly initialisation and error recovery. One station on the ring is designated the *active monitor*. Each station has monitor capability, but only one is active at a given time. When a station wishes to join the ring, it performs an initialisation sequence to inform other ring stations of its existence. Its first action is then to make a bid to become monitor, which will only succeed if it is the first active station on the ring. The monitor is responsible for providing timing for the ring as a whole, as well as detecting and repairing lost token conditions. It regularly transmits MAC frames called *AMP* frames (*active monitor*

present). Should these frames fail to appear, then after a set timeout, other stations will bid to become the new active monitor, by sending a *claim token* (*CT*) frame round the ring. The first station to successfully to circumnavigate the ring with a CT is the new active monitor. Its first action is to purge existing frames and tokens by sending a *PRG* (*purge*) frame.

When no station wants to send a frame, the last transmitted token simply circulates around the ring. To enable this to happen, the ring must have a minimum circulation time or *latency* of 24 bits. To ensure this, regardless of how few stations happen to be attached at any time, the active monitor introduces a special 24-bit shift-register *latency buffer* into the ring. In fact, the latency buffer can vary in size from 24 to 30 bits and is therefore said to be *elastic*. This is to combat a phenomenon encountered on rings called *timing jitter*, which is discussed in Section 3.4.

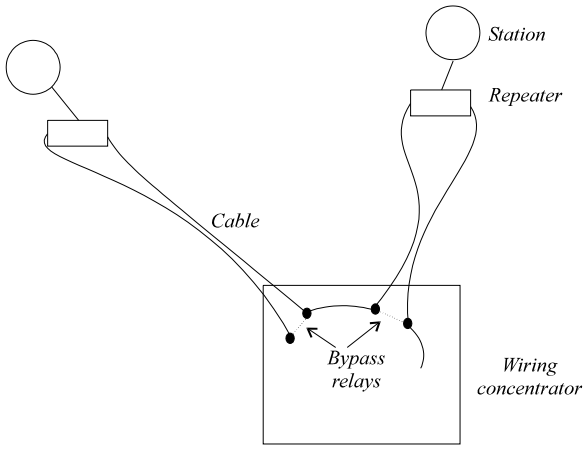


Figure 8.20: Star-shaped ring based on a MAU

From the description of its operation, it is easy to see that if a break occurs anywhere in a ring, the network is completely disrupted. A station detecting a problem, such as a timeout on AMP or on the appearance of a free token, transmits another special MAC frame called a *beacon* (*BCN*). If the beacon circumnavigates the ring, the problem is obviously not with the links, and the beaconing station then attempts to become monitor. If, however, the beacon does not return, a break is assumed. On a true ring there is nothing that can be done in this situation, except to wait for repair, and, indeed, this has conventionally been one of the main criticisms of the ring network. However, ring networks are often physically arranged as so-called *star-shaped rings*, using a hub-like device called a *wiring concentrator* or *multi-access unit* (*MAU*), as shown in Figure 8.20.

It is easy to see the similarity in concept between the MAU and the Ethernet hub, as easily maintainable centralised foci of the network interconnects. The MAU contains bypass relays that not only bypass the repeaters, but most of the ring links as well. If a station reports a suspected cable break or repeater fault, the bypass relay can be used to remove the offending section. This can be done automatically under software control, and the rest of the ring can continue to function normally.

Token passing is not the only way of providing MAC for a ring. Another method is the so-called *slotted ring* first proposed by Pierce in 1972. Here, the ring behaves as a constantly circulating circular shift register, around which travel skeleton packets or *slots*. The number of slots is fixed and depends on the slot length and the bit-length of the ring (including repeaters). Each slot contains a leading bit that marks it as empty or full. A station wishing to transmit, waits for an empty slot, marks it full, and inserts a so-called *minipacke*t of data as the slot goes by. The station is not allowed to transmit another minipacket until the slot returns. The destination station marks response bits in the slot and reads the data. These bits are interpreted as an acknowledgement by the originating station when the slot completes its round trip. The slot is then remarked as empty. Again, a *monitor* station is responsible for emptying slots that have not been emptied by their transmitter. The monitor is also responsible for initialisation of the entire ring at startup. Perhaps the most well known example of the slotted ring system is the *Cambridge Ring* developed at the University of Cambridge in the UK (Wilkes and Wheeler 1979).

In general, the token ring, with its longer frames, has much less control overhead than the slotted ring. Although more vulnerable to failure than a bus, the former is also more fault-tolerant than the latter (failed stations are easily bypassed, and a failed monitor is automatically replaced). The slotted ring has the advantage of a simpler MAC protocol, and very predictable behaviour at any load, but is not able to adjust the bandwidth available to heavy users in otherwise light loading conditions. The adoption of token ring by IEEE 802 has essentially sidelined the slotted ring, which is now mainly of historical importance only.

8.2.4 Token Bus

The last LAN technology originally incorporated in the 802 standard is the *token bus*, 802.4. This system was developed at *General Motors* for use in factory automation scenarios where very high reliability is a primary issue. Its specialist origins and the inherent complexity of its interfaces have left it much less common than Ethernet and token ring, and, in consequence, we shall discuss it here only briefly.

Token bus systems use 75Ω coaxial cable operated in what is called a *carrierband* mode transmitting at 1, 5 or 10Mbps. This is essentially a form of FSK where a “1” is modulated by a carrier equal in frequency to the bit rate, and a “0” by a carrier of twice that frequency. The advantage of this system over baseband transmission lies in its superior noise immunity, although there is a price to be paid in the increased complexity and expense of the interface equipment. The carrierband mode also allows the stations to generate other frequencies used in the frame delimiters.

The token bus frame format is similar to, but simpler than, that of token ring (Figure 8.21). It begins with a short *preamble* to assist receiver synchronisation, followed by a *start delimiter* using the non-data signalling elements. The *frame control* field distinguishes data from control frames, and indicates priority in data frames. Addresses and the CRC are just as in the token ring. A token is simply a special control frame, and, unlike token ring, does not have a special short frame format. The Frame Control field also contains an option that requires the receiver of the frame to acknowledge it immediately. This grants the receiver special permission to send an ACK even though it does not itself have the token—the process is called *temporary token handoff*.

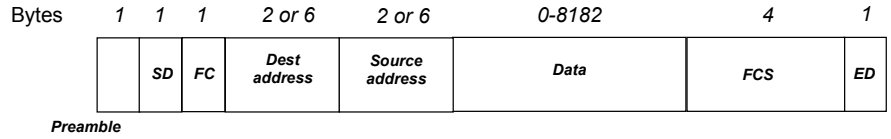


Figure 8.21 Token bus frame format

Data frames may be of four different priorities, which, in increasing order of importance are designated 0, 2, 4 and 6. Each station is guaranteed permission to send priority 6 frames for up to a limit, the *high-priority token hold time*. When a station acquires the token, it must first send any priority 6 frames, until the timer for those expires, then it must move to priority 4 and so on. However, before sending frames of any priority other than 6, it must observe how long the token has taken to circulate round the bus, from the time it last received it. If this *token rotation time* exceeds a set target, the station must relinquish the token at once. The token bus sets a guaranteed worst-case time for token circulation in order to remove the uncertainty associated with systems like CSMA/CD, which have no delivery time guarantees.

Stations on a token bus network that wish to transmit data organise themselves into a *logical ring* for token-passing purposes. There is no equivalent to the monitor station found in the token ring, but instead, each station participates in distributed management and recovery procedures. For example, the following procedure is used to allow newly active stations to join the logical ring. The token holding station periodically sends a MAC control frame called a *solicit-successor*, which contains its own address and that of its current successor. Only stations with addresses between these are allowed to enter, to keep the ring sorted in address order. The token holder waits for twice the propagation delay of the bus, and, if it gets a response, in the form of a *set-successor* frame, allows the responder to enter the ring as the holder's new successor. If more than one station responds there will be a collision, whereupon the token holder issues a *resolve-contention* frame. The contenders then wait for a short random period of time before responding again. The access of new stations to the ring must not cause the worst-case token rotation time to be exceeded, so, if traffic is heavy, a station wishing to enter the ring may have to wait. A station may leave the ring by waiting until it is token holder, then sending a set-successor frame to its predecessor, containing the address of its own current successor. It then passes the token to that successor and takes no further part.

When a token bus starts up the first station that wants to join cannot, of course, participate in the ring algorithm. However, once it notices that there is no traffic on the bus, it can deduce that there is no active ring at all. It then sends a *claim-token* frame. If there is no response, it sets up a ring consisting only of itself and periodically invites other stations to join, in the usual manner.

If a station in the ring fails to react when the token is passed to it, its predecessor repeats the token a second time. If this also fails, the predecessor sends a *who-follows* frame containing its own address. The successor of the non-responding station is supposed to react by sending a set-successor frame naming itself as the new successor, and effectively eliminating the non-responding station from the ring. If there is no response to the who-follows request either, the predecessor sends a solicit-successor

with a full address range to see, in effect, if there is anyone out there who might wish to join a new ring. If the token gets lost, the ring initialisation simply kicks in. If multiple tokens appear, any token holder seeing another transmission simply stops activity and discards its token.

As you can see from this, the token bus is an elegant solution to the MAC problem which has the added advantage of having no single point of failure. Unlike token ring, which originated in the office environment, the factory floor, where this system was originally intended for use, requires very high reliability, and token bus was designed specifically with this goal in mind. However, the complexity of its interface equipment and MAC procedures have left it largely in the shadow of token ring and especially Ethernet, which is quite suitable for the vast majority of situations.

8.2.5 Layering in IEEE 802

IEEE 802 introduced three incompatible MAC standards for use in LANs (we defer discussion of the 802.6 MAN standard until Section 8.3.3). Each of these standards uses a different frame format (MAC PDU), with different header layouts and even different maximum sizes. In all such networks, the MAC sublayer provides a connectionless (datagram) service to its user, which, in the IEEE/ISO model would be the LLC sublayer. The MAC service primitives are simply:

MA_UNITDATA.req, MA_UNITDATA.ind, MA_UNITDATA.confirm.

The request primitive includes the obvious parameters such as the destination MAC address, which may be individual, group or broadcast (as we have seen, at least all 802 addresses do conform to the same basic format). Note that there is no response primitive, and the confirm is therefore generated by the MAC sublayer itself. This may either be done locally, indicating simply that the frame was sent into the physical medium (as in Ethernet) or remotely, indicating that the frame was accepted by the MAC unit at the target (as in token ring). The arrival of a confirm does *not* imply that the remote MAC service user has acknowledged the frame's reception.

The MAC service is extremely basic, and this is, in fact, a strength rather than a weakness. It can be used directly as a subnet access protocol by IP for example, with an Ethernet or token ring frame carrying an IP datagram directly as its payload. However, within the confines of the 802 standard itself, it was recognised that in many applications a reliable datagram or connection-oriented service would be required over the MAC sublayer, bringing its quality up to that provided by, say, an HDLC data-link protocol. This was exactly the aim of the LLC sublayer.

The LLC protocol is in fact pretty much a modified version of HDLC. The sublayer can offer connection-oriented, acknowledged connectionless or unacknowledged connectionless services. A further service called *obtain reply* allows a service user to request an item of data placed in an LLC message buffer by a remote correspondent. LLC addresses are based on the underlying MAC counterparts, but have an additional 7-bit LLC service access point qualifier to permit interlayer routing.

What the LLC protocol aims to do is upgrade the capabilities of the MAC service to something equivalent to a standard data-link service, but without replicating those functions already present in the MAC protocols. For example, 802 MAC frames all

use a powerful 32-bit CRC as FCS. This is already superior to the error detection capabilities of HDLC and it would be pointless therefore to include another CRC in the LLC PDU. Likewise, the MAC frames contain the MAC addresses, so there is little point in including anything other than the LLC address extensions in the LLC header. Thus the LLC frame format is somewhat simplified compared to HDLC as shown in Figure 8.22.

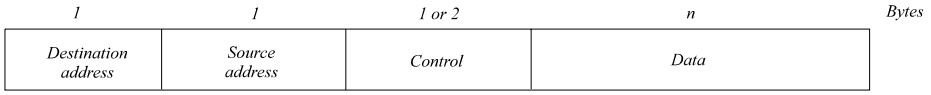


Figure 8.22 LLC frame format

LLC is based only on the ABM mode of HDLC and includes no analogues of NRM or ARM. However, there are *I*-frames, *S*-frames and *U*-frames, with control field formats virtually identical to those used in HDLC. LLC supports three types of operation: *Type 1* supports the unacknowledged connectionless service; *Type 2* the connection-oriented service; *Type 3* the acknowledged connectionless service. For Type 1 operations, the frame format used is UI (*unnumbered information*), which has no sequence number and supports neither acknowledgement nor flow control. Type 2 operations are pretty much like standard HDLC, with sequence-number based error and flow control. One minor difference with HDLC is that there is no SREJ frame and therefore no support for selective repeat. For Type 3 operation, LLC introduces a new frame type not present in HDLC, called an *acknowledged connectionless* (*AC*) *information frame*. Data is sent in an AC *command* frame, using a one-bit sequence number; an acknowledgement, also with one-bit sequence number, is returned via an AC *response* frame.

In closing it is worth mentioning that, in the case of 802.3 Ethernet, IEEE has defined a protocol, *802.1a*, known as *SNAP* (*subnetwork access protocol*), which is carried by LLC and provides for compatibility with the pre-802 Ethernet II standard. This does not use the LLC address fields, which are each set to AA_{16} to signify the existence of a SNAP header. The SNAP header then contains a so-called *ethertype* field, which was used by Ethernet II to distinguish delivery buffers (somewhat equivalent to an LLC SAP).

8.2.6 Connecting LANs: Bridges

We have seen that there are many circumstances where it is desirable to connect subnets together and how network-level gateways can be used to handle the problems of inter-connectivity between such subnets. A special case is the connection together of two or more LANs, administered by the same organisation. Of course, in the most general sense, LANs can be connected together just like any other networks via IP or multiprotocol routers. However, from the point of view of an organisation running several different LANs which it wishes to interconnect, the use of Internet routers, say, pretty much forces the networks to use a specific protocol stack. The reason for this is that connection via such routers is at what OSI calls the network layer, and, in general, once the network layer is fixed, the rest of the stack follows.

An alternative, which does present difficulties of its own is to connect the networks at the data-link level, effectively attempting to convert one MAC protocol

format to another. A device that performs this function is called a *bridge* (Figure 8.23).

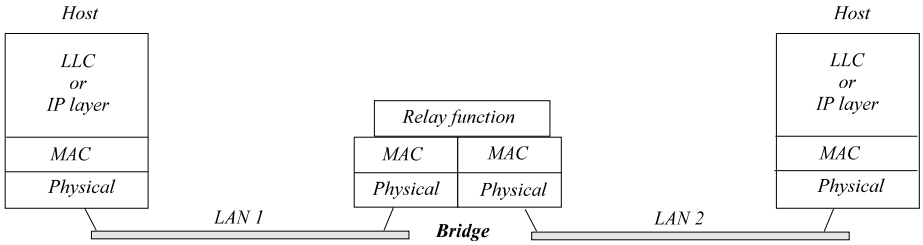


Figure 8.23 Bridge joining two LANs

It is instructive to compare a bridge with the other means of connecting networks, or parts of networks, together. First of all, a bridge is unlike a router, in that it has no involvement in the network or internetwork protocols. When a MAC frame arrives at a router, it is stripped down, and its content, for example an IP datagram or an X.25 packet, is extracted. The router routes according to the information contained in the IP or X.25 header, repackages the packet in the MAC or data-link format of the next network, and then transmits it. A bridge, on the other hand, never examines the packet payload of the MAC frame, but uses information in the MAC control field itself (although see the discussion below on *source routing bridges*). It may use this to route where necessary, and to assist in transforming the frame format to that required on the next leg of the journey.

Another interesting comparison is with repeaters, which are used to join LAN segments together, regularly, as in rings, or irregularly as in conjoined buses. However, a repeater is essentially a physical level device, which simply copies bits from between segments, and does not buffer (except perhaps for a few bits) or attempt translation from one PDU format to another. A repeater, in effect, is used to join two segments that are part of the same network. A bridge, on the other hand, joins two separate networks together, stores complete frames and performs protocol translations between different MAC formats. A bridge is not as general as a router, and relies on the linked networks having similar MAC protocols, but it is, in a sense, more efficient requiring no packet-level functionality at all.

One question that may occur following this discussion is why, in fact, a single organisation might prefer to run several interconnected LANs rather than simply a large all-encompassing one. There are in fact several good reasons. Firstly, as LANs, particularly 802 LANs, are broadcast networks, they have a limited capacity in terms of the number of stations which can be supported by a simple single medium (with or without repeaters), while multiple interconnected media can increase the average bandwidth available to individual stations. Secondly, many organisations are departmentally structured, and administratively, different departments may wish to have their own dedicated LANs, and may indeed, if permitted the autonomy, choose different technologies. Thirdly, dividing a network into several physical subnets can assist in security and reliability, especially where the subnets are broadcast based, and one station sends its transmissions simultaneously to all others. Finally, where there are several sites or one or more very large site, a single organisation may be forced to

use multiple LANs in any case, because of the distance limitations of the LAN technologies. Bridges on their own may not be adequate for this type of connectivity, and some longer range switching fabric or wider area backbone (say a MAN) may be required to link the separate sites. We shall return to this problem later, here pausing only to note that such a backbone may take one of several forms. It might, for example, consist of leased lines rented from the public telephone provider, a limited MAN owned by the organisation itself (FDDI networks are commonly used to cover extended privately owned sites in this way), or a full MAN or wide area backbone facility, provided by a third party.

If full gateway translation is not required on and off a backbone, or if leased lines link the sites, a technique called *remote bridging* can be used. Here, special bridges are required which transform the MAC frames at each end into a format that can be carried across the backbone or leased lines. With leased lines the bridges really need to do nothing except transmit incoming MAC frames to each other directly over the long distance phone lines. With an intervening backbone, the bridges can encapsulate the MAC frames in the packet format of the backbone, resulting in a form of tunnelling between the remote bridges. For example, if the backbone is an X.25 network, a MAC frame can be tunnelled by encapsulating it in an X.25 packet at one end, and retrieving it at the other (Figure 8.24). If the backbone is a MAN, especially an 802.6 MAN, bridging onto and off the backbone may be possible directly, since the MAN format is sufficiently compatible to permit direct translation without tunnelling.

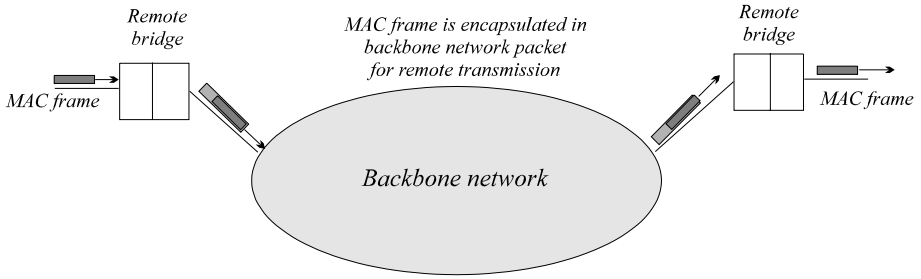


Figure 8.24 Remote bridging with backbone tunnelling

Returning to the simpler case of two or more adjacent bridged LANs, we can observe that there are obstacles to be overcome. Some facilities that operate properly over a single LAN cannot function given a protocol-translating entity intervening between source and destination. Firstly, bridges may have to discard frames under certain circumstances. This is because there is no flow control at the MAC level, so a sudden influx of frames can cause a bridge to run out of buffer space, and, with no way of asking its sources to desist, frame discard is the only option. Secondly, bridge delays may cause problems with standard timeouts set on delivery of data by underlying LANs. Finally, since frames are transformed, error detection check sequences must be recalculated at each bridge, so that data is not protected while actually in a bridge.

These are generic problems, but other difficulties can be caused by incompatibilities between the MAC protocols in use on the source and destination networks. For example, different MAC protocols specify different maximum frame

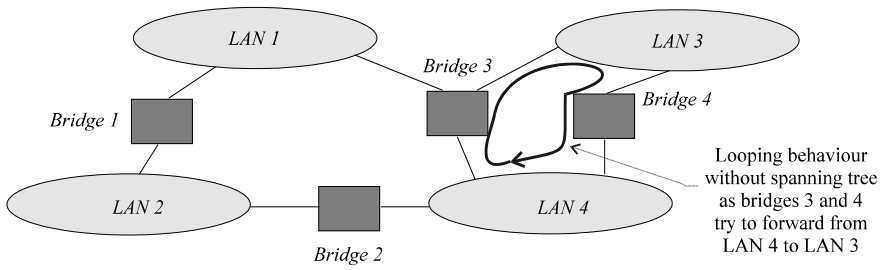
sizes. If a frame that is within the size limit of the source LAN is too big for the destination, it will typically have to be discarded. Problems can also be caused by different priority systems in different MAC versions. A frame entering an 802.3 (Ethernet) for example must have any priority information discarded; while a frame coming from an 802.3 to a network supporting priority, will have some default value inserted by the bridge.

IEEE have studied and standardised bridge technology as part of the 802 project. In particular, two bridging approaches have been adopted. In *transparent bridging*, the idea is that the bridges themselves do all the work, transparently translating frames from one 802 protocol to another and performing routing, based on intelligent interpretation of the destination MAC address. In *source routing* the source host determines the route that a frame must take through an internetwork of LANs and bridges, including this information in the MAC frame itself for the bridges to interpret. Transparent bridging is specified in IEEE 802.1(D) while source routing, which was developed specifically for token ring networks, is included in 802.5. Bridges of each type may connect two or more LANs directly, the connection to each LAN being called a *port*. Each bridge in an internetwork has its own unique identifier, as well as MAC addresses for each of its ports.

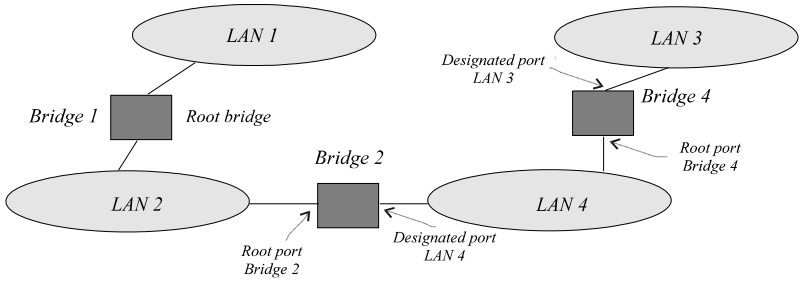
The main complication with bridges is that they may be used to connect multiple LANs together into complex internetwork topologies. Further, MAC addresses have no hierarchical structure that would assist with routing. Transparent bridges attempt to maintain *forwarding databases*, which indicate where a frame with a given destination MAC address should be sent. Because LAN addresses are dynamic, in the sense that the point of attachment of a machine may be changed anytime, transparent bridges try to learn where particular addresses are located by observing the *source* addresses of frames passing through. A frame with source address x arriving on port k of a given bridge, will cause that bridge to create or update an entry in its forwarding database, suggesting that any frame addressed to destination x should be sent on through port k. This is called *backward learning*. Initially, a bridge will flood frames whose destination addresses are unknown to all connected LANs other than the one on which the frame arrived. However, as the forwarding database is constructed, the bridge becomes more discriminating, and sends each frame only to the network on which its destination is known to reside. Unfortunately, in complex internetworks where two LANs are connected by more than one bridge a pathological looping behaviour can set in if this simple learning process is not controlled.

Once the internetwork becomes sufficiently complex, a bridge will see frames that are not destined for any LAN to which it, itself, is connected. The bridge must be able to identify the LAN on which the next bridge to which the frame should be sent resides. 802.1(D) bridges in a complex web of internetworked LANs, communicate with each other using a *bridge protocol*, and use this to construct a spanning tree which they all share, in order to prevent the looping behaviour mentioned above. The bridges use this tree to determine the route that any frame crossing the bridged internetwork must follow. The spanning tree is computed using a *spanning tree algorithm*, whereby, bridges send *BPDUs* (*bridge PDUs*) to each other to determine, firstly a *root bridge*, and secondly, a preferred path from each other bridge through a specified *root port* back to the root bridge. If multiple bridges are interfaced to a given

LAN, only one, the *designated bridge* is allowed to pass frames into that LAN, preventing looping. Even once the spanning tree has been established, 802.1(D) bridges continue to pass BPDUs to each other to keep the tree updated.



a) LANs connected by transparent bridges



b) Possible spanning tree for internetwork of (a)

Figure 8.25 Spanning tree bridging of LAN internetwork

Consider the internetwork shown in Figure 8.25a. Notice that there are a number of places where bridges using backward learning could generate loops. Using the bridge protocol, the bridge with the lowest serial number becomes the root bridge. Other bridges use the rated capacity of each port as a cost measure for crossing the LAN to which the port is attached, and, based on this measure, calculate a spanning tree, such as that shown in Figure 8.25b. In this tree, each LAN gets one designated bridge, with a designated port, which has the sole right to forward frames to that LAN. Some bridges, such as Bridge 3 in this example, may be entirely removed from the tree and are no longer used for forwarding frames, although they may still prove useful if the tree has to be recalculated, say, due to a failure. A bridge that is part of the tree has a single root port through which all frames passing to or from that bridge to the root bridge must pass.

Transparent bridges are intended to connect LANs without stations needing to worry about the relay process. The source routing bridges defined as part of 802.5, are quite different, in that stations are actually expected to decide a route before a frame is sent, and then include that route in the frame itself. The bridges have no decisions to make, merely reading and acting on the route implanted by the source station. This approach makes no pretence at transparency, and indeed requires changes to the MAC

frame format. The frame is augmented to include a *routing information field*, which is inserted just before the data field. To indicate that the field is present, a frame must set the most significant bit of its source address to "1". In a destination address, this bit differentiates individual from group addresses, but since a source address must always be individual, in the absence of source routing, the bit would always be "0". This bit is known as the *routing information indicator* or *RII*. A source routing bridge may perform a hardware examination of the RII bit of an incoming frame to decide whether it needs to store the frame at all.

The routing information field consists of a two-byte *routing control field*, followed by a sequence of up to 14 two-byte *route designators*, each of which corresponds to one "hop" along the route. A route designator consists of a single *LAN number* of 12 bits followed by a *bridge number* of 4 bits.

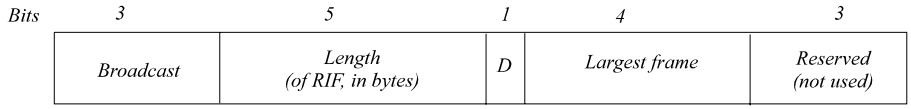


Figure 8.26 Routing control field

The routing control field (Figure 8.26), includes a three-bit field which indicates one of four types of *routing directive*:

- *none* where the destination is on the same LAN as the source, and bridges should not forward the frame;
- *non-broadcast,* which is a simple transfer to one destination, not on the same LAN as the source;
- *all-routes broadcast*, sends a frame along all possible routes through all bridges;
- *single-route broadcast,* sends each frame to each destination on a pre-calculated spanning tree.

In all-routes broadcasting, there are no route designators included at the source, but each bridge adds its bridge number and outgoing LAN number to each copy of the frame that passes through. To prevent infinite loops, a bridge will not pass on a frame to a LAN whose LAN number is already present in the RII. Nonetheless, a destination will typically get multiple copies of the frame, each having arrived by a different route. All-routes broadcasting can be used by a source that is ignorant of the network's topology, to help it discover routes. To do this the source simply performs an all-routes broadcast with a so-called *discovery frame*. When a copy of the discovery frame arrives at the destination, its RII contains a complete route from the source. The destination now simply replies with a non-broadcast response frame along the route it finds in each such copy.

Single-route broadcast requires that the bridges have knowledge of a pre-calculated spanning tree. This tree can be worked out in the same manner as for transparent bridges, or may even be supplied manually. However, once it is known, a station can broadcast in a manner that ensures that all destinations get only a single copy of the transmission. An alternative discovery procedure is possible using this, whereby a single-route discovery frame is sent across the internetwork. Each receiving station then replies with an all-routes broadcast, which ensures that the source will get

copies of the reply along all possible routes to the destination. This approach relieves the destination of the onus of replying to multiple copies of the discovery frame.

The *direction bit* in the routing control field indicates whether the frame is an original or a reply, travelling along the same route but in the opposite direction. This allows a station responding to an incoming frame to use virtually the same routing information field as the latter. Finally, bridges update the *largest frame* field at the route discovery time, to indicate the largest frame that may pass along a given route. Recall that bridges must discard a frame too big for the LAN on which it is to be forwarded, and although source routing bridges are specifically defined in 802.5, they are intended to permit connection of other LAN types as well. In particular, 802.3 has a relatively restrictive size limit of only 1500 bytes.

Source routing bridges were strongly advocated by the committee working on 802.5, and have the advantage that they allow exploitation of all possible routes and bridges, unlike transparent bridges, which are restricted in a given configuration by its current spanning tree. However, greater load is placed on the stations, both in determining routes and in monitoring failures—stations must themselves notice when a route suddenly fails, by timing out responses to their transmissions, and must then engage in new discovery processes. Also the discovery process itself is very expensive in the number of frames it requires to allow all sources to determine routes to all destinations.

One problem shared by the two bridge types is that they are effectively incompatible with each other, and hence would seem to exclude the possibility of internetworks of LANs that use both. Ever determined to give source routing a fair chance, the 802.5 committee have defined a *source routing transparent* (*SRT*) *bridge*. This bridge acts like a source routing bridge until it spots a frame with no RII. Instead of ignoring such a frame, as a true source routing bridge would, the SRT bridge turns into a transparent bridge, and handles the frame accordingly. Stations which are not participating in source routing never set the RII, so frames they transmit can only be forwarded by transparent or SRT bridges.

In a large organisation, the local network topology can become very complex, consisting of many individual interconnected LANs bridged together, possibly remotely over some backbone. An individual LAN may itself consist of several segments connected by a LAN switch (most commonly an Ethernet switch). Of course different stations may use different protocol stacks to communicate with others: although TCP/IP is now the commonest, IPX, Appletalk and NetBIOS are frequently still used in local contexts.

When bridges and simple Ethernet switches are used for connection, any transmitting station which wants to multicast a frame to a group of targets, must broadcast the frame across the whole internetwork, to be sure that every target is reached. This is because MAC and data-link protocols lack the multicast support of network protocols like IP. With a large number of possible sources, intruding broadcasts can eat away a substantial portion of the available bandwidth on a segment. One way round this is to use routers, which can exploit the multicasting capability of protocols like IP instead of bridges. However, such routers are more complex, more expensive and typically slower than bridges. The routers also need to worry about

each type of network level protocol separately—NetBIOS for example, is inherently not routable.

Another approach that offers great flexibility for a network administrator is the use of so-called *virtual LANs* (*VLANs*). The idea is to separate the entire LAN internetwork into *VLANs*, which are *broadcast domains* consisting of stations each of which should see all packets broadcast by any other. Membership of a VLAN may change dynamically, and is independent of the physical layout of the internetwork.

VLAN capability can be added to a bridge or Ethernet switch (sometimes then called a *Layer 2 VLAN switch*) by giving it the ability to store details about which stations belong to which VLAN. An administrator, using VLAN configuration software running on a separate management station can then typically alter such details. However, Layer 2 VLANs are in general only effective for relatively small internetworks. One difficulty is that MAC addresses which, of course, bridges use for routing, are fixed to particular machines and cannot be reassigned easily so there is no simple way for an operator to set up a Layer 2 VLAN other than by listing each individual station's MAC address. It is also not clear how traffic should be passed from one VLAN to another, when this is necessary since there is no provision in MAC protocols that would support such transfers.

An alternative is to use a *Layer 3 VLAN switch*, essentially a simple router that can have the VLAN details programmed into it. This switch will typically be much simpler and therefore faster than a true Internet router. It would not, for example, support routing protocols like RIP or OSPF, and it would not perform protocol interconversion. Layer 3 VLAN switches can group users according to level 3 addresses (IP addresses, for example) using techniques like subnetting to define new VLANs. In this approach the VLAN is clearly associated with network level addresses rather than MAC addresses.

8.3 MANS, BACKBONES AND HIGH-PERFORMANCE LANS

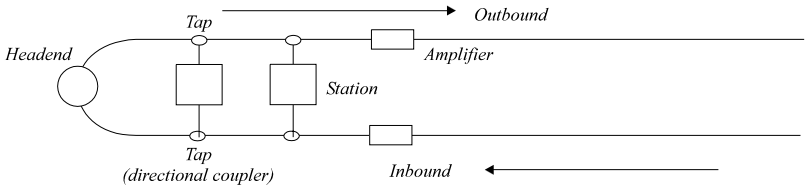
8.3.1 Broadband Cable

A metropolitan area network is intermediate between a LAN and a WAN, being intended to cover a region of approximately city dimensions (10-100km). MANs, which may be operated privately or in the manner of public utilities, are envisaged as ways of providing high bandwidth access at relatively low cost to corporations and individuals within the service area. They are also used to connect LANs in the same geographical region together (a so-called *backbone* function). Because of the additional distance requirements involved, the physical media used in MANs differ somewhat from those encountered in LANs. While LANs tend to use twisted pair point-to-point links and coaxial cable buses, the technologies used in MANs include broadband coaxial cable buses (which may be unidirectional or bidirectional), and buses based on optic fibre or baseband coax (unidirectional only), with repeaters inserted at appropriate points.

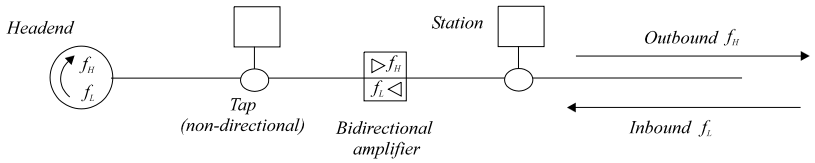
Looking at broadband systems first, FDM is used to divide up the frequency spectrum of the channel. The medium is usually the familiar broadband coaxial CATV (cable television) cable with a useful bandpass from about 5 to 300/400 MHz. The lowest frequencies are not used since transfer characteristics in this type of cable fall

off sharply at low frequencies, leading to significant distortion over long distances. However, within the band stated, broadband signals suffer less distortion than baseband, and, can be amplified to increase range to tens of kilometres. The technology required is essentially the same as that used widely in CATV, and there is scope for extending existing CATV cable networks into MANs.

However, it is inherently not possible for an amplifier to amplify the same frequencies in both directions, so the use of such devices introduces unidirectionality into any broadband system. There are two practical solutions to this problem (Figure 8.27).



a) Dual cable



b) Split cable

Figure 8.27 Broadband cable configurations

a) *Dual cable* configurations use two cables joined at a point known as the *headend*. All stations transmit on one cable, which carries signals towards the headend (*inbound*). Signals received at the headend are then propagated along the second cable (*outbound*) on which all stations receive.

b) *Split* configurations also use separate inbound and outbound paths, but these are implemented as different frequency bands on a single cable. This system is commonly used in CATV, where communication is basically unidirectional, but a limited bandwidth reverse path is often provided, to allow subscriber polling etc. In CATV, therefore, the bandwidth of the inbound path is much smaller than that of the outbound, but in LAN applications the two paths are approximately equal. The typical CATV division, *subsplit*, allows for only a 25MHz inbound path as against a 346MHz outbound. For pure LAN or MAN applications *highsplit* (168MHz each way) is more useful, while for combined CATV/MAN networks there is *midsplit* (111MHz inbound, 232MHz outbound). In MANs, lower frequencies are used by stations for transmission along the inbound path, and are translated at the headend by a device called a *frequency converter*, before being retransmitted in the outbound receive band. So-called *bidirectional amplifiers* amplify the low frequency signals in one direction and the high frequency signals in the other, so that each frequency

band is inherently unidirectional. Incidentally this does not contradict the statement that amplifiers are unidirectional for signals in a given band. In a CATV system, of course, the headend contains the TV signal transmitter.

The differences between the systems are not great: the split configuration is somewhat cheaper, but has only half the total bandwidth and, because of the more complex headend frequency converter, is somewhat more vulnerable to failure.

Channel access in a broadband network is typically organised in one of three ways.

i) *Dedicated access*, where a portion of the cable bandwidth is reserved for each pair of stations, is mainly suitable for a situation where dedicated communication paths are required (rather like leased telephone lines).

ii) *Switched access* requires each station to be able to transmit and receive on several frequencies. Connection to the medium is therefore via *frequency-agile modems*, which can change their frequency on command. Stations wishing to establish a connection send a request to a controller, which assigns an available frequency, and instructs their modems to tune to that frequency. The problem with this apparently flexible approach is that frequency-agile modems are both expensive and usually limited to fairly low data rates (typically up to 56kbps).

iii) *Multiple access* allows many devices to try to use the same frequency band at the same time, sharing by means of some MAC protocol like CSMA/CD or token passing. This is in fact the commonest form of access control in broadband LANs and MANs. The radio frequency, or *RF*, *modems* required for transmission and reception in such systems do not in general have to be "agile" in the sense discussed, so this approach is usually much more cost effective than switched access.

One problem in the whole area of CATV data systems is a lack of standards, so that, for example, cable modems from two manufacturers will frequently fail to interwork. An IEEE working group, *802.14*, is currently looking into this, with the aim of developing standards for the provision of, primarily, LLC (but also, possibly, ATM) service over CATV systems.

8.3.2 FDDI

There is probably a future for broadband public access MANs simply because the CATV infrastructure is already in place. However, where MANs are intended at least partly as backbones for the interconnection of LANs, other technologies are likely to predominate. One technology that has been adopted for use in, especially, private backbones, is an adaptation of token ring operating at a much higher speed and called *FDDI* (*fibre distributed data interface*).

FDDI, originally developed by ANSI, and now defined by ISO 9314, uses, in its standard form, multimode optic fibre with LED transmitters to achieve transmission rates of 100Mbps at error rates of 1 bit in 2.5×10^{10} or better, in each direction, over a total distance of 200km[1]. Up to 1000 stations are allowed, although the distance between any two must not exceed 2km. A single-mode laser-driven variation on this is also defined which permits inter-station distances of up to 60km. There is also a much

[1] To allow for wrap-around during failure, ring circumference is limited to 100km.

shorter-range version of the network which uses Category 5 twisted pair and is sometimes called *CDDI* (*copper distributed data interface*).

The capabilities of FDDI make it suitable for use as a high-speed LAN or as a backbone network connecting many LANs. Because of the high rate of data transmission, Manchester encoding, which would require a baud rate of 200MHz, is not used in favour of a line code called *4B/5B-NRZI*. This code is basically NRZI, with a clever enhancement to ensure sufficient edges in the data, and provide delimiting sequences for frames. The enhancement is to code each block of four data bits as five transmission bits, as shown in Table 8.1, so that a 100Mbps data rate can be achieved with a signalling rate of 125Mbaud. Since there are only sixteen 4-bit strings, sixteen of the five-bit codewords are left over. Some of these are used for MAC control purposes, and are given the names J, K, T, R and S.

Code group	*Interpretation*	*Code group*	*Interpretation*
11110	*0000*	*11010*	*1100*
01001	*0001*	*11011*	*1101*
10100	*0010*	*11100*	*1110*
10101	*0011*	*11101*	*1111*
01010	*0100*	*11111*	*Idle*
01011	*0101*	*11000*	*J*
01110	*0110*	*10001*	*K*
01111	*0111*	*01101*	*T*
10010	*1000*	*00111*	*R*
10011	*1001*	*11001*	*S*
10110	*1010*	*00000*	*Quiet*
10111	*1011*	*00100*	*Halt*

Table 8.1 4B/5B code

The FDDI frame structure is similar to that of token ring, although the difference in line code results in some variations. Figure 8.28 shows the general frame and token format, in *symbols*, rather than bytes. A symbol is simply an item from one of the rows of Table 8.1 (4 bits prior to applying the 4B/5B code, 5 bits thereafter). Unlike token ring, FDDI repeaters resynchronise the incoming bit stream to a local clock. Each station has an elastic buffer, initialised at the beginning of every new frame to 5 bits, but variable in size from 0 to 10. Bits are clocked into this buffer using clocking information retrieved from the 4B/5B-NRZI signalling stream (generated by the transmit clock of the immediately upstream station) and clocked out by the repeater's own local clock. Clocks in an FDDI installation must be accurate to 0.005%, and frame size is limited to 4500 bytes (45000 bits) which is enough to guarantee that the buffer will not empty or fill up in the duration of a frame. If bits come in slightly too fast, then, when the discrepancy accumulates to one complete extra bit, the elastic buffer is expanded by one bit to hold it; if bits come in slightly too slowly, the elastic buffer contracts. If the clocks are in phase at the beginning of a frame, both at the centre of a bit time, then a drift of four and a half clock cycles is needed to cause 5 extra or 5 lost bits. At the level of accuracy stated, this will take at least 90000 clock cycles, which gives a considerable margin of error

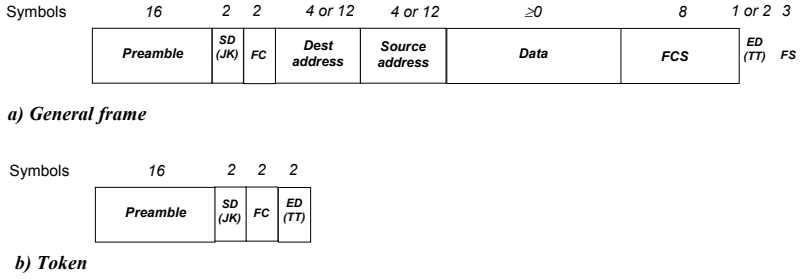


Figure 8.28 FDDI frame formats

As a frame reaches a repeater the receive and transmit clocks must get into phase, and the elastic buffer is initialised to its mid value of 5 bits. To give the repeater scope to achieve this, a *preamble* prefixes each frame before useful control or data appears. The preamble initially consists of 16 idle characters (in 4B/5B-NRZI this is a sequence of 80 alternating high and low levels), although repeaters may increase or decrease its size as the frame passes them and they initialise their elastic buffers. All in all, typical delay caused by a repeater, including the buffer and propagation through the logic circuits, is of the order of 1μs.

C	L	F	F	Z	Z	Z	Z

Figure 8.29 Frame control byte

The *start delimiter* (*SD*) simply consists of the symbols JK, while the *end delimiter* (*ED*) may be either one or two T symbols. The *frame control* (*FC*) field is 8 bits long as shown in Figure 8.29. *FF* indicates whether the frame is a true data frame or a MAC frame or some reserved type. If a MAC frame, *ZZZZ* records the type. *L* indicates whether the addresses used will be 16- or 48-bit (in FDDI, both types of address are permitted on the same network), while *C* is employed to distinguish between asynchronous and synchronous frames (discussed below). In a token the FC field is either 10000000 or 11000000.

The FS field includes three symbols: *E* or *error detected*; *A* or *address recognised*; and *F* or *frame copied*. Each of these can be transmitted as the symbol *S* (set) or *R* (reset). The data carried by the frame may be variable length, but a maximum frame size of 4500 bytes (9000 symbols) is stipulated.

Unlike token ring, an FDDI installation includes two rings operating in opposite directions. The second ring may be used in concert with the first, providing a network with a bandwidth of 200Mbps. However, the real utility of the second ring is as a fail-safe that allows the network to keep operating in the case of catastrophic failure. The idea is that the rings can be cross-connected as shown in Figure 8.30 should a break appear in the rings.

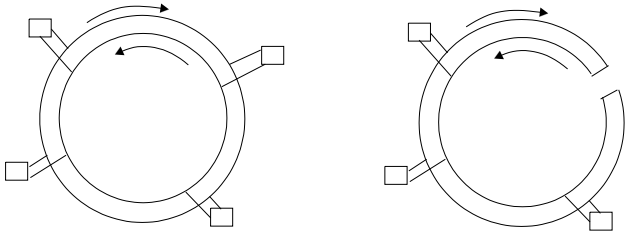


Figure 8.30 FDDI failure

A station may be attached to one or both rings. However, in practice, a hub-like arrangement is most commonly used, with interconnected wiring concentrators. Since the recognised start delimiter is physically a 10-bit string (JK), each station receives bits into a 10-bit shift register. The access protocols for FDDI are similar to those for 802.5 with early token release, but omitting the 802.5 priority mechanism. In FDDI all frames are the same priority, but stations do use a mechanism similar to that of token bus (802.4) to determine how long each may retain the token. The FDDI ring has a parameter called the *target token rotation time* (*TTRT*), which is the time expected for the token to pass once around all stations on the ring. Each station measures the actual *token rotation time* (*TRT*) and computes the difference, called the *token hold time* (*THT*):

$$THT := TTRT - TRT \ .$$

If $THT > 0$ when the token arrives, an *early token*, the station may retain the token and transmit frames, until THT = 0 (or there are no more frames to send), whereupon it resets TRT and sends the token on. If $THT \leq 0$, a *late token*, the station simply resets TRT, but cannot send any frames until the token comes round again. TRRT must be set to be long enough to allow a token to be transmitted, propagate round the ring, and permit at least one station to transmit a frame of the maximum allowable size. Thus

$$TTRT > T(token) + L + T_{\max}(frame)$$

where, $T(token)$ is the token transmission time, L is the *ring latency*, and $T_{\max}(frame)$ is the maximum frame transmission time. In practice the minimum value stipulated for TRRT is 4ms, the typical value is about 8ms, and the maximum 165ms.

The trouble with this mechanism is that it can result in quite high delays in getting frames onto the ring. In particular, under heavy load, a station, *A*, which acquires the token, will retain it until it exhausts the *TTRT*. When the token is passed on, no other station on the ring can use it, so it passes back to *A*. However, *A* cannot use it either, since its *THT* is now less than zero, so only then does it get passed on to *A*'s successor, which will have a *THT* of (*TRRT* - *L*). This station will then use all this time to transmit frames, then pass the token on again. Each time the token is passed on, under such maximum load conditions, it must circulate once round the entire ring before passing to the successor station (Figure 8.31).

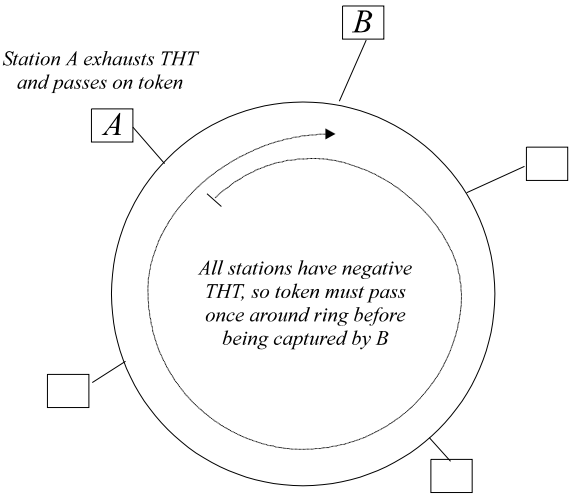


Figure 8.31 FDDI token transfer under maximum load

The worst case time, T_A, for a frame to gain access to the ring, ignoring local queuing, is thus:

$$T_A = (n-1)(TRRT - L) + n(L + \frac{L}{n})$$

where we assume there are n stations, equally spaced, so that the direct latency from A to B is $\frac{L}{n}$. The value of T_A can be quite high for a large ring, and this is sometimes undesirable for certain types of data. To circumvent this, FDDI optionally allows for the transmission of higher priority, so-called *synchronous frames*, which have the C bit of the FC byte set. A station is permitted to send synchronous frames up to a certain allowance of ring time, called its *synchronous allocation time* (*SAT*), regardless of the value of *TRT*. The SAT is granted to a station by a special *ring management station*. Each station on the ring can request a SAT from the ring management station using a *station management* or *SMT* protocol. The SMT protocol is also used to negotiate the value of the *TRRT*, which is now set so that:

$$TTRT > T(token) + L + T_{\max}(frame) + \sum_i S_i$$

where S_i is the SAT for station i. Transmission of asynchronous frames can proceed just as before, but now each station can send its allocation of synchronous frames, regardless of whether its *THT* is positive or not. As a result, the actual *TRT* for a station can now exceed the specified *TTRT* by as much as $\sum_i S_i$, so that the actual *TRT* may be well in excess of *TTRT*, although it will not be as much as $2 \times TTRT$.

The use of synchronous frames makes the FDDI MAC protocol rather like that of token bus with just priorities 6 (analogous to synchronous) and 4 (asynchronous)

implemented. The main benefit is a considerable reduction in ring access times for incoming frames, which can be useful for certain types of data traffic, such as voice. However, even with synchronous access, FDDI retains too much packet jitter to make it a really ideal vehicle for multimedia traffic. In an attempt to overcome this problem, a superset of FDDI, called *FDDI-II* has been developed which allows the conveyance of isochronous traffic by synchronously time-division multiplexing the ring bandwidth to permit the carrying of precisely sampled streamed data such as PCM voice or video. However, FDDI-II, which also operates at 100Mbps, is not common at present, and will probably be unable to compete with faster, more flexible, technologies such as ATM.

If a station does not see a valid token for twice the stipulated *TRRT*, it assumes the token has been lost and starts a so-called *claim token procedure*. This involves the issue of a continuous sequence of SMT *claim* frames, each of which includes a suggested value for the new *TRRT*. If more than one station initiates this, the one with the lowest suggested *TRRT* wins (the one with the largest address takes precedence in the event of a tie). If a station sees claim frames higher in priority than its own, it repeats the former and ceases to transmit the latter; if a station sees claim frames lower in priority than its own, it discards the former and continues to generate the latter. Eventually a station will receive one of its own claim frames, which has completed a circuit. This station's bid is the successful one, and its *TRRT* is henceforth used by all. The lowest *TRRT* is given precedence to allow a station with a time critical application to bid for a more responsive ring. The winner now becomes the ring management station and is responsible for initialising the ring, which it does by transmitting an *unrestricted token*: this may not be captured on its first lap but it causes each station to reset its TRT and begin the timed rotation procedure. Stations may bid to raise the TRRT at a later stage by sending a request to the management station. If the request is agreed, the new value of TRRT is communicated to all stations.

One final point concerns the layer structure of the standard. FDDI is structured according to the same layer divisions as IEEE 802, with MAC and physical layers. Since FDDI can be implemented over a number of physical media, some physical functions are media specific, others are not. For example, the twisted pair implementation uses a different coding system (called *MLT-3*) from the optical 4B/5B-NRZI. The functionality of the physical layer which is media specific is placed in a sublayer called the *physical medium dependent* or *PMD* sublayer. The functionality involved in making sure that all physical media look the same to the MAC layer, then becomes the *physical convergence* sublayer. This distinction is drawn in many technologies where a single MAC protocol is used over multiple possible media. Incidentally, FDDI considers the station management (SMT) function to be a functional block outside and parallel to the MAC-physical protocol stack, but in communication with it at all levels.

8.3.3 Distributed Queue Dual Bus

Since IEEE 802 was intended to be a universal standard for MANs as well as LANs, IEEE wanted to include a definitive recommendation for the former. The technology they chose was based on dual unidirectional buses, which would eventually be based

on optic fibres but could also use cable technology. This MAC standard, encapsulated in *IEEE 802.6*, is called *distributed queue dual bus* or *DQDB*.

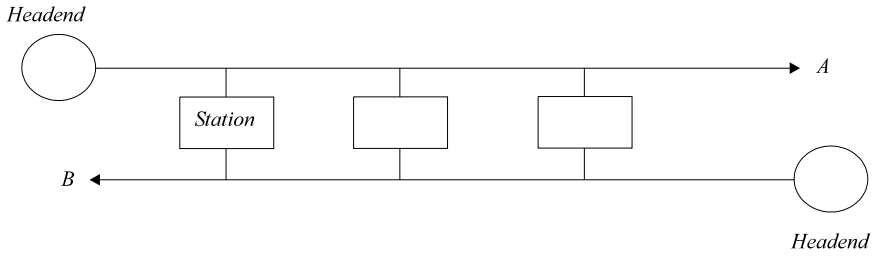


Figure 8.32 DQDB configuration

Like the dual bus broadband configuration, DQDB uses two unidirectional cable buses (Figure 8.32). Unlike dual bus broadband, however, the buses, which we shall refer to as *A* and *B*, are not separated into transmit and receive functions and are used in an asynchronous TDM (ATM) mode. Further, the *headend* functions are carried out by stations, which happen to be near the end of each bus. In fact there is an alternative to the so-called *open bus* topology shown in Figure 8.32, where the buses are connected round into a ring, with a single station acting as headend for both—this is called *looped bus*. DQDB technology is intended for backbone use, and so many of the "stations" attached, are actually likely to be bridges or gateways to other subnets. Each station is attached to both buses, typically via repeaters of some kind, using *A* to send to targets to the right and *B* to send to targets to the left.

Each bus *headend* generates a continuous series of 53-byte cells or *slots*, which can carry a 52-byte payload, called a *segment*, but are initially empty. The idea is that stations obtain the right to place data segments in these empty slots which are then carried to a chosen destination. The initial byte of a slot, which is not part of its segment, is called the *access control field* (*ACF*) and governs the way in which stations can access the slot. The first bit of the ACF is the *busy* bit which is "0" in an empty slot, but "1" in a slot carrying data..

The DQDB MAC layer supports three types of service: *connectionless* which can carry frames (MAC SDUs) of up to 9188 bytes; *connection-oriented data*; and *isochronous*. Data from all these services are ultimately fragmented and entered into the slots. For connectionless and connection-oriented services this is done using a procedure called *queued arbitrated* (*QA*), while for isochronous data an alternative procedure called *pre-arbitrated* (*PA*) is employed. Data from the connectionless and connection-oriented services, once submitted at the MAC service interface, is used to generate MAC PDUs, the format of which is specified in the 802.6 standard, but need not concern us here. These PDUs are converted into a stream of 48-byte *segment payloads*, each containing a 44-byte fragment or *segmentation unit* of the original PDU. The segment payload is then encapsulated into a 52-byte *segment*, as shown in Figure 8.33, ready for insertion into an empty slot. The *slot type* bit in the ACF is set to 0 for a QA slot and to 1 for a PA slot.

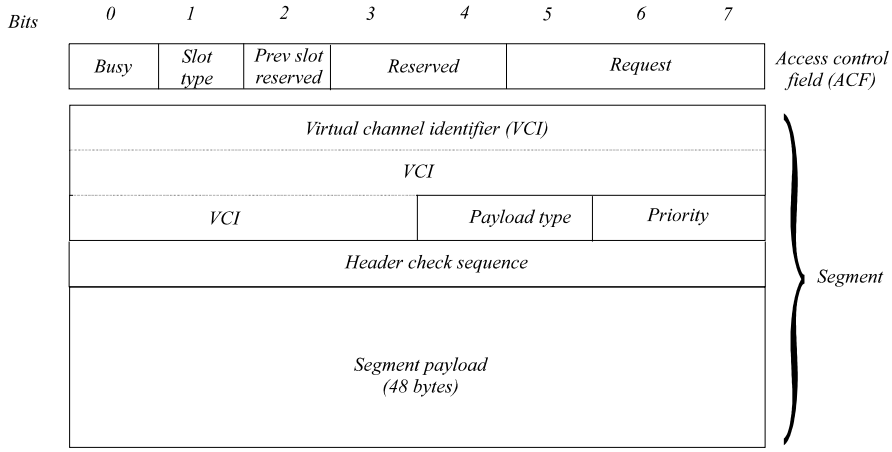


Figure 8.33 Slot and segment format in DQDB

The *virtual channel identifier* (*VCI*) is used by the connection-oriented QA service and the isochronous PA service, to identify a connection, but is set to $FFFFFF_{16}$ for connectionless QA. All user data services use a *payload type* of "00". Other values of this field, and the *priority* field, are reserved for future expansion of the standard.

Access to QA services is via an algorithm called the *distributed queue access protocol* (*DQAP*), which gives the standard its name, and which we now describe. Suppose that node *M* wishes to transmit to node *N* downstream of it on bus *A*. *M* must first request permission to use a forthcoming slot on bus *A*, which it does by using a slot going the other way. To this end, it sets one of three *request bits* in such a slot, representing one of three possible priorities which the segment it wants to send may take. Priority class 2 is the highest and is used only for network management. Class 0, is the normal priority and class 1 is used for particularly delay sensitive information, which nonetheless does not merit use of a PA connection.

All stations watch the buses for slots passing with a request bit set. Each node maintains three *request counters* (*RC0*, *RC1* and *RC2*), and three *countdown counters* (*CD0*, *CD1*, and *CD2*) for transmission on bus A (another six counters are maintained for bus *B*). All counters are initialised to zero. For a node which has nothing to transmit itself, when a request of priority *k* appears on bus *B*, counters *RCi*, for $i \leq k$, are all incremented. When an empty QA slot on bus *A* passes, all counters are decremented, unless they are already zero. In this way, counter *RCi* records the number of outstanding requests for transmission of segments of priority *i* or greater by all stations downstream on bus *A*. As soon as a station has a segment of its own to transmit, of priority *k*, say, it transfers the contents of *RCk* to *CDk*, and sets *RCk* to zero. When an empty slot passes, *CDk* is now decremented instead of *RCk*. *CDk* is incremented if a request of *higher* priority is observed, while *RCk* is incremented only if a request of the same priority passes. When the counter *CDk* reaches zero, all outstanding requests downstream of higher priority, or of the same priority which arrived before the local one, have been transmitted, and the queued segment is sent in the next empty QA slot on bus *A*.

The operation is complex, and is probably most easily understood via an example.

Example 8.1

Consider the DQDB system shown in Figure 8.34. Assume that all counters at all stations are initially set to 0. Then the following sequence occurs:

(a) A Priority 1 segment queues at Station 3 to be sent to Station 4 on the *A* bus. Station 3 transfers its *RC1* counter to *CD1,* resets *RC1* (which of course is, strictly, unnecessary, since it is already 0) and sends a Priority 1 request along the *B* bus. In principle, since its *CD1* counter is 0, Station 3 can use the next empty QA slot on the *A* bus.

(b) Stations 1 and 2 see the Priority 1 request and increment counters *RC0* to *RC1* by 1. However, before an empty QA slot arrives, a Priority 2 segment arrives at Station 4, for delivery to Station 5 (not shown). Station 4 transfers RC2 to CD2 and sends a Priority 2 request.

(c) Stations 1 and 2 see the Priority 2 request and increment counters *RC0* to *RC2* by 1. Station 3, which has active *CD1*, increments *RC0*, *RC2* and *CD1*. It can no longer use the next empty slot, which now rightfully belongs to Station 4.

(d) An empty slot is sent down the *A* bus. Stations 1 and 2 decrement all *RC* counters by 1. Station 3 decrements *RC0*, *RC2* and *CD1*. Station 3 now has active *CD1* = 0, and now again has the right to the next empty slot. The current empty slot is used by Station 4 to transmit its data.

(e) At last another empty slot appears. Stations 1 and 2 decrement all *RC* counters by 1. Station 3 uses the slot to send its data.

A little study should convince you that this algorithm simulates a system of three prioritised FIFO queues distributed amongst the stations. There remains the question of what happens if a second segment is queued for transmission at a node while a previous segment of the same priority is still awaiting transmission. The DQDB protocol will not allow a station to issue a second request while another request of the same priority remains unsatisfied (i.e. an empty QA slot that could be used to transmit the first queued segment has not yet arrived). If this restriction were not imposed, a station, *S*, near the headend of say bus *B*, with a lot of data to send on bus *A*, could flood bus *B* with requests. Since a slot with a request bit set cannot be used as a request by any other station, stations upstream from *S* on bus *A* may never get the chance to submit requests at all, since all slots passing them on bus *B* are already carrying requests inserted by *S*.

Isochronous data is treated differently. Here there is no distributed arbitration of the type just discussed. Instead the station carrying out the headend function must allocate a VCI on request to a station wishing to transmit via an isochronous connection. The headend must then generate a regular stream of PA slots with the VCI number included, which the requesting station may use. Unlike QA slots, the VCI number is considered part of the access control field in a PA slot, and is supplied by the headend. In fact, a connection may use all the data capacity of the segments with its VCI for itself, or may be required to share (if for example, it does not need that

much bandwidth). If PA segments are shared, the *PA function* at the station will be told not only what VCI it should expect, but also what offsets it may use into the slot payload. Other connections, possibly belonging to other stations, may then exist with the same VCI but different offsets.

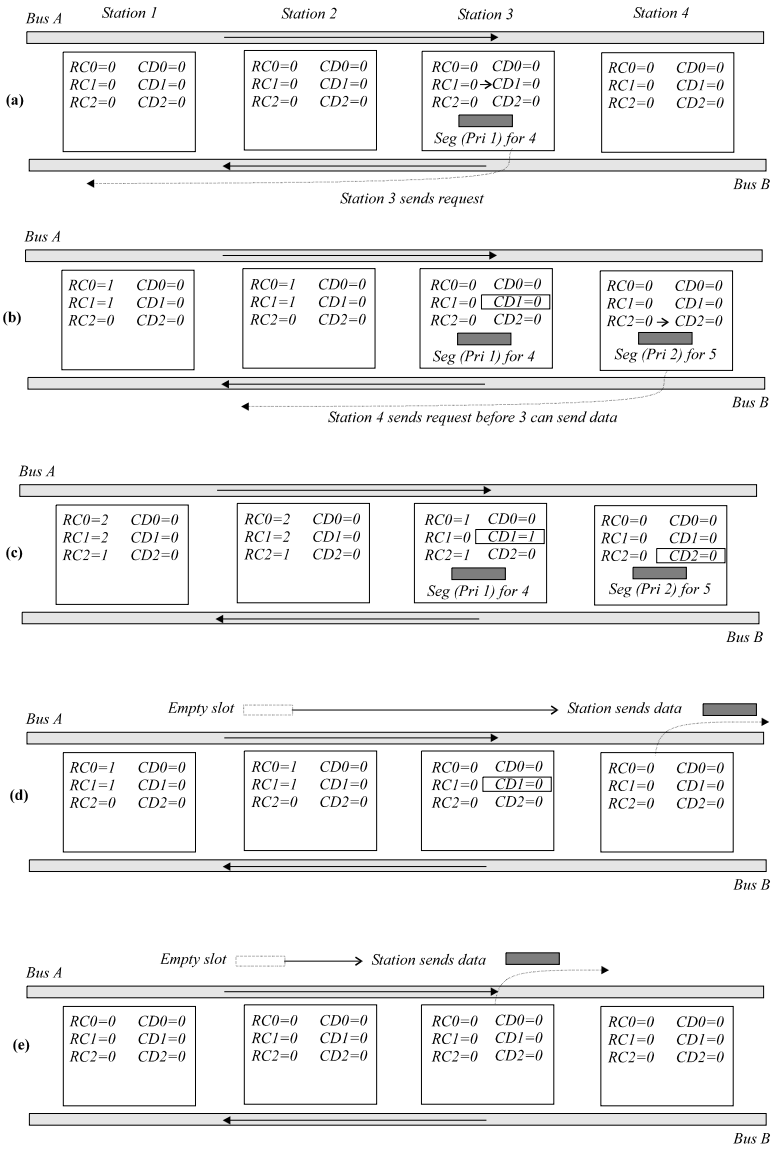


Figure 8.34 Example of distributed queue access protocol

DQDB is a complex MAC access procedure (see Figure 8.35), and its success to date has been limited. Its advantage is that its slot structure is compatible with ATM

(in the I-series sense), and also that it can be used over a variety of physical media, including optic fibre carrying SONET or SDH. In such situations the DQDB slots are simply carried as data in the SONET or SDH frames, in exactly the same way as ATM cells are.

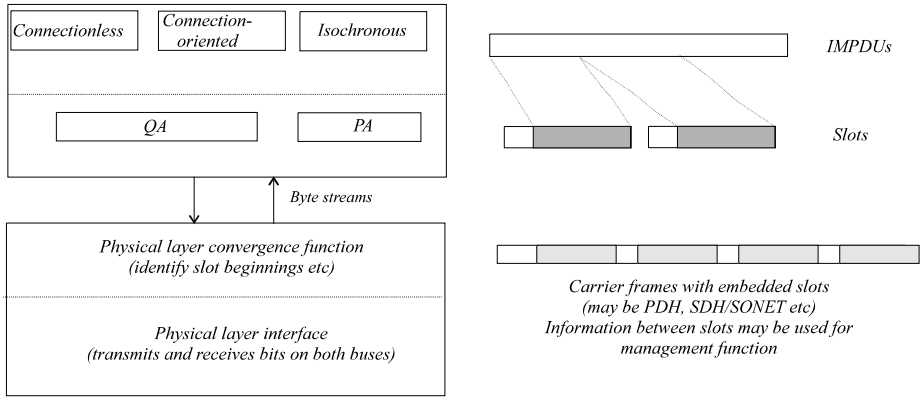


Figure 8.35 Layering structure of DQDB

The DQDB MAC structure should be thought of as divided into two "sublayers": one reponsible for the QA or PA protocols, the other supplying the convergence for either connectionless, connection-oriented or isochronous services. The connectionless MAC convergence function, for example, actually has two types of PDU defined. The so-called *initial MAC PDUs* or *IMPDUs*, can be up to 9188 bytes in length, reflecting the structure of MAC SDUs, which are supposed to be capable of accommodating MAC units from any other type of (802) network. These IMPDUs are immediately broken down into 48-byte *derived MAC PDUs* (*DMPDUs*), which will form the data in the segments. The PDUs of the QA "sublayer", however, are the 53 byte slots with their segment payloads. The physical layer can be thought of as comprising a convergence function and a physical interface. The physical interface simply transmits and retrieves bits from the two buses, while the convergence function identifies the structure of the underlying carrier frames, extracts the slots, and streams the bytes up to the MAC layer. As we have seen, DQDB may use underlying carriers of many different types (PDH, SDH, etc), and the convergence function hides the details of their differences from the MAC layer, which sees only a stream of bytes identifiable as empty or full slots.

8.3.4 Switched Multimegabit Data Service

The demand for remote LAN interconnection is so great that it was inevitable that public carriers would wish to provide such connectivity as a chargeable service. MANs and, in particular DQDB MANs, provide a natural vehicle for this, but, in fact, there is no need to tie the service to any specific technology. What is required is an interface standard and associated quality of service specifications which any user can employ to access the inter-LAN network, and just such a standard, the *switched multimegabit data service* or *SMDS*, has been developed by Bellcore. The SMDS specification was certainly developed with DQDB in mind as the likely underlying

technology, but it may, in fact, be provided, not by a broadcast dual-bus MAN, but rather by a switched network based on *SMDS switches*, which support the MAC sublayer protocols of 802.6. A parallel can be drawn with switched Ethernet, where a switching fabric mimics the operation of a standard 802.3 cable network.

The point is that each client LAN connects via a single point-to-point access leased line to the so-called *SMDS cloud*, which may be implemented either as a broadcast or a switched fabric (Figure 8.36). SMDS, like X.25, is really an access standard, albeit one based in conception on a particular connectionless MAN technology.

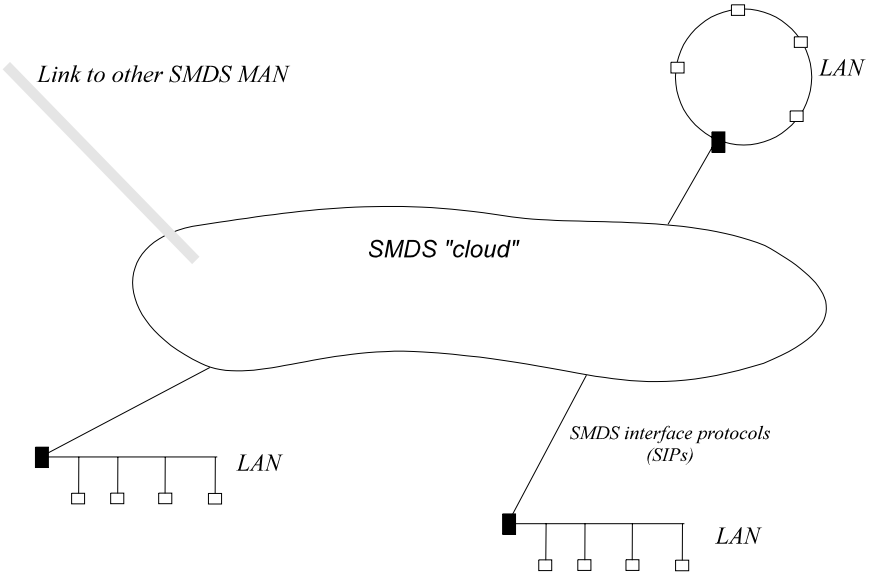


Figure 8.36 SMDS connection of LANs

The interface is a three-layer mini-stack. *SIP* (*SMDS interface protocol*) *Level 1* corresponds to the physical layer convergence functions of 802.6, *SIP Level 2* to the MAC sublayer using QA access, *SIP Level 3* to the MAC connectionless convergence functions. The PDU formats are identical to the DQDB connectionless service. The idea is that a LAN attached to an SMDS network will include a bridge that talks SIP on one side and the LAN protocol on the other. SIP can operate over several types of physical connection to the SMDS network, including lines running at speeds from 1.5Mbps to 45Mbps (supporting token ring, Ethernet, ATM LANs and US PDH lines). Further support is being added for the SONET hierarchy. As can be deduced from the DQDB protocols that it supports, SMDS is intended to support bursty inter-LAN data traffic, not isochronous data streams.

SMDS service is now widely offered by carriers in the US and Europe. The European Telecommunications Standards Institute, *ETSI*, has adopted a service called *connectionless broadband data service* (*CBDS*) which contains SMDS as a subset. In fact, the SMDS and CBDS are so close in nature that the names are used almost interchangeably, although CBDS uses the European PDH hierarchy at rates up to

34Mbps (E3). The British Telecom version, for example, although strictly an implementation of CBDS, is marketed as SMDS and is now extensively used as the backbone of the high-performance *SuperJanet* academic network in the UK.

It has been argued that SMDS and CBDS are really just precursors of the much more general concept of *broadband ISDN*, offering the types of volume data transfer services which the latter is ultimately intended to bring in a more general and widely available manner. It is certainly true that the SMDS technologies are not receiving the development attention of, for example, ATM, which is widely predicted ultimately to replace them. Perhaps the safest approach at present, is to consider these public MAN services as precursors of broadband ISDN, occupying a temporary gap in the match between customer needs and the maturity of the latter as a practical technological enterprise.

8.3.5 High Speed LANs

As hardware has become inexorably faster, desktop machines and workstations have continued to increase in performance, and, inevitably, the applications which can now be run on these systems have data transfer requirements far in excess of those envisaged when the first generation of LANs were developed. High data transfer requirements have long been an issue in supercomputer installations, and special purpose, but limited and expensive solutions, have been developed. These include the parallel-bus based *HiPPI* (*high-performance parallel interface*) which uses parallel 50-way STP or multimode fibre-optic cable to yield data rates of 800Mbps or 1600Mbps between stations. HiPPI is a crossbar switch-based connection-oriented network, and was intended for specialist applications involving the connection of supercomputers and fast peripherals. A similar undertaking, considerably more complex, is *Fibre Channel*, with defined speeds up to 3200Mbps.

HiPPI and Fibre Channel are high-end solutions, and are hardly suitable for general use. It is becoming increasingly obvious that the mass market as well will soon no longer be satisfied with 10Mbps data rates, and, while the performance of these exotic interconnects is perhaps not yet necessary at the desktop, an order-of-magnitude improvement (100Mbps) is being seriously sought. It now seems that this improvement is most likely to be delivered to the average user by LANs based on a development of 802.3, called *Fast Ethernet*. Before examining this technology it is perhaps worth glancing at a couple of solutions which have been suggested as possible alternatives, namely FDDI and ATM.

FDDI has been around for several years longer than the 100Mbps Ethernet, and was initially considered a possible candidate for the next generation of LANs. It has, however, suffered from the still high cost of optical technology and the complexity of its interfaces, and, as we have discussed, has found its niche more in the backbone and MAN application areas than in LANs. ATM, on the other hand is a technology under rapid development. Already adopted as a vehicle for carrying data traffic over long-haul SONET and SDH links, there has been much speculation about its widespread take-up in LANs and MANs. However, despite its technical advantages, it has, like so many international standards, taken time to agree, and it remains a relatively high-cost solution for the desktop. This and the perception that data transfer rates of 100Mbps are needed now, has given the cheaper, less technologically advanced Fast Ethernet the opening required to bid for control of this fledgling market.

The desirability of a fast replacement for Ethernet, the most popular LAN technology, was apparent by 1992, and IEEE began to investigate possibilities. In fact, the work spawned two alternatives, *802.3u*, which is now universally known as *Fast Ethernet*, and *802.12*, so-called *100VG-AnyLAN*, which, although theoretically superior, is less compatible with the existing 802.3, and has subsequently proved much less popular.

The goal of 802.3u (and 802.12), was a 100Mbps data rate. It can be seen from Equations 8.11 and 8.12 that, in order to maintain throughput, any increase in the data rate, R, must be compensated by either increasing the packet size, L, or restricting the maximum propagation delay. The former option is undesirable since using different packet lengths would compromise compatibility with standard Ethernet. In most cases therefore, it is preferable to restrict the network size, in most cases to 100m maximum cable runs. For this reason, and the fact that it is more difficult to engineer the electrical characteristics of multidrop than point-to-point cables, at high signalling rates, 802.3u specifies only a hub implementation, *100Base-T*, and has no equivalent of 10Base5 or 10Base2. Another motivation for reducing network size is to allow the minimum packet size to remain unchanged. Recall that the minimum packet size in standard Ethernet is chosen so that any collision will always be detected before transmission is complete. If the transmission rate is a factor of ten faster, the cross-network propagation delay must be reduced by a factor of ten to compensate. Note, however, that these restrictions do not apply to switched full-duplex Ethernet, where collisions as such do not physically occur.

In 100Base-T, stations are attached to a central hub or switch using twisted pair or fibre optic point-to-point links. Ideally, in a twisted pair implementation, two high-grade Category 5 UTP cables would be used, one for each direction, and this version is called *100Base-TX*. For the same reasons as with FDDI, a variant of 4B/5B signalling is used in preference to Manchester, allowing the system to operate at 125Mbaud, which, over 100m, is well within the specification of Category 5 UTP. The variant is called *MLT-3*, and basically involves an extra scrambling and coding stage that gives the stream a more amenable PSD for transmission via an electrical signal.

However, many modern office buildings are already wired with Category 3 UTP, to allow easy installation of telephone extensions[2]. To permit the use of this widely available and more cost-effective wiring, another variant of 802.3u called *100Base-T4*, is also defined. Since Category 3 UTP cannot carry signals of 125Mbaud over the requisite distances, a T4 connection consists of 4 twisted pairs which operate at 25Mbaud, but use ternary signalling, with 3 signal levels, 0, $+V$ and $-V$. The coding system, called *8B6T*, maps 8 bits onto 6 ternary signalling elements, which means that at 25Mbaud, a line can carry:

$$25 \times \frac{8}{6} = 33.3 \text{ Mbps.}$$

[2] Note that local loop and domestic telephone twisted pair is not typically of Category 3 quality.

This means that 3 pairs, operating in parallel, can transfer at 100Mbps as required. This is, in fact, exactly how 100Base-T4 works, as shown in Figure 8.37. When the station is receiving, three of the four pairs, 2, 3 and 4, are used to obtain the 100Mbps rate. When the station is transmitting, pairs 3 and 4 turn around and together with pair 1, again yield the requisite rate: pair 2 continues to be used by the hub, but now acts as the collision detect signal to the station.

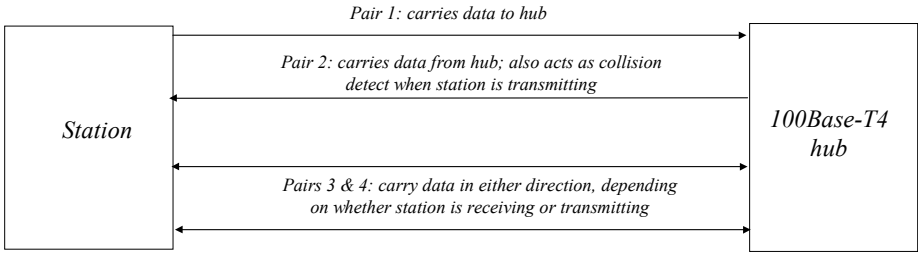


Figure 8.37 Operation of 100Base-T4 Fast Ethernet

Note that full-duplex operation is possible with 100Base-TX, but inherently not with 100Base-T4. Of course full duplex operation requires the use of a 100Base-T switch, as opposed to a simple hub, which merely mimics the operation of a single shared bus.

Another implementation is *100Base-FX*, which uses two multimode optic fibres, and operates like 100Base-TX, but the stations can be farther away from the hub: half-duplex installations permit 450m, full-duplex installations, which are serviced by switches and do not suffer collisions in the normal sense, up to 2km.

The general operation of Fast Ethernet is similar to that of its lesser analogue, although, in principle at least, it operates ten times faster. For example, in 802.3, frames must be separated by an inter-frame gap of 9.6μs; in 802.3u, this is reduced to 960ns. The whole point of 100Base-T is to make migration from 10Base-T as easy as possible, so most 100Base-T hubs and switches can handle a mix of stations operating at both 10 and 100Mbps. Likewise, most 100Base-T station adapters (Ethernet cards) can be connected to a 10Base-T or 100Base-T port, and will adjust themselves to the correct speed automatically.

The strength of 100Base-T is that it allows an easy upgrade path from 10Base-T, but the downside is that it does suffer from much the same shortcomings as its predecessor. The promoters of the less successful 802.12 standard, including IBM, HP and AT&T have abandoned the CSMA/CD approach in favour of a MAC scheme called *demand priority*, based on a round-robin algorithm. Advantages include support for message priorities and isochronous transmission, as well as an ability to internetwork easily with token ring as well as Ethernet LANs.

However, 100Base-T is by no means the last gasp of the Ethernet paradigm. The industry has well-developed plans to take Ethernet performance forward by another order of magnitude. In May 1996 a consortium of 11 companies formed the *Gigabit Ethernet Alliance* (*GEA*), a grouping whose avowed aim was the development of standards for a 1000Mbps Ethernet. In July 1997 IEEE issued a draft standard (*IEEE 802.3z*) defining this *Gigabit Ethernet,* and the future of the technology now seems

assured. 802.3z, known as *1000Base-X*, is a hub or switch-based technology which uses a fibre channel physical layer, and can employ three types of media: an 850nm laser over multi-mode fibre in *1000Base-SX*, over a distance of up to 550m; a 1300nm laser over single-mode or multi-mode fibre in *1000Base-LX* up to 3km; and STP in *1000Base-CX* up to 25m. The line code used is *8B/*10B, also used in Fibre Channel and related to the 4B/5B used in FDDI. A related standard, *IEEE 802.3ab*, still in development, defines *1000Base-T*, for use over Category 5 UTP over 4 pairs up to 100m long.

The MAC layer in Gigabit Ethernet is just as in standard and fast Ethernet. Although the logical minimum frame length of 64 bytes is maintained, actual physical frames are always padded out to a real minimum of 512 bytes to allow longer cable distances to be supported on non-switch (hub) implementations. This is called *carrier extension*. However, if a large number of small frames were sent, this would result in a large overhead greatly reducing performance. To avoid this, a technique known as *packet bursting* has been introduced. This allows a source with a number of small packets to send, to concatenate them (with a small inter-packet gap) up to a maximum length of 1500 bytes

In earlier Ethernet technologies hubs are used as, essentially, extensions of a cable bus and can support only half-duplex connections to stations, whereas switches, which are really packet-switching nodes, support full-duplex access. However, switches are more expensive than hubs, and, due to the technology required at gigabit speeds, this difference is likely to be particularly pronounced in Gigabit Ethernet. As a compromise, a new device called a *buffered distributor* has been introduced, which has the repeating characteristics of a hub, but includes input and output FIFOs at each port allowing simultaneous data transfers to and from a station (Figure 8.38). Also collisions no longer occur on links so cable length restrictions are removed, clearly essential for 1000Base-SX and 1000Base-LX.

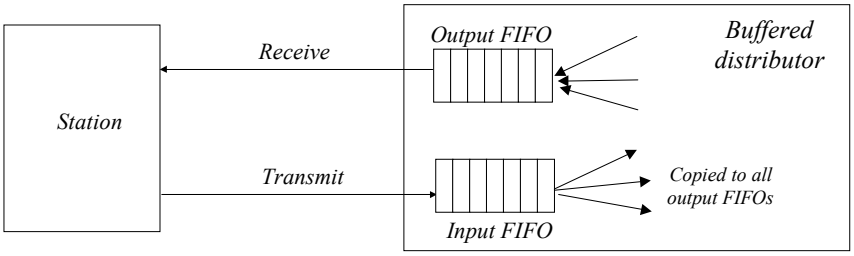


Figure 8.38 Architecture of a buffered distributor

A simple flow control mechanism is defined by the associated standard *IEEE 802.3x* to prevent a station from overrunning the FIFOs in the distributor (this is much more likely at the high data rates of Gigabit Ethernet).

There is no doubt that Gigabit Ethernet technology is a strong contender for the high-performance LANs of the future and many observers believe that it is destined to compete with ATMin this regard. As with 100Base-T, the great touted advantage is that, as an extended implementation of 802.3, Gigabit Ethernet can easily coexist with earlier forms of Ethernet. Ethernet does not, however, have the sophisticated support

for different qualities of service that ATM provides, although efforts are being made to graft on capabilities, notably via *IEEE 802.1p* and *802.1q*, which add frame priorities and VLAN support to the Ethernet standard. If these efforts are successful then even the faster 622Mbps ATM technologies may find that they have a struggle on their hands, at least for high-speed LAN applications (see also the discussion in Section **Error! Reference source not found.**).

8.4 WIRELESS NETWORKS

The networks discussed so far use guided media, such as wires or optic fibres. There are, however, alternatives to this, in the form of directed but unguided microwave or laser transmission, and free-space broadcasts of radio or infrared waves. Directed transmission is essentially point-to-point and, if tightly focused, behaves, at all but the physical level, as would a cable or fibre. However, when free space broadcasts are used, a single transmission can bathe a number of receivers, and the issues that arise are those of broadcast networks, albeit with some interesting additional twists.

There are differences in character between systems which use radio and those that use infrared. In particular, like visible light, infrared signals cannot penetrate solid objects and therefore systems communicating by such means must be in the same room. Infrared signals can also be adversely affected by sunlight or other infrared sources and so are often not suitable for use outdoors. Infrared transmission is either by LED, which can be modulated at up to perhaps 20MHz, or laser diode, capable of several hundred MHz. The limited range of infrared transmission, however, means that one can relatively easily control the interaction of the systems participating, since all must, in any case, be located in the same room. There is no question, for example, of infrared transmission in one room interfering with communication in another. Within a room, infrared signals can be directed, just like optical lasers, to provide point-to-point links between say two computers or a computer and a printer. Alternatively, they can be diffused to broadcast across the entire space, at some cost in data rates due to dispersive ISI, although the use of a reflecting ceiling dome called a *satellite* can be used to provide active relaying of directed beams to all targets.

Radio waves have much greater range and can pass through walls and other solid objects, although their progress is impaired, particularly by conducting media. They may be used over relatively short ranges (so-called *wireless LANs*) or for more extensive networks, using transmitters with higher power—we have already seen, in Section 8.1.1, that the first packet radio network was used to connect hosts spread across the Hawaiian archipelago. Radio signals suffer so-called *path losses* that, after a certain range, makes it impossible for reliable reception. For signals in the VHF band, which must be used to obtain reasonable data rates, this range is only a little better than line-of-sight even for high-powered transmitters. The result is that, in a radio network, some systems may be out of range of others, so that the medium is not truly a “broadcast medium” in the sense we have previously used the term.

For example, in Figure 8.39, we can see that, because of range limitations, two stations, *A* and *C*, can both receive signals from a third, *B*, but not from each other. This can present problems if all stations are in the same network, because, clearly, *A* and *C* cannot talk to each other directly, and must use an intermediate station, such as

B to relay any messages. The inaccessibility of *A* from *C*, and vice versa, also rules out the use of broadcast-based MAC algorithms, which rely on all stations being able to listen to each other's transmissions. Suppose *A* and *C* both wish to transmit to *B*, using the same carrier frequency. Then, while mechanisms like ALOHA will work, CSMA techniques on their own will not, since *A* cannot know that *C* is already transmitting, nor can it detect a collision until an ACK fails to appear.

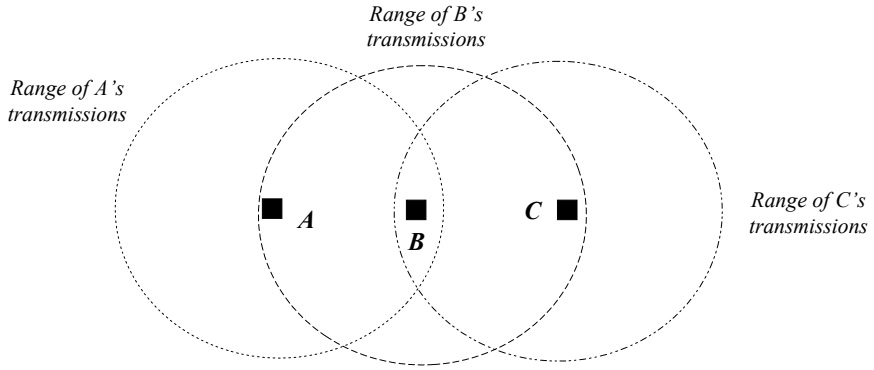


Figure 8.39 Effect of range limitations in radio networks

One way round this is to require a potential transmitter to send a *request to send* (*RTS*) message to a potential message recipient. The target then replies with either a positive or negative acknowledgement (*clear to send* or *receiver busy*). In the former case, the originator can send its data at once, while in the latter, it must defer, and presumably retry later. This mechanism can be used with CSMA/CD or CSMA/CA both of which still have utility in such situations, because they reduce the chance of two stations that are within range of each other from sending RTS messages simultaneously, and so generating a collision. The technique is called the *distributed foundation wireless MAC protocol* (*DFW MAC*).

DFW MAC works in situations such as that above, where stations use their own fixed power transmitters to send to each other directly. These are sometimes known as *peer wireless networks*, to distinguish them from networks that use hub relays to pick up transmissions from stations and then route them by more conventional means. Networks of the latter type are effectively organised into adjacent areas called *cells*, each of which contains its own hub. Within a cell all transmitters are within range of the hub, and talk directly to it. It then relays messages by radio if the destination is in its own cell, or over a master inter-hub network, if the destination is not. Obviously, if stations can all communicate with the local hub, their transmissions can also leak into adjacent cells, and so neighbouring cells must never use the same transmission frequencies. However, cells that are suitably separated may use the same frequencies, which is important since the amount of available radio bandwidth is extremely limited.

Using three separate groups of transmission frequencies and cells that are hexagonal in shape, full area coverage is possible without adjacent cells ever using the same groups. This is shown in Figure 8.40 where the circles indicate the nominal range of the hub transmitters, and the hexagons the formal boundaries of the cells.

This frequency group pattern is called *three-repeat*. In effect, each cell is a separate radio network, but they do not interfere because they are not using the same bands. Of course, each station must identify itself to its cell, so that the hub knows of its presence. Further, a system of this type can cope with mobile hosts, so long as responsibility can be gracefully passed from one hub to another as the host crosses a cell boundary.

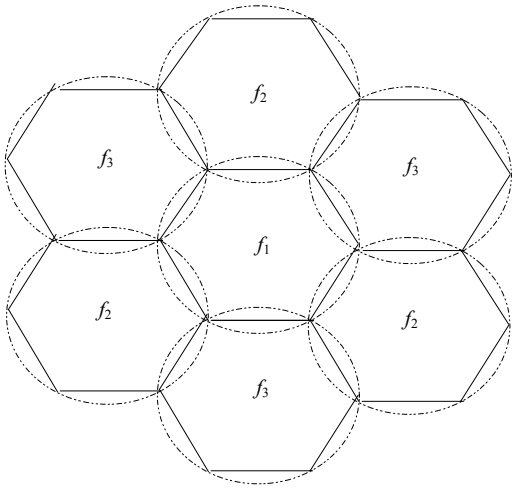


Figure 8.40 Reuse of frequencies in nearby cells in three-repeat pattern

This is exactly the principle on which mobile or *cellular* phone systems operate. The *advanced mobile phone system* (*AMPS*) used in the US, and the *TACS* system used in the UK use cells about 10km across but with a *seven-repeat* pattern which ensures that the same frequencies are not used by any cells which have less than two intervening cells between them. Within a cell, these systems use FDM to allow many mobile phone users to communicate simultaneously and the transmissions are purely analogue. However, more recent cell phone technologies, such as *global system for mobile communications* (*GSM*), use digital transmission with a combination of TDM and FDM within cells.

The cellular telephone systems are of course WANs, but similar principles can be used to establish restricted area wireless LANs, although in most circumstances cells need not be so densely packed here as in a phone network (the LAN may only be required to work in certain rooms or buildings). Whether a wireless LAN is peer-to-peer or whether it is cellular, stations must use FDMA or TDMA to control access to the available frequencies. Typically fixed frequency bands are allocated dynamically on a TDMA basis using a MAC algorithm like CSMA/CA or CSMA/CD with DFW MAC. In any case, a transmitter will send its data using modulation with an allocated carrier frequency. One problem that arises is that radio frequencies are in great demand for other uses, including special mobile communications for use by police, emergency services, taxis, car alarms, CB-radio, and even microwave ovens for some high frequency carriers. The result is that, even if good design principles are observed,

wireless networks may find transmissions subject to interference from these external sources. However, the military have been users of radio systems for data communications for longer than the commercial world, and have, if anything, an even greater need for reliability. Military users have widely adopted a powerful technique called *spread spectrum transmission*, which has the ability to reject such interference much more effectively than standard modulation techniques. Spread spectrum, and an associated MAC system called *code division multiple access* (*CDMA*) are now being adopted for use in commercial wireless networks, and are described below.

Essentially there are two spread spectrum techniques: *direct sequence*, and *frequency hopping*. Frequency hopping, the older of the two, relies on splitting an available frequency band into n equal sub-bands, or *channels*, which we number, 1, 2, …, n. Now the source generates a pseudo-random sequence of numbers drawn from the set $\{1, 2, \ldots n\}$. There are a number of well-known algorithms, which, with an initial *seed value*, produce such a sequence of values, which, although deterministic, actually appear random if the seed is not known. Let the sequence computed be (n_1, n_2, n_3, …), with $n_1, n_2, n_3, \ldots \in \{1, \ldots, n\}$. A time unit called the *chip period*, ΔT, is defined, and the transmitter sends in sub-band n_1 for the first chip period, in sub-band n_2 for the second and so on. The sequence (n_1, n_2, n_3, …) is called the *hopping sequence*, and $\frac{1}{\Delta T}$ is the *chipping rate*. The procedure is illustrated for a system with eight channels in Figure 8.41.

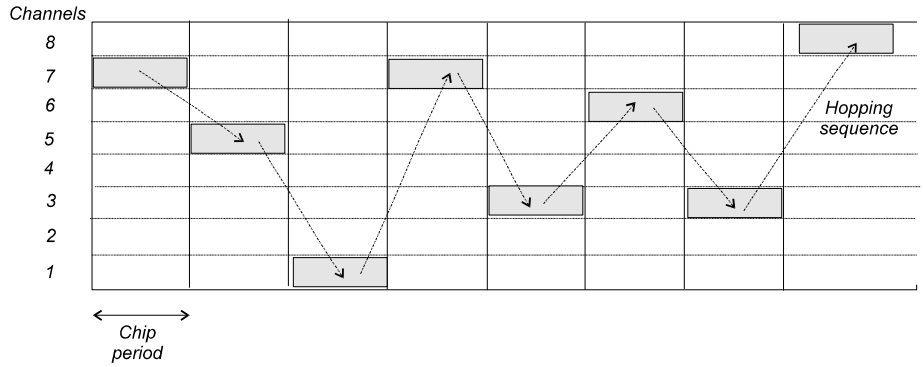


Figure 8.41 Frequency hopping spread spectrum

Frequency hopping can be used with analogue or digital data. With digital data, the chipping rate may be higher than the baud rate, in which case the mechanism is called *fast frequency hopping*, or slower, in which case we have *slow frequency hopping*. In fast frequency hopping each signal element is actually sent multiple times at different frequencies, while in slow frequency hopping a number of bits are sent in each chip period. In either case the receiver must also know the random sequence and must be in sync with the chipping rate of the source.

Frequency hopping has two important advantages. First of all in situations where some, but not all of the channels are affected by interference, the effect is greatly lessened. With slow hopping this works best if the transmitter excludes the affected

channels from its hopping sequence. With fast hopping, this is not necessary, since every bit gets transmitted multiply on many channels; however, fast hopping is, unsurprisingly, more difficult and therefore expensive, to implement. Secondly, the message is extremely difficult to eavesdrop unless the pseudo-random sequence is known, since all that can be heard receiving on any channel is a series of apparently disconnected blips.

Direct sequence spread spectrum is similar in nature to frequency hopping but operates in the time domain on a binary digital data stream. A pseudo-random sequence of bits is produced, and used to construct a pseudo-random bit-stream, called the *spreading sequence*, with a bit time, the *chipping period*, which is some fraction of the bit time of the original data stream. The data and pseudo-random streams are then exclusive-ORed together, as illustrated in Figure 8.42, and the result, a so-called *chipping code* is then transmitted, after modulation, so that each data bit is actually sent as a group of *n* code bits or *chips*. The number of chips per data bit is called the *spreading factor*. The chipping sequence is most commonly modulated using multilevel PSK.

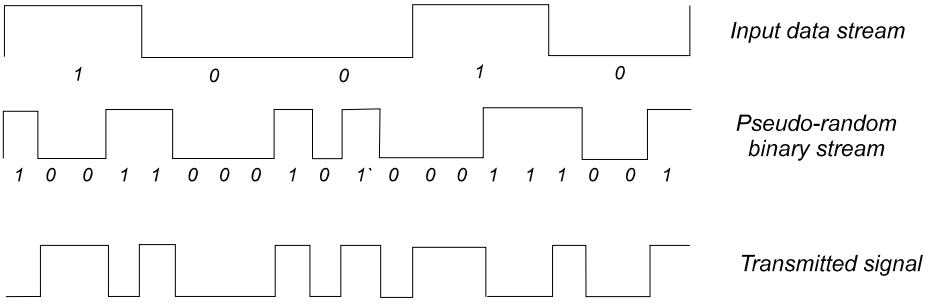


Figure 8.42 Direct sequence spread spectrum

The transmitted signal can be recovered at the receiver by exclusive-ORing it again with the spreading sequence. Without the spreading sequence, recovery of the data is virtually impossible. Again, even if some of the chips in a data bit are corrupted by noise, the data bit can usually still be reconstructed, since it is composed of multiple chips, so the scheme, like frequency hopping, provides some resistance to corruption.

Both spread spectrum techniques increase the bandwidth used by the basic signal, trading this off against improved noise immunity and security. However, in a wireless network, if all stations are using the same band, a TDMA MAC technique must be applied. This can be done using any conventional TDMA approaches, but there is in addition another method, CDMA, which relies on the properties of spread spectrum transmission, and can, in addition, recover some of the extra bandwidth surrendered by the spreading process.

CDMA is based on direct sequence spread spectrum. The technique is relatively recent and quite complex in practice, so the description given here is simplified. Each data bit, as before, is divided into a fixed number of chips, *n*, typically 64 or 128. All stations are precisely synchronised, so that the chip periods are universal across the network. For reasons which will become apparent, the signal elements used for "1"

and "0" are $+A$ and $-A$, respectively, for some characteristic basic pulse amplitude, A. However, instead of a pseudo-random sequence, the spreading sequence for a given station is periodic, with a given pattern of n chips repeated for each data bit. This *chip sequence* is unique for each station, and can be represented by an $n\times1$ vector, say $\mathbf{s}_i$ for station i, with elements +1 and -1 (the A can be omitted by appropriate choice of units). Since $|s_\mathbf{i}| = \sqrt{n}$, the derived vectors:

$$\mathbf{e}_\mathbf{i} = \frac{\mathbf{s}_\mathbf{i}}{\sqrt{n}}$$

are normalised (length 1).

When a station wishes to send a "1", it transmits its chip sequence, $\mathbf{s}_\mathbf{i}$; when it wishes to send a "0" it transmits $-\mathbf{s}_\mathbf{i}$. When several stations transmit simultaneously, the chip sequences from the transmitting stations superpose, creating an n-bit chip sequence for the medium, which can also be represented as an $n\times1$ dimensional vector $\mathbf{c}$, where

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ . \\ . \\ c_n \end{bmatrix} \qquad \text{with } c_\mathrm{i} \in \{-n, -(n\text{-}1),\ldots, (n\text{-}1), n\}$$

Assuming that all stations operate on the same data bit boundaries, $\mathbf{c}$ represents a superposition of precisely one data bit transmitted from each active station. In fact:

$$\mathbf{c} = \sum_i a_i \mathbf{s}_i = \sqrt{n} \sum_i a_i \mathbf{e}_i \qquad 8.14$$

where $a_i \in \{-1, 1\}$, with $a_\mathrm{i} = 1$ for a station transmitting a "1" and $a_\mathrm{i} = -1$ for a station transmitting a "0". The set of all possible values of $\mathbf{c}$, forms a subset of the n-dimensional vector space over $\mathbf{R}$, which we represent as $\mathbf{R}_\mathrm{n}$.

The key is to select chip sequences, $\mathbf{s}_\mathbf{i}$, which are orthogonal to each other in the sense that:

$$\mathbf{s}_\mathbf{i} \cdot \mathbf{s}_\mathbf{j} = \mathbf{0} \qquad \forall\, i, j$$

In this case the $\mathbf{e}_\mathbf{i}$ can be extended to an orthonormal basis of $\mathbf{R}_\mathrm{n}$. Recall from linear algebra that a basis for an n-dimensional space must have exactly n elements, and that any element of the space can be uniquely represented as a linear combination of basis elements. Now any station can see that the superposed signal in the medium is $\mathbf{c}$, and can pick out the bit transmitted from station k, say, by taking:

$$\frac{1}{\sqrt{n}} \mathbf{c} \cdot \mathbf{e}_k = \frac{1}{n} \mathbf{c} \cdot s_k = a_k$$

It follows that, if the chip sequences are known, the data transmitted by any station can be recovered. Although the technique is actually more complex than this to implement in practice, because, for example, synchronisation of all stations is usually not possible in the manner described, the essence is as described.

CDMA works best where a receiver obtains roughly equal power transmission from all transmitters. This is often the case where a single hub station receives and transmits to many relatively equi-distant clients. This is the case, for example in cellphone and satellite networks (see Section 8.5). It does not provide any greater bandwidth than equivalent FDM or TDM, and is in general much more difficult to implement, but it has the spread spectrum advantages of noise immunity. A variant

Example 8.2

If three stations transmit using an 8-chip code with chip sequences

Station 1 $\mathbf{s}_1$ = (1, -1, 1, 1, -1, -1, 1, 1)

Station 2 $\mathbf{s}_2$ = (1, 1, 1, -1, 1, -1, 1, -1)

Station 3 $\mathbf{s}_3$ = (-1, 1, 1, 1, -1, 1, 1, -1)

It is easy to check that these chip sequences are orthogonal.

Now suppose Station 1 sends a “1”, Station 2 sends a “0”, and Station 3 sends a “1”. It follows that:

$$\mathbf{c} = \mathbf{s}_1 - \mathbf{s}_2 + \mathbf{s}_3 = (-1, -1, 1, 3, -3, 1, 1, 1).$$

To retrieve the bit sent by, say, Station 2, form

$$\frac{1}{8}\mathbf{c} \cdot s_2 = a_k = \frac{1}{8}(-1-1+1-3-3-1+1-1)$$
$$= -1$$

which indicates a “0” as required. For Station 1 on the other hand:

can also be used in frequency hopping, where individual transmitters use mutually exclusive hop sequences. Note that, as described, however, CDMA does not include the same randomness as the individual spread spectrum techniques, since all stations must be aware of each other’s sequences.

8.5 SATELLITE NETWORKS

A communications satellite is essentially a relay that permits ground based stations using microwave signalling to communicate with each other beyond the normal terrestrial range of such frequencies. Most, but not all, communication satellites are placed in a *geostationary orbit*, in which they always appear in a fixed position relative to ground stations. Such an orbit must be equatorial with a period of exactly one day, and to achieve this, requires an orbital radius of approximately 36000km. Satellites using the same frequencies for communication in a geostationary orbit must

not be too close together in the sky, a separation of about 2° being sufficient with current technology to avoid interference (Figure 8.43). This, however, limits the number of geostationary orbit *slots* available in total.

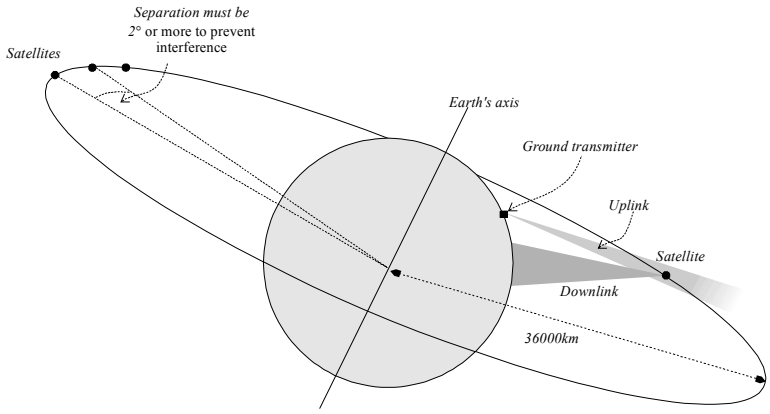


Figure 8.43 Geostationary satellites

A communications satellite receives signals from ground stations on a ground-to-space channel called an *uplink*. Two ground stations must not transmit to the satellite at the same time on the same carrier frequency or they will interfere, and a MAC protocol is required to ensure this does not happen. The satellite then retransmits received signals on a *downlink*, which, using frequencies different from any of the uplinks, broadcasts a directional microwave beam to some or all of its ground stations. Downlinks may be wide area (perhaps 10000km^2) or *spot beams* (as little as 250km^2) aimed at specific points on the Earth's surface. Typically the wider the area the downlink covers, the greater its bandwidth, so that it can carry several channels at once. A satellite may have several downlink transmitters or *transponders*, which it can direct at different areas of the surface. Satellites may simply echo the received signals with a frequency change (so called *bent-pipe* operation), or may also perform onboard processing.

Three commercial satellite communication bands have been specified by international agreement. The *C Band* has 0.5GHz reserved for downlink channels, between 3.7GHz and 4.2GHz, and 0.5GHz for uplinks between 5.925GHz and 6.425GHz. The *Ku Band* also has 0.5GHz bandwidths available beginning at 11.7GHz for downlink and 14.0GHz for uplink. Finally the *Ka Band* has 4GHz for downlinks beginning at 17.7GHz and for uplinks at 27.5GHz. There are other bands for military and general government use. Individual satellite transponders may have transmission bandwidths up to 50MHz, giving total bit rates of 50Mbps or so, which may be shared by FDM or TDM—a satellite may have more than a dozen such transponders.

Very small (<1m diameter dish) low-power (~1W) ground-stations called *VSATs* (*very small aperture terminals*) are now sometimes used, but since satellites relay signals, these may not generate sufficient power or lack the necessary sensitivity to allow one to pick up the retransmitted message from another directly. Because of this, a central large ground-station called a *hub* may be necessary, which has a large

enough antenna to receive a relayed VSAT transmission, and can then re-send it back to the satellite sufficiently amplified, that its target VSAT will be able to resolve it (Figure 8.44). The penalty is that the signal must travel twice into space, a combined distance of at least 144,000km, and a delay, therefore, close to 0.5 seconds, even for an electromagnetic signal travelling at the speed of light. The VSAT uplinks avoid conflict by means of fixed channel assignments. These uplinks are usually relatively slow, with a typical bit-rate of no more than 64kbps, while the downlink is a faster common channel of perhaps 2Mbps.

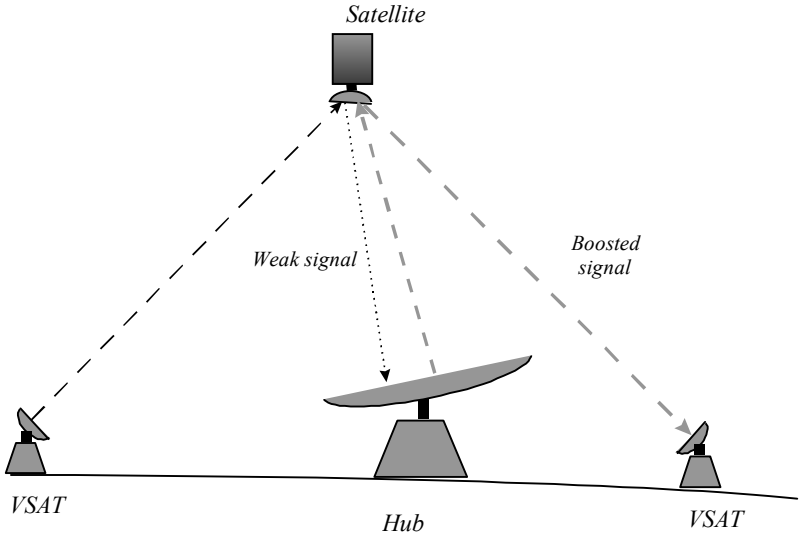


Figure 8.44 VSAT communication via a hub

Of course, even when more powerful groundstations or satellites are used, and only a single trip to orbit is required, the distance travelled still equates to a delay of greater than 0.25 seconds, much greater than that experienced in any terrestrial transmission. Consequently, computer networks based on satellites must use communication algorithms that take these comparatively large delays into account. In data transmission, for example, the delay associated with a satellite link is so long that extended window size options will typically be required in data-link protocols. The most interesting questions however are associated with those MAC methods that carry across from terrestrial broadcast networks.

In a satellite network, the hosts are the ground-stations, which establish satellite links with each other using some uplink and some downlink bandwidth. In fact, only the satellite itself transmits on the downlink, so there is no MAC problem associated with it. However, the uplink bandwidth must be divided between all transmitting ground-stations, and an appropriate MAC algorithm is required.

The most obvious approach is to use FDM to divide the uplink bandwidth up into separate frequency bands, which may then be allocated to ground stations in a fixed or dynamic manner. One problem is that the channels need reserved guardbands that waste some of the available capacity. In general, however, fixed allocation is too

inflexible, and the standard problems of FDMA must be tackled. This can be done by using slotted ALOHA (pure ALOHA is too inefficient), polling over a secondary terrestrial network, or some form of TDM. All of these methods need precise timing since they are slot-based. This is easily achieved by using one central ground station to transmit clock pulses, which are then broadcast to all others by the satellite. Note that because of the end-to-end propagation delay, algorithms like CSMA are not viable. The network latency is so great that a station may not realise that another has started or stopped transmitting for up to half a second after the event. However, dynamic TDMA is possible, with stations competing for time slots using, for example, reservation protocols such as those discussed in Section 8.1.2. Of course, one disadvantage of TDM is that stations must be able to transmit at very high peak speeds for short bursts (in a TDMA slot) in order to achieve much more modest average data transmission rates. Finally, more recently, CDMA techniques have been employed to allow dynamic allocation without a need for strict synchronisation. However, CDMA cannot achieve the same capacity as TDM, and, requiring a high chip-rate, needs a very fast transmitter.

As in many areas of telecommunications, satellite technology is developing rapidly. Much of the credit for this must go to *NASA*, the US *National Aeronautics and Space Administration*, who, over the last thirty years, have launched a series of enterprising and ambitious experimental communications satellites, starting with the *Relay* and *Syncom* programs of the early 1960s. In September 1993, the space shuttle *Discovery* launched the latest of these, the *ACTS* (*Advanced Communications Technology Satellite*) satellite, into geostationary orbit. ACTS is an experimental "proof-of-concept" communications satellite capable of very high data rates in the Ka band. It can operate as a wideband bent-pipe (*microwave switch matrix* or *MSM* mode) or it can demodulate incoming data and switch it digitally before retransmitting (*baseband processor* or *BBP* mode) it. The satellite has been used as a testbed for a number of experiments, including the *ACTS Gigabit Satellite Network*, linking up to 64 ground-stations capable of high-speed circuit switching at rates as high as 622Mbps. Another experiment involves high-performance mobile communications using the satellite's novel multibeam and steerable antennae to track fast moving targets such as aircraft. This latter technology has to combat the effects of Doppler shift and the susceptibility of Ka transmissions to absorption by atmospheric water.

Interestingly, a recent development has been renewed interest in low earth orbit (*LEO*) satellites. These are cheaper to launch, impose much less of a propagation delay on signals and can use cheaper and more portable ground-based antennae. However, they orbit the Earth rapidly and consequently, viewed from any fixed point on the surface, do not remain above the horizon for long. A way of combating this is to use a *necklace,* a series of equally spaced LEO satellites all in the same orbit, so that, as one disappears from view over the horizon, another has already risen to take its place.

One of the most ambitious proposals of this type is Motorola's *Iridium* system for global mobile telephone communications. This requires the use of 6 *necklaces* each in a different polar orbit 750km above the surface, and each consisting of 11 satellites. Although the orbital period is short, as one satellite moves out of range of a point on the surface, another has already taken its place. Unlike other cell phone systems,

Iridium will be available across the entire globe, and will not be affected by natural obstacles such as mountains. At the time of writing deployment of Iridium has not yet begun, but other less ambitious LEO systems, such as *Orbcomm*, a joint venture of Orbital Sciences Corporation and Teleglobe Inc, are already operational. The latter will eventually use 36 LEO satellites to support low rate data and fax, provide geolocation capabilities and so on. With fewer satellites, Orbcomm does not guarantee full coverage at all times (a user may have to wait a few minutes for a satellite to come into view). Further it employs low bandwidth VHF uplinks and downlinks and is not suitable for high-volume data transfer. Nonetheless it provides a useful service.

Satellite communication is not without problems. As we have seen, atmospheric conditions can affect signals, especially in high-frequency microwave bands where water-absorption is a significant issue. Another source of trouble is solar activity, which can seriously disrupt or even disable satellites. However, despite these difficulties, satellites provide an alternative to the installation of huge networks of ground-based links and are the most promising vehicle for telecommunications with truly global coverage. For these reasons they will play an increasingly important role in the future of telecommunications and mobile telecommunications in particular.

8.6 SUMMARY OF CHAPTER 8

- The bandwidth in a broadcast network can be allocated via TDMA or FDMA, and can use static or dynamic methods. FDMA often relies on in-band TDMA.
- Dynamic allocation is more efficient than static.
- Dynamic TDMA divides into two main categories, collision resolution and collision avoidance.
- The simplest resolution methods are ALOHA and slotted ALOHA. CSMA provides an enhancement, while CSMA/CD has proved the most successful of all.
- Collision avoidance methods include polling, token passing, CSMA/CA and reservation protocols. Rings can also loosely be considered to fall into this category.
- LANs are the most familiar examples of broadcast networks. The most important standard in LAN and MAN technology is IEEE 802, which defines a layering structure for LANs and defines a number of protocols, including MAC for CSMA/CD, token ring, token bus and DQDB.
- Ethernet (more accurately 802.3) is the most widespread LAN technology. Originally based on a serial cable bus, it is now often implemented using hubs or switches connected to hosts by twisted pair (10Base-T).
- Ethernet uses 1-persistent CSMA/CD, with any station detecting a collision stopping transmission immediately.
- An Ethernet switch is really a high-performance store-and-forward packet switch, using the same host-network interfaces as standard Ethernet.
- Token ring (802.5), advocated by IBM, tries to address the problems of indeterminacy and lack of traffic priority classes in Ethernet.
- Token bus was developed for use in noisy factory environments, where reliability is all-important.

- Bridges are relays that operate at the data-link level, translating one MAC frame format to another. Bridging is only viable between networks with fairly similar MAC protocols. However, it is possible to construct complex internetworks of similar LANs connected only by bridges.
- There are two types of bridge for use on 802 LANs. Transparent bridges connect LANs without stations needing to worry about the relay process. Source bridges, defined in 802.5, rely on stations to determine the route for packets before transmission. In general the former are favoured since they do not load host stations.
- A more recent idea than the LAN is the metropolitan area network. MANs may be used to provide residential data services, or to act as high-speed backbone interconnections for LANs.
- CATV cable systems using FDM are a possible candidate for residential data services.
- The optical FDDI network is already widely used for backbone provision. Its operation is similar to token ring, but it is much faster operating at 100Mbps.
- The DQDB (802.6) system uses a complex but elegant algorithm to implement a distributed FIFO queue of transmission requests from stations. It uses slots the same size as ATM cells, and provides a mechanism to support isochronous data.
- SDMS is an interface standard for connecting LANs to a backbone. It has been adopted by public carriers, who offer a service now largely based on special switched networks. SMDS is intended for use with asynchronous traffic and does not support isochronous flows.
- Fast Ethernet technology has become immensely popular for use in high-speed LANs. The primary technology, 100Base-T, operates at 100Mbps, is hub or switch based and operates over category 3 or category 5 UTP.
- Wireless LANs may use radio or infrared. The latter are really only suitable for use within a single room, but the former can be much more wide-ranging.
- CSMA techniques must be adapted for use in radio networks. The DFW MAC protocol can be used to compensate for the fact that not all stations can directly reach each other.
- Another approach, used in mobile telephone systems, is to use a network of adjoining cells, each with a central relay station.
- Spread-spectrum techniques can enhance resistance to noise and security in radio networks.
- Communication satellites in geostationary orbit are in a fixed position relative to the ground. They can be used to relay transmissions from ground stations, either directly (bent-pipe) or after processing.
- LEO satellites are cheaper to launch, require lower power and suffer less of a propagation delay than those in geostationary orbit. However, they rise and set as seen from the ground, so if they are to provide a continuous service, they must be organised with several to each orbit.

8.7 EXERCISES

8.1. Why might packet arrivals in a real network not follow a Poisson distribution?

8.2. Derive Equation 8.8 for slotted ALOHA.

8.3. Show that if propagation delay is taken into account, the equation for slotted ALOHA becomes:

$$S = Ge^{-2(a+1)G}$$

where a is the propagation delay across the medium (measured in packet times). What simplification is inherent in this proposition?

8.4. Find the maximum theoretical throughput for 1-persistent CSMA.

8.5. At what load does nonpersistent CSMA become superior to 1-persistent CSMA?

8.6. What happens if p-persistence is used with p=1?

8.7. Discuss the advantages and disadvantages of a bidirectional ring topology.

8.8. What is the bit length of a ring link of length L metres, with a clock rate of R Hz and a signal velocity of c_{sig}.

8.9. Discuss the effect of changing the maximum packet size in a CSMA/CD network.

8.10. Discuss when it would be appropriate to use the various Ethernet implementations, 10Base5, 10Base2 and 10Base-T.

8.11. How many global Ethernet addresses are available?

8.12. In switched Ethernet, when a switch's buffer space is exhausted, why is it preferable to signal a collision back to the sender, rather than just discarding packets?

8.13. In an Ethernet switch, discuss whether the collision signalling system can implement flow control.

8.14. Compute the maximum frame size in a 4Mbps token ring with a maximum token holding time of 10ms.

8.15. Discuss the advantages and disadvantages of early token release, and explain why it is more important in faster rings.

8.16. Why is token bus performance likely to be inferior to token ring, operating at the same speed, when framer sizes are small?

8.17. In Figure 8.22, the LLC frame format, unlike that of HDLC does not include frame flags. Why are these deemed unnecessary?

8.18. Explain why backward learning in transparent bridges can cause looping.

8.19. In the LAN internetwork of Figure 8.25, draw two other possible spanning trees, assuming that Bridge 1 is the root.

8.20. Compare the cost of all-routes and single-route discovery in source-routing bridged internetworks

8.21. In the DQDB of Example 8.1 explain in detail what will happen if:

a) a Priority 0 segment is queued at Station 2 after the segments queued at Stations 3 and 4, but before an empty slot appears;

b) the second request at Station 4 is Priority 1.

8.22. The distributed queue access protocol is in fact, not entirely fair, even with the "one outstanding request" rule. To see this consider two widely separated stations, S and T, with S far upstream of T. Now suppose both stations are

sending large amounts of data on bus *A*, and that *S* starts transmitting first. Explain why *S* is greatly favoured in this situation as far as its share of the bus *A* bandwidth is concerned? (*Hint*: the large distance between *S* and *T* is very important).

8.23. Discuss the reasons why CSMA and CSMA/CD are not viable MAC strategies in satellite networks.

8.24. If a satellite uses 1ms TDM frames with 1728 slots, each containing 64 bits, how fast will the ground-station have to transmit during its slot? (these figures are taken from the *Advanced Communication Technology Satellite*, launched in 1992).

8.25. Re-draw Figure 8.40 to illustrate the use of seven frequency groups.

8.26. In direct sequence spread spectrum, prove that the data in a chipping code can be recovered by exclusive-ORing it with the spreading sequence again.

9
WIDE AREA BROADBAND NETWORKS

9.1 ISDN

Communications networks have evolved at different times from a number of different requirements. The oldest are those, like the telephone and telex systems, which arose to support voice and text transfer; then came those inspired by the requirement for computers to exchange asynchronous data; and, most recently, those intended to support the most demanding high-bandwidth isochronous data generated by multimedia applications. Although these needs have arisen at different times, and, to some extent, initially generated quite different solutions, it has long been a goal of network providers to unify them and establish networks capable of providing users with a communications service to handle all forms of data.

The first major international effort to achieve this integration of services was published in 1984 under the auspices of the then CCITT. This somewhat predated the explosion of interest in high-bandwidth multimedia that has occurred in the 1990s, and the emphasis was therefore more on unification of voice and data (meaning asynchronous, or bursty, computer data). However, it was a concrete effort to establish standards that could realise the aspiration of a unified service. The initiative recognised that all WANs, including the telephone network, would soon be digital, and was therefore called *integrated services digital network* or *ISDN*.

There are a couple of points worth making at the outset. Firstly, ISDN was envisaged as an attempt to create a wide area network and network service, based largely on the IDN telephone infrastructure. It did not consider the position of LANs, which were seen at the time as being primarily for the transfer of asynchronous data between local computers. This is not to say that ISDN is incompatible with LANs, simply that ISDN service was seen as being offered to subscribers by a public network, just like the PSTN and the packet-switched public data networks (*PSPDNs*)

which were springing up at the time. ISDN was to be a unification of the circuit-switched IDN and the PSPDNs, which would offer additional services besides.

Secondly, when ISDN was designed it was not appreciated just how much bandwidth the integrated services of the future would require. Several years later, when this became apparent, CCITT realised that much higher bandwidth technology would be required, and by 1988, had published the first recommendations concerning standards for a much higher bandwidth ISDN network. This required postulation of technology that did not at that time exist and was an altogether much more ambitious undertaking than the original ISDN proposals. To distinguish the two, the later version is known as *broadband ISDN* or *B-ISDN*, and the original is now sometime referred to as *narrowband ISDN*. The term "broadband" is used here to mean "high bandwidth", and does not imply, as in earlier usage, that FDM techniques are being employed. In fact B-ISDN will be based on ATM operating over SDH or SONET, which are not bandpass or FDM technologies. Although this chapter is primarily concerned with these "broadband" technologies, they are so intimately connected with the idea of integrated service, and narrowband ISDN is so clearly the first step towards this, that we will begin by examining it in a little detail. In what follows, we will omit the term "narrowband" and distinguish simply between ISDN and B-ISDN.

ISDN is defined in ITU-T's *I-series Recommendations*. The I-series is divided into six main groupings:

- *I.100* covers general concepts and terminology;
- *I.200* specifies services;
- *I.300* discusses network issues such as addressing and common signalling;
- *I.400* is concerned with the user-network interface, including specification of network access protocols;
- *I.500* deals with internetworking, i.e. interfacing between an ISDN and other types of network;
- *I.600* covers standards for the maintenance of the "network" portion of the ISDN subscriber interface.

As we have seen, ISDN is based on the IDN, but it requires that subscribers have digital access. This means that the standard analogue local loop used by most telephone subscribers must be replaced with a digital link. As we shall see, in most cases, this does not necessitate replacement of the actual twisted pair subscriber-exchange cable, merely the equipment to which this cable interfaces at the exchange and at the subscriber's site.

In ISDN a subscriber's equipment is connected to the local ISDN exchange by a full duplex *digital pipe*. In reality the ISDN exchange will usually be the (digital) telephone exchange. The digital pipe can support several types of full-duplex *channel*:

- a *B-channel* (*bearer*) runs at 64kbps and is intended to carry one PCM-encoded digital voice stream or arbitrary computer data over a circuit or packet-switched connection;
- a *D-channel* (*data*) runs at 16kbps or 64kbps depending on the situation, and is intended for common-channel signalling or packet-switched data;

- a *H-channel* may run at 384kbps (H0), 1536 (H11) or 1920 (H12), for video or high-speed data.

ISDN can be offered to a subscriber in two different forms. The *basic rate interface* (*BRI*) provides simultaneously 2 B-channels and a 16kbps D-channel (2B+D), is intended to operate over a standard telephone twisted pair, and therefore to act as a replacement for a PSTN local loop. The *primary rate interface* (*PRI*) is a much higher bandwidth connection which can be offered over a T1 (US) or E1(European) link. Over T1, PRI combines 23 B-channels and a 64kbps D-channel (23B+D), while, over E1 it consists of 30 B-channels and a 64kbps D-channel (30B+D). All ISDN networks do not offer H-channels, but where they are available, a PRI access can be alternatively configured as, for example, 3H0+D, 4H0 or H11 on a T1 link, 5H0+D or H12 on an E1 link. Channels operating over the same digital pipe can be used independently to service quite different conversations, or may sometimes be *bonded* together to provide greater bandwidth on a single link.

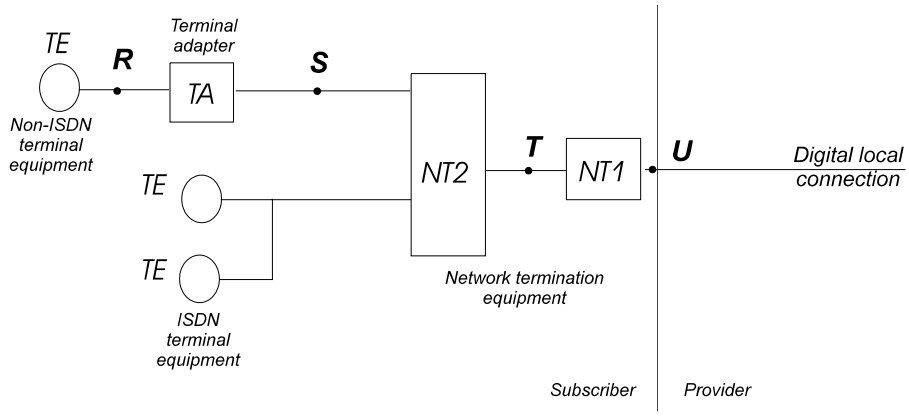


Figure 9.1 ISDN subscriber reference points

An ISDN subscriber may attach equipment of varying nature to the network. This equipment may include, for example, a telephone, a computer, a *PBX* (*private branch exchange*) or even a LAN. Indeed several items may be attached to the same ISDN interface. The standard general structure of such an interface is shown in Figure 9.1. As can be seen from this diagram, the physical digital connection from the ISDN exchange is terminated at the subscribers premises by *network terminating equipment* (*NTE*), which may be divided into two functional blocks designated *NT1* and *NT2*. NT1 is the basic physical termination of the external line, and provides a physical connector for the subscriber. The interface to NT1 on the network side is given the designation *U*, while that on the subscriber side is called *T*. NT1 may be owned by the subscriber or by the provider depending on local regulations but it does, in any case, supply the most basic interface to which user equipment can be attached. In fact the I.400 recommendations do not specify the U-interface at all, since it is not strictly of concern to subscriber equipment. However, CCITT have issued a separate specification for this interface in *Recommendation G.961*.

In BRI the U interface must carry 144kbps of data full-duplex (2B+D). G.961 allows several alternative line codes, although a common implementation is firmed up in *ANSI TI.601*, which we now outline. The data is carried in a TDM format that together with framing bits etc. requires an actual data transfer rate of 160kbps full duplex. Fortunately, most local loop twisted pairs can support data transfer at these rates, so long as the analogue terminating equipment at exchange and subscriber premises are replaced, the line is no longer than about 5km, and all inductive loading is removed.

To achieve 160kbps full duplex on a normal telephone twisted pair, TI.601 specifies the use of quaternary (four-level) signalling in a coding system called *2B1Q* (*two binary one quaternary*), operating at 80kbaud. Out of interest, the signal levels specified are +2.5V (representing the bits 10), 0.833V (11), -0.833V (01) and -2.5V (00). Full duplex transmission may be achieved by echo-cancellation or by a technique called *time compression multiplexing* (*TCM*), where each end of the line is given an equal period to transmit while the other is idle, in a precisely synchronised fashion.

The NT1 equipment, converts the two-wire signalling to four-wire, and essentially provides a physical level connection to any subscriber equipment using the T interface. The connector specified has 8-pins (ISO 8877), although only four or six are used at a time. Two pins are for the transmit pair, two for the receive pair and the others are used for power transfer from the NT1 unit to the equipment. Unlike an analogue phone line, a digital local connection does not transfer power from the exchange, and therefore a battery at the subscriber's premises must supply emergency power. NT1 also converts the TDM format used over the exchange line to that specified in ITU Recommendations *I.430* (*BRI*) and *I.431* (*PRI*). I.430 stipulates a 48-bit frame structure repeated every 250ms, for an overall bit rate of 192kbps. Each frame contains 16 bits from each of the B-channels and 4 bits from the D-channel. A pseudoternary code is used (see Section 3.5.3).

The situation in a PRI installation is very similar, except, of course, that the U-reference point is replaced with a T1 or E1 line. The T-reference point then gives access to whatever channel combinations are being implemented, using a much faster data rate and a frame structure similar to that of the T1 (193 bits every 125ms) or E1 (256 bits every 125ms) carriers, encoded as B8ZS or HDB3 respectively.

A connection to the T-interface will allow a piece of equipment raw access to the B, D or H-channels, but interfacing to an ISDN network is not simply a physical issue, and this, as we shall see shortly is where NT2 comes in. ISDN allows for three basic categories of end-to-end service (ignoring frame relay), notably circuit-switched, semi-permanent and packet-switched connections. We now consider these in turn.

Circuit-switched calls over a B-channel are set up using the D-channel for signalling, and may employ any end-to-end protocol. The B-channel can then be used to carry PCM voice produced by a subscriber's ISDN telephone or terminal adapter, or a data communications protocol stack chosen by the users. A three-level protocol stack has been defined for the D-channel to support signalling and the situation is illustrated in Figure 9.2. Since the signalling and communications are separate, the stacks in use for each are also separate and are said to form the *control* or *C-plane*, and the *user* or *U-plane* respectively.

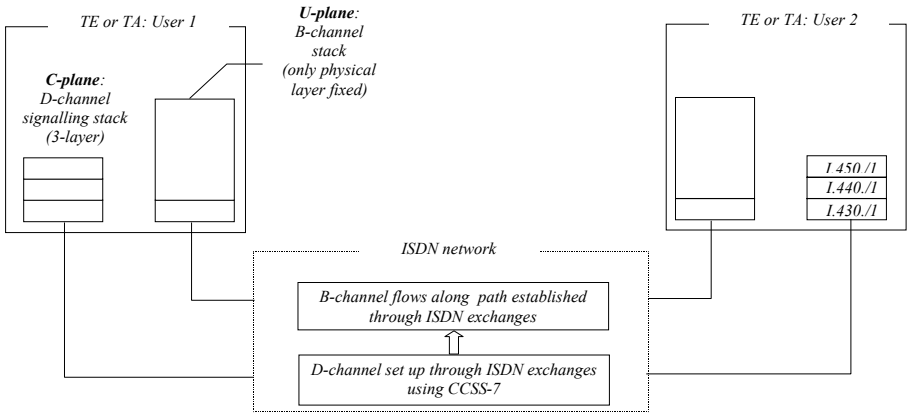


Figure 9.2 ISDN circuit-switched connection

The C-plane uses a three-layer protocol stack for signalling. The bottom layer is just I.430/1, which is shared by the U-plane. Above this is a data-link protocol, *I.441*, also called *LAP-D* (*link access procedure: D-channel*), which is a modified form of HDLC. Finally above this is the control signalling protocol proper, *I.451* also known as *Q.931*. This is a packet protocol that allows one user to send control messages to establish and clear calls, provide information and so on. These packets are interpreted by the ISDN exchanges, which use CCITT common channel signalling system no 7 (*CCSS-7*) to communicate with each other. The B-channel follows the path set up by D-channel call establishment, and this mechanism is the same regardless of whether the B-channel is to carry voice or data.

A s*emi-permanent connection* is similar to a circuit-switched call but is established between two ISDN access points for a predetermined period or periods agreed by the subscribers. As with ordinary leased lines, while the semipermanent connection is in force, no call setup or closedown is required.

Packet-switched connections may be provided in two ways. The original intention was to have a fully integrated ISDN packet service provided by ISDN packet switches which might be distinct from, or part of, the normal circuit exchanges. In this scenario, a subscriber establishes a circuit-switched connection to the local packet switch, or alternatively acquires a semi-permanent connection to the packet-switch, and may then communicate with that switch using a B-channel connection running LAP-B and X.25 PLP (packet layer protocol). Interestingly however, it is also allowable for the subscriber to use a D-channel running LAP-D and X.25 PLP. Since this is permissible, ISDN exchanges must be able to distinguish between LAP-D frames carrying I.451 packets for signalling and those carrying X.25 packets as part of a packet-switched link: this is done using LAP-D's addressing scheme, which is capable of distinguishing between I.451 and X.25 packet handling entities.

The second approach is to use packet switches provided by a separate *PSPDN,* to which, again, subscribers may connect using an ISDN circuit-switched or semi-permanent link. In this case, however, the connection must use the B-channel running some protocol stack such as LAP-B with X.25. This way of accessing packet-switched

services was added to the original, as it became apparent that many PSPDNs would continue to exist alongside any ISDN. Note that an ISDN user is then accessing the PSPDN in the same manner as a telephone user might do at present, and can communicate with any other user of the PSPDN whether the latter is connecting via ISDN or not.

The originators of ISDN realised that packet switching would remain a popular method of data communication even if fast-connect circuit switching were available. However, initially they assumed that the major mechanism for such packet switching would be X.25. By 1988, it was clear that the overhead involved in X.25 would become increasingly difficult to justify as networks became more reliable and performance demands increased. In an attempt to tackle these issues, CCITT introduced in Recommendation *I.122,* a new low-overhead means of packet communication, called *frame relay*. Considerable development has been invested in frame relay since its inception, and we will look at it in more detail in Section 9.2.

It is clear that these basic service categories require varying degrees of protocol support. Given the three-level structure of the D-channel signalling stack and the use on B-channel of services like X.25 and frame relay, it turns out that ISDN fits quite nicely into the OSI Reference Model. The fit is not perfect, in that some aspects, like the separation of the C and U planes, are not present in the OSI model, but it is a fit nonetheless. The basic services (circuit-switching, semi-permanent connection, packet switching and frame relay) are used to define a number of so-called *bearer services* in ITU Recommendation *I.230*. These latter are based on the former but are a little more specific. They include, for example: a 64kbps unrestricted circuit-switched service for general data transfer; a 64kbps structured circuit-switched service for PCM speech; a 2×64kbps unrestricted circuit-switched service (two *bonded* B-channels); 384kbps, 1536kbps and 1920kbps unrestricted circuit-switched services; a virtual circuit packet-switched service etc.

The NT1 function provides only a basic physical interface capable of transmitting B-, D- and H-channel streams. ISDN TEs may be connected directly to it at the T interface, or may be attached through an NT2 unit at the S point. The NT2 function, if present, is a device that performs switching and concentration of many ISDN TEs onto a single physical interface. Probably the commonest NT2 unit would be a digital PBX, but terminal controllers and certain types of LAN might also be used in this role. In BRI, the S- and T interfaces are essentially identical, and similar ISDN capable equipment may be attached to each. Indeed when an NT2 device is not explicitly present, the two reference points may be considered to be the same, and we will henceforth refer to the point to which an ISDN TE is attached as S. Non-ISDN equipment in attached via a *terminal adapter* (*TA*) via a so-called *R* point. The R-point may be an appropriate standard interface such as RS-232 or RS-423 to which standard computer serial ports can be attached, or it may supply a connection point for a normal analogue telephone or modem.

In fact, it is possible to attach up to eight TEs to one S (or T) connector in a serial multidrop configuration, known as an *S-bus*. This, of course, presents problems of ensuring that contention is resolved and that incoming data, which of course is supplied in TDM format, is correctly delivered to the appropriate device. The D-channel is used to resolve contention using the following algorithm. On the S-bus, the

electrical characteristics are *wire-AND*, which means that if any one TE transmits a "0" on a shared line, this will pull the line low, regardless of how many other TEs are transmitting a "1". TEs with nothing to send transmit a "1" on the S-bus during all I-430 frame time reserved for the D-channel (recall that there are 4 D-channel bits in each such frame), while those wishing to send transmit a "0". The S-bus goes low if and only if there is one or more TE request. However, the I.430 frame structure also includes an *echo bit* for each D-channel bit, and the NT unit uses these to reflect back the D-values received from the combined TEs. A terminal seeing a sequence of echoed "1s" for some *threshold time* assumes that the D-channel is free and may send a LAP-D frame. Of course collisions can occur, but these are spotted because the TE compares what it has sent with the echo bits.

Once a D-channel dialogue is established, a TE can be granted sole use of a B-channel to make a call. Note that even if all B-channels available to a subscriber are in use, D-channel control signalling can still progress. When a D-channel packet arrives at an S-bus, the addressing in the LAP-D frame directs it to the correct TE (more of this shortly). B-channel data only appears after the B-channel in question has been granted to a particular TE, so the addressing issue does not arise here.

At this point it is worth saying a brief word about ISDN addressing (see Figure 9.3). First of all every ISDN subscriber installation has at least one *ISDN number*. The numbering plan for ISDN is outlined in *I.330* and *I.331*. The ISDN number is structured just like a telephone number and, like the latter, provides the circuit-switched network with enough information to route a call. However, a single number may serve many S-points in an installation which includes, say, a PBX. To distinguish these "extensions", each is given an *ISDN address*, which consists of an ISDN number together with a suffix called a *subaddress*. In some cases S points may be given their own separate numbers, so that they can be reached without an extension subaddress. This is called *direct dial-in* (*DDI*), and, typically, a PBX using this feature will be allocated a block of ISDN numbers. Of course, this uses up the number space faster and its use is generally restricted.

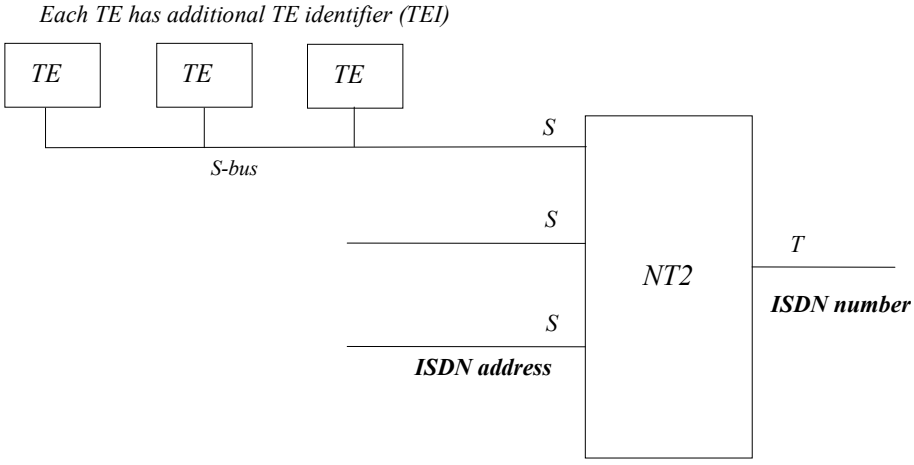


Figure 9.3 ISDN numbering and addressing

Even when an S point has been uniquely identified by an ISDN address, if it is being used to support a multidrop S-bus, a further subaddressing mechanism is required. This is provided as discussed above using an address field in the header of LAP-D frames. Each device on an S-bus is allocated a unique (locally) 7-bit *terminal endpoint identifier* (*TEI*). Additionally, within a TE, a distinction is drawn between different layer-3 protocol entities, by means of a 6-bit *service access point identifier* (*SAPI*). The SAPI values are fixed for each type of protocol. For example, a SAPI of 0 is for call-control signalling, a SAPI of 16 is for X.25 packet communication, 32-62 for frame relay, and so on. This is how, as mentioned previously, ISDN terminals and the network itself can distinguish between control and packet traffic on the D-channel.

In addition to bearer services, *I.241* defines a list of *teleservices,* which an ISDN network can provide using standard protocol stacks that build upon the three layer stacks discussed so far. These services include fax, teletex, videotex, (teletex and videotex have both been rather overtaken by events, and probably have little future) as well as OSI compatible messaging services. Further *supplementary services* are also defined which are associated with particular bearer or teleservices. The supplementary services include facilities such as *call waiting*, *call transfer*, *calling line identification*, *conference calling*, *credit card calling* and so on. Many of these are already familiar features of digital PBXs and even of the standard IDN telephone network itself, available to all subscribers whether or not they interface via ISDN.

In fact the ISDN exchange network is already in place, for modern digital telephone exchanges are ISDN capable in any event. Unfortunately, due to the high charges levied by many major telecom operators, ISDN take-up, particularly by private subscribers, has been low. ISDN, at least in BRI form, suffers, in a sense, from its lack of ambition. It can offer data transfer rates of at most 144kbps, which is really not adequate for many applications such as real-time multimedia. Having said that, it is a superior alternative to even the fastest modems now available to private users wishing to download information from web or FTP servers. However, in most countries with the capability to offer an ISDN service the price-performance is not satisfactory and unless charges are reduced, it seems likely that BRI will be completely obsolete long before it attains any degree of market penetration. Even PRI is only rated at 2Mbps, which although very attractive for many users, is well below the performance offered by even average LANs. Attention is now turning to B-ISDN, which we shall examine shortly. However, before that, we will look in more detail at one of the most important spin-offs to come from the ISDN project.

9.2 FRAME RELAY

X.25 and its intranet companion protocol X.75, were devised at a time when physical communication networks were less reliable than they are now, and the requirements for high throughput were less important. If we look at X.25 critically as a mechanism for the rapid transfer of data, it is seen to be so concerned with the possibility of error, of dropping packets etc, that it sacrifices performance. While this might be appropriate in some circumstances, in a highly reliable network like an ISDN it is often seen as a drawback.

Anticipating the growing demand for a low-overhead packet-based data transport system, ITU developed *frame relay* as an additional bearer service to be offered on ISDN networks. Frame relay essentially revamps the X.25/X.75 protocol stack, reducing the elements to the bare minimum necessary to provide rapid packet transport. The service has proved so attractive non-ISDN networks now also offer it.

The first point about frame relay is that reduces the three-layer X.25 protocol stack to two, performing several functions at layer 2 (the "frame layer"), which OSI and X.25 saw as the responsibility of the packet layer. The service is, like X.25, connection-oriented and two TEs wishing to interchange frames must first establish a frame relay or *data-link connection* (*DLC*), to which each assigns a connection number called a *data-link connection identifier* or *DLCI*, entirely analogous to an X.25 VCI. Like an X.25 virtual circuit, a frame relay DLC can be switched or permanent. At present, frame relay tends to be used, rather like SMDS, as a way of interconnecting LANs at different sites and in most networks only the permanent connections are actually implemented. However, the standards for switched frame relay connections have been published and we will consider both possibilities here.

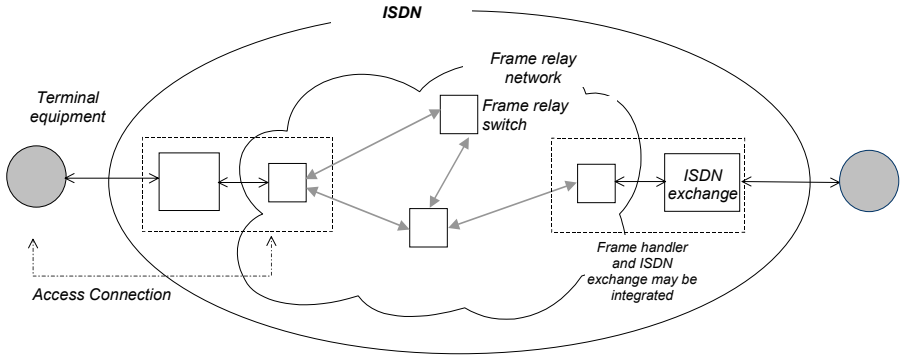


Figure 9.4 Frame relay network in ISDN

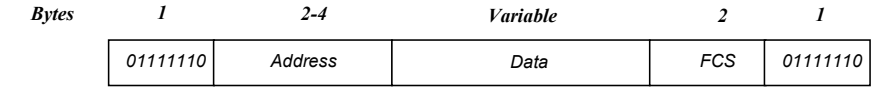
The frame relay service, like the packet-switching service, is provided via special nodes called *frame relay switches* or *frame handlers*. In an integrated ISDN frame relay network of the type envisaged by ITU, frame handlers are associated with or integrated into the ISDN exchanges, and communicate with each other using the trunk fabric of the ISDN itself (Figure 9.4). In a non-ISDN frame relay network, leased lines or some equivalent would be used to implement inter-switch links. In either case we call the collection of frame handlers and their inter-switch links, the *frame relay network*.

To set up a frame relay connection, a subscriber TE must first make contact with the local frame handler. On ISDN, this may be done using a B or H-channel, switched or semi-permanent circuit, or using the D-channel LAP-D with an appropriate SAPI. In any case, the connection between TE and frame handler is called a frame relay *access connection*. If permanent DLCs are being used, no further setup is required and frame exchanges can begin using a pre-arranged DLCI. Note that, as with X.25, the two ends of the connection may use different DLCIs to identify it, with automatic conversion occurring within the network. This is because different numbers may

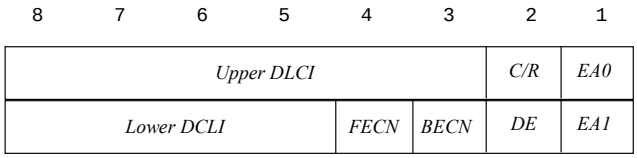
already be in use at each end. However, where switched frame relay connections are required, a path must be established through the frame relay network itself.

Frame relay, like ISDN, uses a separate protocol to control DLCs. This protocol is indeed a subset of the Q.931 signalling system used by ISDN itself, and is called *Q.933*. The Q.933 packets are sent to the frame relay network where they are interpreted and used to set up switched DLCs between the entry and exit frame handlers (see Figure 9.4). Interestingly, the way the Q.933 packets are sent to the entry frame handler depends on whether the frame relay network is integrated with the ISDN or not. In the former case, the Q.933 packets are sent over the D-channel using LAP-D with a SAPI of zero (indicating true ISDN control signalling). In the latter, however, they are sent as the data fields of frame relay frames, using the reserved DLCI of zero, to indicate that they are, in fact, to be interpreted as messages to the network rather than true data for transport.

Once the path through the frame relay network and the two access connections are established, frame exchange can begin. In an integrated network, frames can pass from TE to local frame handler via B, H or D channels as appropriate; in a separate network it is not permissible to use the D-channel for this. In either case, no further routing is required, and all a frame handler need do is maintain a table for each DLCI that allows it to direct a frame on a given connection correctly. DLCIs allow multiple frame relay connections to be multiplexed over the same physical (e.g. ISDN) link. Further, in frame relay networks, the switches perform minimal error and no flow control. When trouble arises, frames are summarily discarded, and it is left to higher-level end-to-end protocols to detect and recover from this.



a) Frame format



b) Address format (2 byte)

Figure 9.5 LAP-F frame format

The frame relay data protocol is *LAP-F*, a simplified form of the data-link protocol LAP-B. LAP-F frames have the format shown in Figure 9.5a. Unlike LAP-B or HDLC frames, LAP-F frames have no control field, and consequently there is only one type of frame. By default, the address field is 2 bytes long and includes a 10-bit DLCI, but it may be extended to 3 or 4 bytes (with 17 or 24-bit DLCI fields). The standard 2-byte format is shown in Figure 9.5b. The EA bits are used to distinguish long from short addresses, being "0" in all but the last address byte. The DE, FECN

and BECN bits are used for congestion control, while the C/R bit is application-specific.

Since frames have no numbers, there can be no data-link sequence checking or acknowledgements and no flow control. A frame progresses along its route and is simply checked for validity at each frame handler, using the still-present FCS. Faulty frames are discarded

Incidentally, in the unusual case where the D-channel is used to send LAP-F frames, these are distinguished from LAP-D frames by means of the 6 most significant bits of the most significant address byte. These correspond to the upper DLCI field in LAP-F and the SAPI field in LAP-D and values between 32 and 62 indicate a LAP-F frame.

Frame relay eschews the traditional approach of controlling the flow of PDUs in the network. Of course there will be times when handlers become congested and run out of buffer space, and, in the absence of flow control, new frames will continue to arrive. Frame handlers then have no choice but to discard valid frames. Frame relay recognises this and tries to implement simple mechanisms to help the handler in making a more intelligent choice than simply discarding the frame most recently arrived. The key is the *DE* (*discard eligibility*) bit, which the network sets in frames that are to be discarded first. To try and be fair, each connection in a frame relay network is allocated a *committed information rate* or *CIR*, which stipulates the maximum rate at which the network will expect to have to handle data on that connection. The idea is that frames sent in excess of the CIR on their connection, have their DE bits set and are the first to go, if congestion ensues.

In fact frame handlers measure the activity on a connection over some fixed measuring interval, *T*. The maximum amount of data which the network expects to handle on a given connection in time *T* is, of course,

$$B_c = T \times CIR$$

a quantity known as the *committed burst size*. In fact, another quantity, called the *excess burst size*, B_e is also defined, and, if the amount of data sent on the connection in time, *T*, reaches $B_c + B_e$, the network will simply refuse to handle any further frames until a new measuring period begins.

As we saw in Section 7.5.1, load shedding does not in itself tackle the causes of congestion. While frame relay does not include any mechanisms to help avoid or control congestion, it does provide help to protocols that may run over LAP-F. In particular the bits *BECN* (*backward explicit congestion notification*) and *FECN* (*forward explicit congestion notification*) are used by frame handlers to inform hosts and other frame handlers that there is at least an incipient congestion problem. BECN notifies that there the rate in the reverse direction to the frame carrying the bit is too high, and is a direct signal to a host that it should slow down its send rate. FECN, on the other hand, carries the same message for the direction in which the frame has just travelled. To use FECN the receiver must send a message to its peer, telling it to slow down its send rate. When a frame handler begins to get worried about buffer space it may set these bits on a single offending connection, on all connections using a particular path, or *in extremis* on all connections passing through the handler.

If congestion avoidance fails, and load shedding is initiated, the network must attempt to dissipate the problem. This must be done by a protocol operating above LAP-F. Note that any data-link protocol can be used end-to-end across the frame relay network at the level above LAP-F, and, in particular, an extension to LAP-F, the *LAP-F control protocol* has been defined with this role in mind. This is like LAP-D but includes features to permit variation in window size for dynamic adjustment of flow control.

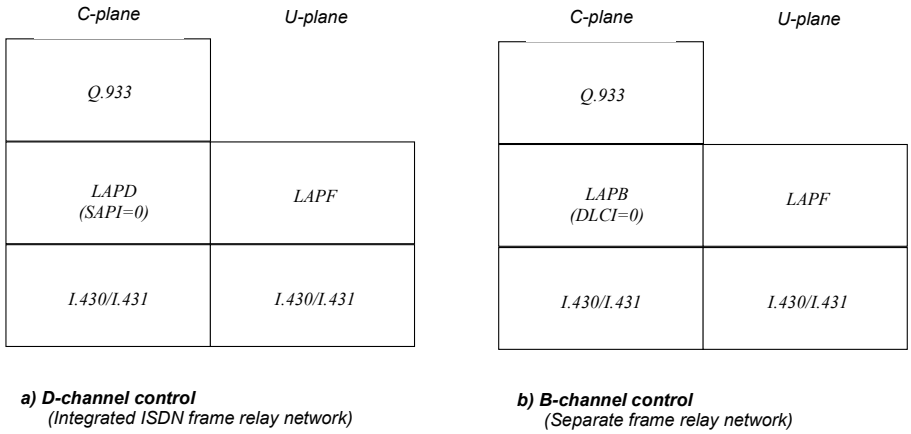


Figure 9.6 Frame relay protocol structure with ISDN access

Frame relay divides its protocol stack into C- and U-planes, just like ISDN. The C-plane may use a D-channel or B-channel to get the Q.933 packets into the frame relay network (see Figure 9.6). Of course, as we have seen, although frame relay was initially specified as an ISDN service, it is possible to access an independent frame relay network in other ways as well. Since frame relay provides a means of flexible switched or semi-permanent access between access points it could for example, in principle, be used as a fabric for interconnecting LANs, via an SDMS-like interface. However initial frame relay offerings have been pitched at the 1.5-2Mbps rates dictated by T1 and E1 access to ISDN, and, while respectable, these fall below the rates within the LANs themselves, and below the rates supported by SDMS.

Frame relay's most important role at present is as a replacement for the conventional leased lines used by many organisations to link LANs and routers at different sites into wide area networks. It is interesting to consider why telecommunications companies prefer customers to use a fast packet-switched service like this. One reason is that the packet-switched nature of the network is kinder in its use of bandwidth, since much of the time data is not actually being transferred over a standard physical leased connection. However, since most practical frame relay networks use essentially semi-permanent DLCs and many customers will also use semi-permanent access connections, this advantage is somewhat diluted. But, consider the diagram of Figure 9.7, which contrasts the two methods. As can be seen, with leased lines, each site must maintain two or more separate connections into the telephone network (although this does not necessarily imply two separate cables),

while with a technology like frame relay, a single connection to the frame handler is adequate.

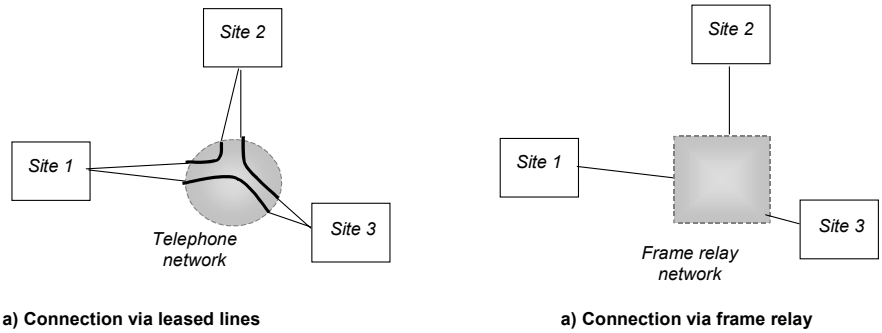


Figure 9.7 Connecting several sites via leased lines and fast packet switching

One important characteristic of frame relay is that, in common with conventional packet technologies, variable length frames are permissible within the network switching fabric. Also there is no mechanism to bound packet jitter and guarantee a service capable of transporting isochronous data. Frame relay technology is, like SDMS, unquestionably intended for asynchronous data traffic.

We now turn to an alternative fast packet technology, *cell relay*, which has been specifically designed to circumvent all constraints on the types of traffic that can be carried. Cell relay, however, will only reach its true potential when it operates over a physical system that is capable of flexibly supporting very high bandwidth transfers. MANs and LANs notwithstanding, the arena at which cell relay is really targeted is the telephone network. This is, after all, universally available across the entire globe, and is the basis of the public telecommunications providers' dream of a global integrated digital network with the capacity to support the multimedia traffic of the future. Cell relay is a single aspect of this plan, although one which is as yet only in the development stage. A great deal of effort is already being directed at fundamental improvements to the physical fabric of the PSTN/IDN, and, in particular, at the question of how subscribers can gain high-bandwidth access at reasonable cost. Before studying the cell relay, ATM, technology, therefore, we will examine these improvements briefly.

9.3 NEW TECHNOLOGY IN THE TELEPHONE NETWORK

The telecommunications providers have to contend not only with a growth in demand for telephone and data traffic, but with a new lucrative market for the transfer of high-volume data services like quality audio, video and even high definition television (HDTV). They must also attempt to combat the incursion of new competitors, such as the operators of cable TV and metropolitan area networks, who are installing high-bandwidth cables across many cities and towns and offering many types of communications services as well as television channels.

To provide an effective defence the telephone companies need to upgrade both the internal fabric of their networks and the local access channels which subscribers use to connect to local exchanges. Upgrading the internal bandwidth is the easier task because trunk fibres with enormous carrying capacity can be installed between exchanges relatively easily. Of course the exchanges themselves must also be upgraded, and, although this is a major undertaking, it has, as we saw in Section 4.2.2, been in progress for some time. Nonetheless, if new technology like ATM is adopted, these exchanges will have to be replaced again, since they are at present based purely on circuit switching and this, of course, defeats the object of packet-based resource allocation.

The internal movement of data in the telephone network is to be based on the SDH or SONET hierarchy of carriers, which, as discussed in Chapter 4, is replacing the current PDH systems. Recall that SONET was developed at Bellcore, while SDH is a derivative stan-dardised by ITU, but, in fact, terminology apart, the two systems are very similar. SONET is based on a fundamental TDM stream called an *STS-1 carrier* (*synchronous transport signal*) or, on optic fibre, an *OC-1* (*optical carrier*) which runs at 51.84 Mbps. This is enough to carry 672 64kbps uncompressed PCM telephone channels. A whole series of faster carriers are defined, with STS-3 resulting from multiplexing three STS-1 streams, STS-9 from three STS-3 streams, and so on.

SONET	SDH	Bit rate (Mbps)
STS-1/OC-1	-	51.84
STS-3/OC-3	STM-1	155.52
STS-9/OC-9	STM-3	466.56
STS-12/OC-12	STM-4	622.08
STS-18/OC-18	STM-6	933.12
STS-24/OC-24	STM-8	1244.16
STS-36/OC-36	STM-12	1866.24
STS-48/OC-48	STM-16	2488.32

Table 9.1: Hierarchies for SONET and SDH

The SDH carriers, called *synchronous transmission modules* (*STM*) are completely compatible with SONET, but begin with STM-1 at the STS-3 rate and are less numerous. In fact STM-*n* is the same as STS-3*n*. The two series are summarised in Table 9.1.

STS-1 (OC-1) frames are 125μs and 810 bytes long. The frame format is shown in Figure 9.8. The frame is transmitted as 9 rows or *segments* of 90 bytes each. It is best viewed as a matrix, with each segment beginning with 3 bytes of management, or *header*, information, and 87 bytes of payload. Things are a little more complicated than this, however, because SONET and SDH are structured to allow transmission across a hierarchy of types of route. In a typical run from a source to a destination, a signal will encounter multiplexers and demultiplexers at exchanges (we will use the term “multiplexer” to mean either a multiplexer or demultiplexer) and repeaters (on cable runs). SDH distinguishes between *sections*, runs of cable between two repeaters,

lines, runs of sections between two multiplexers, and *paths*, runs of lines between a source and destination (seeFigure 9.9).

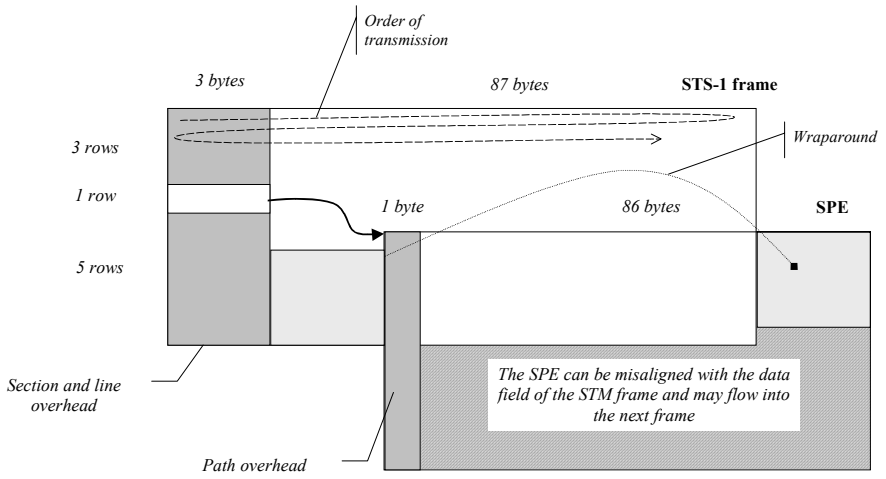


Figure 9.8 STS-1 (OC-1) frame format

Each SONET and SDH frame includes three levels of PCI, the *section overhead*, *line overhead* and *path overhead*. In an STS-1 frame, for example, the section and line overhead is contained in the 3-byte header at the beginning of each of the nine rows. The remaining nine blocks of 87 bytes, comprising the frame payload, are used to carry what is effectively a higher level PDU, called a *synchronous payload envelope* or *SPE*. This applies to SONET and SDH frames of faster carriers as well, although the rows are longer and the section and line overhead in each row takes 3*n* bytes in an STS-*n* carrier. The SPE is itself composed of rows, and the first byte of each row forms the path overhead. In an STS-1 frame, therefore, 9×4 bytes is PCI.

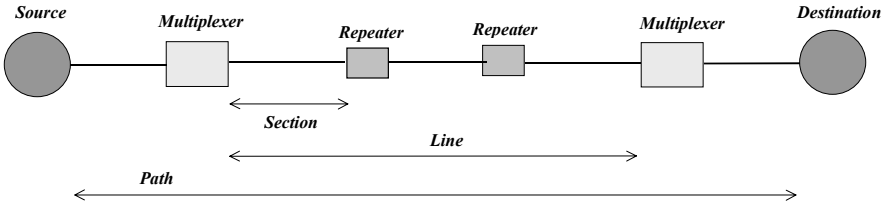


Figure 9.9 Structure of path in SONET/SDH

However, the interesting part is that SPEs do not in fact have to be aligned with the frame payload. A single SPE may therefore be split across two frames, as shown in Figure 9.8, with a pointer in the line overhead indicating the start of the unit in the frame. The SPE in a frame may carry the data from a single source, or data multiplexed from many slower sources. For, example, as we shall see in the next section, an ATM source running at STM-1 speed will simply fill its SPE container with ATM cells.

In other circumstances an SDH or SONET stream will be required to multiplex data from multiple sources, including PDH sources that are not precisely synchronised to the SDH/SONET stream. To facilitate this, a number of data blocks called *virtual containers* (*VCs*) have been defined. A VC of type 3 (VC-3) will fill the SPE of an STS-1 frame, while a VC-4 will fill that of an STM-1 frame. When multiple SDH/SONET streams are being multiplexed to form higher level carriers, they are simply interleaved byte by byte. An STS-3 (STM-1) stream may be formed from three STS-1 streams interleaved in this way.

For older PDH streams, these are first encased in STS-1 or STM-1 frames. To facilitate this, further virtual containers, with their own PCI overheads, are defined, which are smaller than the SPEs of any SDH/SONET frame. For example, the VC for a T1 source is VC11, while that for E1 is VC12. These are then embedded in VC-3 or VC-4 containers for transport in SDH/SONET frames. These VCs are sized so that they can contain the number of bits which a T1 or E1 signal can send in the 125μs period, plus some extra capacity to fit the container into an integral number of 9-byte columns. For example, a VC12 is composed of 36 bytes, which transmitted in 125μs, gives a rate of 2.3Mbps. However only 32 of the bytes are used for E1 data (2.048Mbps), one is reserved as a pointer and the others are reserved as stuffing bits. The other common PDH carriers, T3 and E3 are both directly conveyed in VC-3 containers.

SDH/SONET has great advantages, some of which we discussed earlier. It extends to much higher bit rates than PDH, and its synchronous nature allows a single multiplexed stream to be extracted or inserted without being forced to demultiplex the entire hierarchy (see Section 4.2.2). However, while, running on high-bandwidth optic fibre, it may solve the internal bandwidth problems of the telephone network, there remains the difficulty of how ordinary subscribers are to be given access.

The problem, of course, is the local loop, which, even conditioned for ISDN, is conventionally limited to 80kHz. The telecommunications providers would very much like to market high-volume data services such as *video-on-demand* and perhaps even HDTV, to their customers, as a way of competing with new cable operators, but are unable to do so at these speeds. One solution mooted, called *fiber-to-the-home* or *FTTH*, would be to replace the local loops with optic fibre links, but this would be exorbitantly expensive. Another possibility is *FTTN* (*fiber-to-the-neighbourhood*), where fibre is brought close to subscribers premises, but terminated at a local *optical network unit*, with the final run to the house or office being a limited-length copper loop.

However, recent research and development has shown that even without a huge investment in new cable, enormous gains can be made in the data rates used over most local loops. The actual copper wire pair that comprises the local loop actually has a much higher data capacity than is normally realised. The 3000kHz bandwidth imposed on modem operation is due to exchange equipment (filters) and artificially introduced inductive load coils. Without these restrictions, and with sophisticated signal processing, great increases in performance are possible, for, even though greatly attenuated, most local loop wires can pass frequencies of above 1MHz over even quite long distances. Considering that, using clever multilevel signalling, V.34 modems achieve data transfer rates of almost 10bits/Hz over the 300-3400Hz range of a voice

grade line, there is clearly great scope for improvement if bandwidths of 1MHz or so are made available. Of course, there are difficulties, since frequencies of 1MHz are attenuated by as much as 90dB over long lines, and in such circumstances, noise rules out the signalling constellations used in say V.34. However, for 24-gauge copper wire, widely used in local loops, the following practical limits (Table 9.2) which hold for simplex transmission may prove surprising.

Distance (km)	*DTR (Mbps)*	*Equivalent to*
5.5	*1.5*	*T1*
5	*2*	*E1*
3.5	*6.5*	*T2*
2.7	*8.5*	*E2*
1.5	*12.5*	*-*
1.0	*25*	*-*
0.3	*52*	*STS-1*

Table 9.2 Practical bandwidth of 24 gauge copper lengths

An ISDN line is in fact implemented by a run of copper pair between two so-called *digital subscriber line*, or *DSL*, modems, which transmit across the band from 0 to 80kHz. This clearly interferes with the standard POTS telephone service, which cannot be offered simultaneously on the same line (although analogue telephones can be used with ISDN TAs). Similar equipment can be used to offer two POTS connections over a single loop, by sampling the two analogue signals and then using TDM to cross the loop.

However, as can be seen from Table 9.2, even lines of 5.5km, the supposed maximum length in the ISDN standard, are capable of carrying much higher bit-rates if appropriate equipment is used. At a cost, a service called *High data rate DSL* (*HDSL*) is available, which allows the transmission of T1 up to distances of about 3.5km, using standard AMI signalling with only 1 bit per Hz, on two 24-gauge lines, or E1 on three. Even this, however, is below the practical limits for such lines. Recent advances in signal processing have permitted the development of DSL modems that can transmit an E1 rate signal over 3km on a single copper loop, and still leave space for a POTS service. This is called *single line DSL* or *SDSL*.

One problem with SDSL is that, unlike HDSL, it must use multilevel signalling, which is more susceptible to noise, particularly at the very attenuated levels involved. Some of this noise comes from crosstalk between wires that are bundled together as they emerge from the exchange. It turns out, however, that such crosstalk can be reduced if full-duplex data streams that are faster in one direction than the other are used. DSL modems have been developed which can attain rates of 1.5Mbps over 5.5km rising to 9Mbps over 2.5km (close to the rates in Table 9.2), in one direction, if the data coming the other way is limited to a much lower rate, ranging from 16kbps to 640kbps. Such modems, and a line offering this service, are called *asymmetric DSL* or *ADSL*.

Like SDSL, ADSL uses frequency bands that are above those used by POTS or even IDSN, and is able to coexist with these services as shown in Figure 9.10. The

term *upstream* refers to the subscriber-to-provider direction, while *downstream* is the opposite.

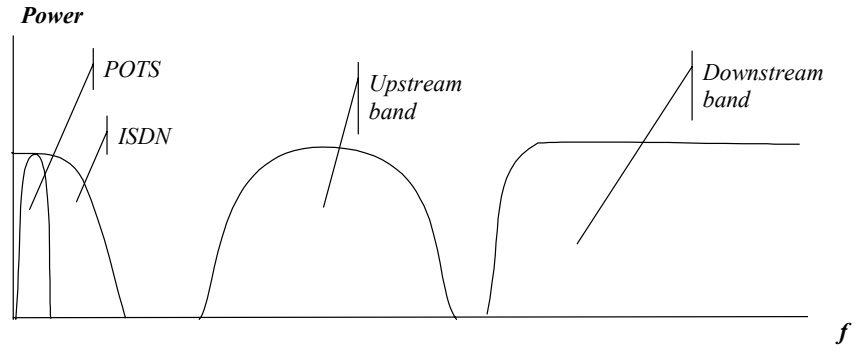


Figure 9.10 FDM separation of bands in ADSL

Many telecommunications companies are extremely interested in this technology, using the fast rate for delivering data, because, of course, this is exactly the pattern which many services, such as video-on-demand and Internet access, require. The advantage of ADSL over, say SDSL, is that it can attain rates of 6Mbps out to 3.5km from the exchange, admittedly in only one direction—but the useful direction. ANSI has, in fact, recently approved an ADSL standard, *T1.413* for rates up to 6.1Mbps.

ADSL is considered so promising because it satisfies the projected consumer requirements with the kind of line plant which is already in place, over the kind of lengths which cover the great majority of existing subscribers. It does not attempt to exploit the really high rates possible over shorter lines than 1.5km. However, if FTTN becomes available, much shorter copper loops will become the norm, and rates from 12Mbps to 50Mbps will be attainable. A service capable of delivering these rates is at a much earlier stage of development than ADSL, but is based on similar principles and is known as *very high data rate DSL*, or *VDSL*. VDSL may or may not be asymmetric, and offers the possibility of supplying HDTV over the telephone system, but its deployment is considerably further off than that of ADSL.

Both technologies are pushing existing plant to the limit, and error rates will be much higher than in FTTH, for example. When the data being carried is isochronous, forward error correction will be used, but this will involve delaying all data in the stream by perhaps 20ms or so, to give a chance to apply the correction algorithms. Such an additional delay is undesirable and unnecessary in data traffic, so ADSL, and presumably VDSL, will be able to deactivate the forward correction depending on the type of data in transit.

As with all technologies that have not yet seriously entered the market place, it would be premature to declare that ADSL will be successful. It will require careful marketing by the telecommunications companies, if it is to encourage sufficient subscribers to adopt it. Yet, it has one great advantage over previous marketing failures like ISDN—the companies driving its introduction need it to succeed!

9.4 ATM

9.4.1 Introduction to the architecture of ATM

In Section 4.5.4 we introduced the concept of cell relay and ATM, and mentioned that the term "ATM" has now come to mean the particular cell relay form standardised by ITU as the enabling technology for B-ISDN (e.g. in *I.361*). In this section we shall look at this technology in some detail, and henceforth, we will use the term in this specific sense.

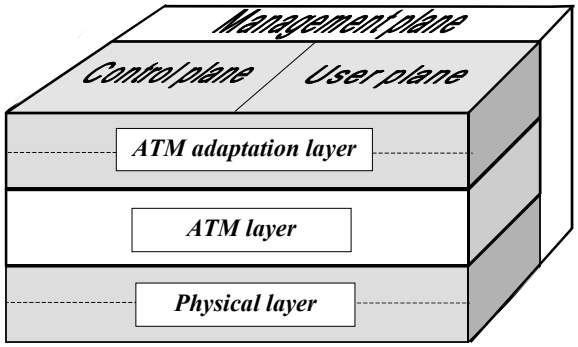


Figure 9.11 ATM reference model

The ATM project has been largely motivated by the telecommunications industry, as its anointed vehicle for the transport of all kinds of data through the public network of the 21st century. However its potential is such that it has also been seen as a possible candidate for the implementation of fast LANs and MANs. Its aim is the development of a streamlined network technology, defining the approximate equivalents of the three lowest layers of the OSI model, for both intranet communication and network access. ATM uses its own reference model which differs from OSI not only in the functional responsibilities of the layers, but also in that, like ISDN, it defines separate stacks or *planes* for control and data transfer (called, following ISDN, the *control* and *user planes*, respectively). In fact ATM also includes a third *management plane*, which is used for interlayer co-ordination and management of resources. The complete model is depicted in Figure 9.11.

ATM is connection-oriented with individual connections referred to as *virtual channel connections* or *VCC*s. In fact VCCs may be set up between two end users for data transfer, between an end-user and the network for control signalling, or between two intranet network entities for management and routing. A VCC is simplex, and two are necessary for duplex communications. Groups of VCCs with the same endpoints may be grouped together into *virtual path connections* (*VPCs*). Both VCCs and VPCs are allocated numbers, called respectively, *virtual channel identifiers* (*VCIs*) and *virtual path identifiers* (*VPIs*). As in X.25 and frame relay, a single VCC or VPC may be known by different numbers at different *links* on its path, with ATM switches making the necessary conversions: a link is basically a segment of a VCC or VPC with a single number. Once a VPC is established, the end points may divide it up into

VCCs of different capacities as they see fit, although of course the sum of all the VCC capacities in a VPC cannot exceed the total capacity of the VCC.

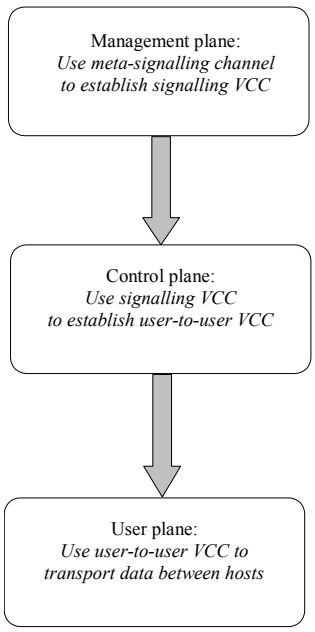


Figure 9.12 Setting up a VCC

As in ISDN, connections between users may be semi-permanent or switched. In the former case, no control setup is required, but in the latter, a user-network *signalling VCC* must be established first and then used to create the user-user data connection. These signalling VCCs are setup and maintained in the control plane, by means of a signalling protocol, *Q.2931*, a close relation of ISDN's *Q.931*. Of course, this begs the question as to how the signalling VCC is set up in the first place. In fact, it is done using a special permanent user-network VCC, which every host possesses, called a *meta-signalling channel*. The meta-signalling channel belongs to the management plane, and uses a simple protocol to establish the signalling VCC. Thus the formation of a switched user-user VCC is a two-stage process, as summarised in Figure 9.12. Entire VPCs can be set up at once, using the same approach. Note that VPCs can be set up by the network for its own purposes as well as initiated by users. If a VPC between two end-points already exists, the participating hosts can allocate their own VCCs within the VPC without further network intervention. To do this, they use the meta-signalling channel to establish a user-user signalling VCC, and then employ the latter to deal with each other directly so as to create a data VCC. Properties, such as the capacity of VPCs or VCCs, are negotiated when they are set up—the network itself ensures that the connections operate within the limits agreed.

VCCs are used to carry ATM cells, which they deliver in order. These cells are all 53 bytes long with a 5-byte header and 48-byte *payload*. There are a number of cell types, but the commonest, which includes all data cells, uses a relatively fixed

standard header format. In fact, this format is slightly different for cells passing between users and the network, crossing the so-called *user-network interface* (*UNI*) and those passing between ATM switches (*network-network interface* or *NNI*). The UNI cell header is shown in Figure 9.13—the NNI cell header is identical, except that the 4-bit *generic flow control* (*GFC*) and 8-bit VPI fields are replaced by a single 12-bit VPI field. UNI cells are transformed to NNI format at the first switch they encounter. Note that the bytes are transmitted most significant bit first.

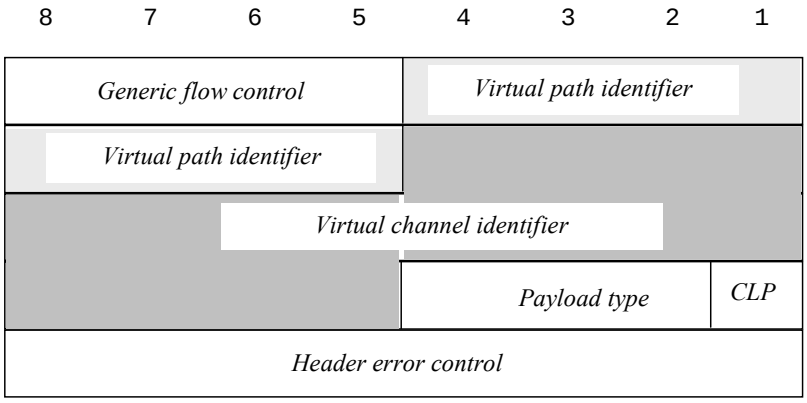


Figure 9.13 ATM user-network cell header

Examining the fields in Figure 9.13, we have already discussed the VPI and VCI. The GFC field is interesting in that its use is reserved at present, although as it is replaced at the first ATM switch encountered, it can only be of significance at the UNI interface. The *payload type* (*PTI*) field indicates the type of information carried in the cell. This is a 3-bit field, with the bits having functions as follows. The most significant bit distinguishing between user data cells (MSB of "0"), or cells for network management (MSB of 1). The second most significant bit is called the *congestion indication* (*CI*) bit and indicates whether a cell has experienced congestion on its journey. All cells start out with CI = 0, but this may be changed by any switch which is congested. The last bit, called the *ATM user-to-user* or *AUU* bit, distinguishes between two types of user cell, a feature which is helpful for some higher level services which ATM is required to support (notably the AAL 5 protocol discussed below).

The *cell loss priority* (*CLP*) bit performs virtually the same function as the DE bit in frame relay. If load shedding is necessary, the network will first discard cells with CLP = 1. Finally, the *header error control* or *HEC* field checksums the header of a cell only, using an error-correcting polynomial code of degree 8 (error correction is possible because the header being checked is only 4 bytes long).

All ATM data cells use this standard header format. However, as we have seen, ATM has control and management planes as well as the data-carrying user plane. The control and user planes are divided into the three layers shown in Figure 9.11, and while some management functions are layer-independent (so-called *plane management*) much management activity is also associated with the three user/control layers (*layer management*). At the ATM and physical layers, both control and

management functions use their own cell types, the latter being referred to as *operation and management* or *OAM* cells. Many of these control and management cells do not adhere to the header format shown, but have instead *preassigned headers*, which have fixed values for the first four bytes, although, admittedly, all have HEC fields. The physical layer, for example, can generate its own *physical layer cells*, as well as carrying cells passed to it by the ATM layer above. These include physical layer OAM cells, as well as so-called *idle cells*, whose function will be discussed below.

To understand the physical layer, it is important to realise that ATM is not tied to one physical transport mechanism. The standards discuss several possibilities, including direct transmission of cells over a cable or optic fiber, embedding the cells in PDH or SDH (SONET) hierarchical carriers, and even embedding them in FDDI. The underlying physical medium dictates features such as the speed ATM equipment must operate at, the error rates expected and so on. Despite this flexibility, the default carrier for ATM is expected to be SDH, and ITU Recommendations *G.707-709* describe how this is done at a standard rate of 155.52Mbps (STM-1), and a higher rate of 622.08Mbps (STM-4). Work is in progress on an STM-16 carrier at 2.5Gbps.

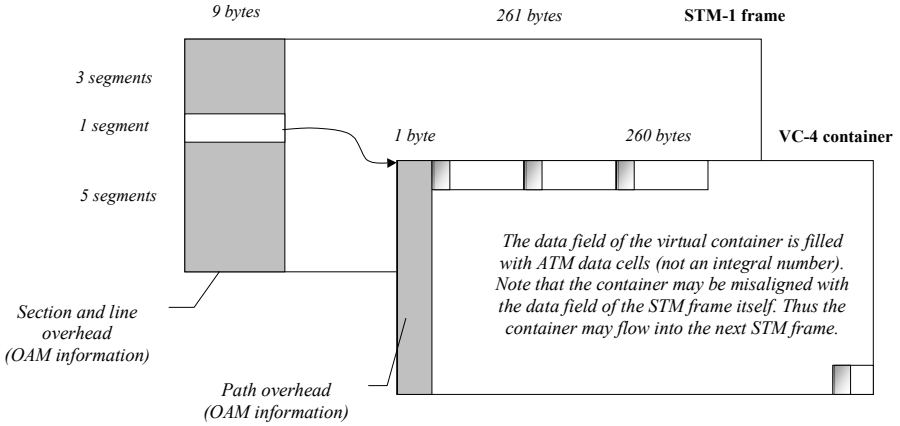


Figure 9.14 ATM cells in STM-1 frame carrier

STM-1 frames are transmitted at 8k frames per second, and are 2430 bytes long, with 90 bytes of control information and 2340 bytes of data, a ratio of control to data of 1:26. The ATM physical layer, when operating over SDH, constructs and transmits the SDH frames itself, filling their payloads with ATM data cells, and their control fields with OAM information (see Figure 9.14). This means that the actual rate of transmission of ATM data cells is only 26/27 of the STM-1 transmission rate. This also applies at the faster STM-4 and STM-16 carriers. Of course, being synchronous, the STM rates must always be maintained, so, if there are no data cells to send, the physical layer fills in with its own idle cells. Recall that the "asynchronous" in "ATM" refers to the way in which cells are filled from different data sources, and not to the underlying carrier bit-stream like SDH, which is of a bit-synchronous nature.

For the so-called *cell-based interface*, where no carrier is used and cells are simply transmitted in sequence, the standards specify the same transmission rates as for SDH, 155.52Mbps and 622.08Mbps. In this case there is no super-frame structure, but for every 27 cells sent, one must be an OAM cell generated by the physical layer, so that actual transfer rates are identical with the SDH interface. Again idle cells fill in when there are no ATM layer cells to send.

When SDH is not available, ATM can be used over PDH carriers at rates of 34 or 45Mbps (T3 or E3) and 1.5 or 2Mbps (T1 or E1). For "private" ATM interfaces, an interface to FDDI at 100Mbps is defined where cells do not actually have to be back-to-back but can be sent any time the line is idle. Finally, an ATM interface at 52Mbps, using SONET-like framing is defined for use over private UTP cabling.

The ATM physical layer is actually split into two sublayers. The lower *physical medium dependent*, or *PMD*, sublayer provides timing for the transmission and reception of bits and is different for the different underlying media. The upper *transmission convergence* (*TC*) sublayer accepts a stream of cells from the ATM layer and adds an HEC to each. It then constructs the correct *transmission frame* (e.g. an STM-1 frame) and adapts the cell flow to fit into it, inserting idle cells and OAM information as necessary. The transmission frame is then sent as a bit-stream to the PMD sublayer for sending.

In the receiving direction, the TC sublayer takes a stream of bits from the PMD sublayer and identifies the cell boundaries. Note that ATM cells have no delimiters, so where there is no guidance provided by the carrier, as for example in the cell-based interface, another technique must be used. The trick is to examine the relationship between incoming bits, until 8 bits are found which are a valid HEC for the previous 32. This should happen in any valid ATM layer cell but it could also happen by chance, so, to verify that cell synchronisation has been achieved, once the sublayer discovers such an occurrence, it checks again for a valid HEC, 53-bytes later. This may be repeated several times before cell synchronisation is confirmed. Continuous HEC checking, which is done anyway to verify header integrity, can thus be used to detect any loss of cell synchronisation.

The ATM layer, for whose protocols ATM cells are the PDUs, is responsible for generating the cell headers of outgoing cells and interpreting those of incoming cells. It extracts the VPI and VCI values, and uses them for multiplexing and demultiplexing cell flows at hosts. The layer is also present at nodes, where it translates the VPI value on one link of a connection to that on the next. The ATM layer provides a service whereby fragments of the correct size are fed to it and delivered at the appropriate destination, regardless of whether they are user data (user plane) or signalling information (control plane). It is the top layer present on ATM switching nodes, and is responsible for route setup (control plane) and the control of cell traffic.

The final layer in the hierarchy is the ATM adaptation layer, or *AAL*, defined in ITU *I.362* and *I.363*. This layer offers services to application processes directly, or through further communication layers, and some effort has gone into identifying generic types of service which can be used to support the many types of traffic that ATM is intended to carry. *I.362* lists four classes of service as follows.

- *Class A*, supports transfers at a constant bit rate over a connection in real-time (another way of saying, with imperceptible delays), supporting fixed rate traffic such as uncompressed sound and video.
- *Class B*, is like Class A, but with variable bit rates, such as the data generated from many compression techniques used in video and audio. This may require fewer network resources, but actually complicates the protocols required to cope with the variable rates.
- *Class C*, is like Class B, but relaxes the real-time constraint, and is intended for data that is not time-critical.
- *Class D* is essentially Class C but is connectionless.

To provide these classes, four *AAL protocols* were initially proposed, labelled *AAL 1* to *AAL 4*. However, the standardisation process did not proceed quite as anticipated. It was soon appreciated that AAL 3 and AAL 4 were sufficiently similar to be merged into AAL 3/4, and, even then, criticism was levelled at the latter for its PCI overhead. Further problems were encountered in the standardisation of AAL 2, which was not fully completed, and doubts were widely expressed about the entire service classification. More recently, a more efficient protocol, *AAL 5*, has been introduced, which is now touted as a replacement for, at least, AAL 3/4 and possibly AAL 1 and AAL 2 as well. It should be remembered in this discussion that the specification of ATM is very new and ongoing, and changes are to be expected in such circumstances.

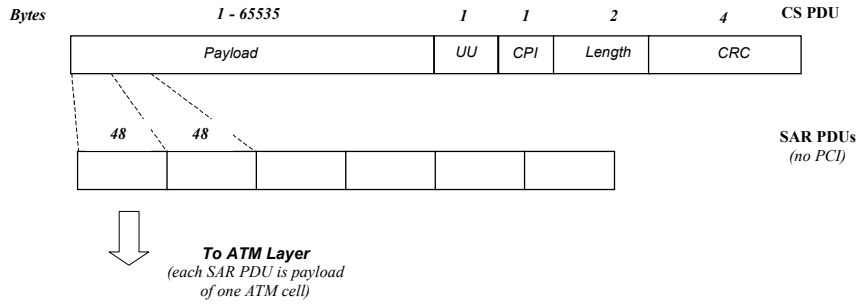


Figure 9.15 AAL 5 PDU relationships

In fact, the AAL protocols are all structured into two sublayers. The upper *convergence sublayer* which accepts data from the application and, in some cases at least (AAL 3/4 and AAL 5), forms its own PDUs. In AAL 1 and AAL 2, there are no CS PDUs, but rather a continuous stream of data, which the CS sublayer breaks up for the lower *segmentation and reassembly*, or *SAR*, *sublayer* to digest. The SAR sublayer formats smaller PDUs, each 48 bytes long, and capable of fitting into an ATM cell payload. The division and reassembly of CS PDUs in AAL 3/4 and 5 is done by the SAR sublayer, so the separation of function between the sublayers is rather fluid. Here we will only consider AAL 5 in any greater detail.

AAL 5 actually permits several options, but the most straightforward is the transmission of messages from one application to another as simple datagrams: this is

ideally suited, for example, to the transport of IP packets over an ATM network. These messages can be any length up to 65535 bytes, and are passed to the CS sublayer, where an 8-byte trailer is added. The CS PDU has no header, but the payload is padded to make the entire unit a multiple of 48 bytes, so the structure is as shown in Figure 9.15. There is a 32-bit CRC for error checking and a 16-bit *length* field which indicates the true length of the payload (without the padding bytes). The *UU* (*user-user)* byte is reserved for AAL service users, who can employ it to carry higher level information, while the *CPI* or *common-part indicator* byte is to be used to differentiate PDUs of this type from others, not as yet defined.

The CS PDUs are passed to the SAR sublayer where they are split up into an integral number of 48-byte units, each of which is passed to the ATM layer to be placed in the payload of an ATM cell. Note that SAR cells in AAL 5 have no PCI, and the receiver can only tell how many of these should be reassembled to reconstruct the original CS PDU, because the AUU bit in the PTI field of the ATM cells generated is set in the last fragment.

AAL 3/4 differs from AAL 5 in that the former incorporates PCI (including a short CRC), in each SAR cell, so that, in fact, each 48-byte SAR PDU carries only 44 bytes of data. This overhead has proved to be one of the major factors in the displacement of AAL 3/4 by AAL 5. However, AAL 5 is a datagram protocol and, if a PDU is lost, it is up to some higher-level transport protocol to engineer a recovery. Of course, even although the protocol is connectionless, the underlying ATM cells are being transmitted over a VCC at the ATM layer.

Before closing this section, it is perhaps worth saying a few more words about the control plane. Essentially, wherever possible, ATM tries to use the same underlying cell transfer mechanisms for all types of information, and the recommended control plane stack is now based on AAL 5. However, the control AAL uses a further protocol called *service specific connection-oriented protocol* or *SSCOP*, which can operate over AAL 5 and provides reliable transport of control messages, adding features such as selective repeat retransmission and sliding window flow control. SSCOP can be used to carry signalling protocol messages at both the UNI and NNI interfaces. The UNI interface protocol is Q.2931 as already mentioned, while the NNI interface uses the common channel signalling system no 7. In SDH applications, CCSS7 packets can also be transferred over the separate CCSS7 signalling network, as in N-ISDN, although this is clearly a less unified approach than using the ATM pathways themselves.

9.4.2 Traffic Control in ATM

Although the attempts to classify services at the AAL and to devise protocols to support these classes has proved somewhat abortive, the fact remains that, given the types of application which ATM networks are intended to sustain, a number of different types of traffic must be supported by the network. Remembering again that ATM standards are still under development, the most recent ATM specifications concentrate on matching the types of service reported to traffic types, and do not attempt to match these to specific flavours of protocol.

The isochronous service which I.362 called Class A requires a traffic pattern that is called *constant bit rate* or *CBR*. CBR is intended to emulate a smooth flow of bits across an ATM network, such as would be provided over a circuit-switched link or a

point-to-point line. Of course this apparent constant bit flow must in fact be engineered using ATM cells, arriving with acceptable delay and jitter to unload their payloads.

Real-time variable bit rate, or *RT-VBR*, traffic is generated by applications such as compressed video, where rather than transmitting a constant sequence of video frames, data reduction is achieved by sending only differences between frames, resulting in much more bursty flows. However, although bursty, there are still tight constraints on delay and jitter. In fact both for CBR and RT-VBR these constraints dictate that, should a cell be dropped or corrupted, there is really no point in attempting a retransmission.

Other traffic types are less demanding, in that they do not require real-time characteristics. *Non real-time VBR* traffic relaxes the constraints on jitter but still specifies low delay. *Available bit rate* or *ABR* is intended for asynchronous data with specified minimum and peak bandwidths to be supplied by the network provider. The network will always attempt to inform the users when congestion is a problem and will generally not lose cells so long as the users respond to such notification. Finally *unspecified bit rate* or *UBR* is low-priority, low-cost traffic, which specifies no minimum and gets no congestion information. UBR cells are sent with CLP bit set, and will be discarded by the network at any sign of trouble.

Access to these traffic types is gained by specifying a required *quality of service* (*QOS*) on each new VCC. This is done by means of a number of *QOS parameters*, such as:

- *Peak*, *minimum* and *sustained cell rates* (*PCR*, *MCR* and *SCR* respectively) are self explanatory;
- *Cell transfer delay* (*CTD*) gives an average and maximum cell latency*;*
- *Cell delay variation tolerance* (*CDVT*) is the maximum jitter acceptable;
- *Cell loss ratio* (*CLR*) is the fraction of cells delivered too late or lost completely;
- *Cell error ratio* (*CER*) is the fraction of cells with bit errors;
- *Cell misinsertion rate* (*CMR*) is the number of cells per second, delivered to the wrong destination (if, e.g., a VPI or VCI header field is corrupted and not detected).

A QOS can be requested by setting worst-case limits on any or all of these parameters. In CDR traffic, for example, a low CDVT is important, while in ABR or UBR it is not. In any case, if the network cannot achieve the QOS requested by a user, it will refuse to set up the connection.

Once committed, however, an ATM network is obliged to attempt to meet the QOS it has agreed with a user on a given VCC. Of course, this is nothing new: any public network must agree QOS with a subscriber when a connection is set up. It is, incidentally, much easier for a provider to make and monitor agreements on a connection-oriented than a connectionless basis, which explains why public providers favour connection-oriented service at the network level. What makes ATM different is that it tries to make promises about isochronous and RT-VBR traffic, which are much more difficult for an essentially packet-switched network to keep than undertakings about asynchronous data.

On the plus side, ATM networks use very high-performance switches and links that enable low and predictable delays to be achieved at least when the switches are not heavily loaded. However, if there is congestion in the network, this predictability is no longer possible, and CBR and VBR services will begin to fail. It is, therefore, even more important for an ATM than a more conventional switched network that congestion is avoided.

Even if congestion is not a problem, the fact remains that with a service like CBR, an ATM network is pretending to be something that it is not, namely circuit-switched. The very nature of cell transmission will introduce jitter, even before the data enters the switching network, since precise spacing of the cells sent by a host on a given VCC cannot in general be achieved. This is because cells from multiple active VCCs must be interleaved with each other, and with OAM cells, generated at the ATM and physical layer. A given cell on a given VCC can be delayed by an unpredictable, although generally small amount, before it even reaches the first ATM node on its journey.

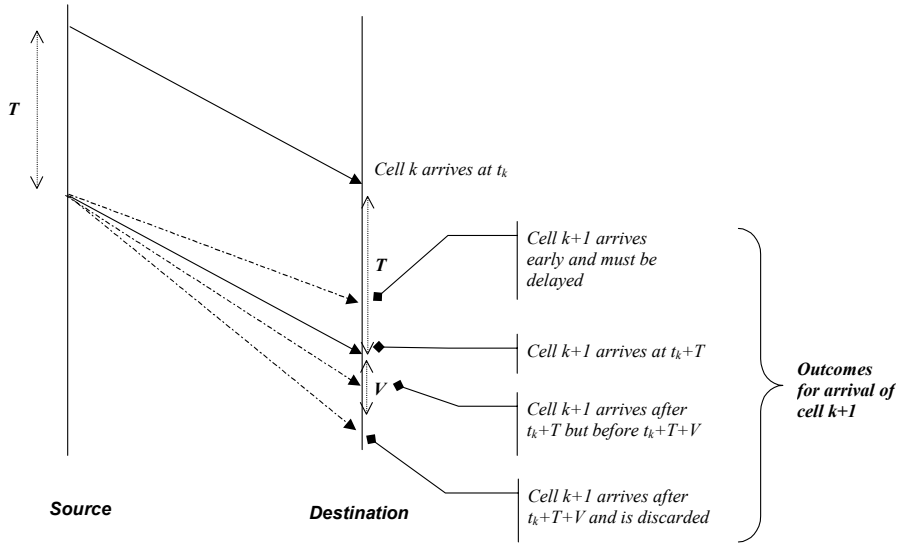


Figure 9.16 Cell delay variation

If a CBR source is sending cells at intervals of length T, and cell k arrives at time t_k, then ideally cell k+1 should arrive at $t_k + T$, although, in reality, it may arrive too early or too late (see Figure 9.16). If it arrives too early, it can easily be delayed until the correct time for delivery to the receiving entity. If, however, it arrives late, the provider must decide whether the CDVT (V in the diagram) has been exceeded or not. If not, it is delivered at once; if so, it is discarded.

To prevent this occurring, congestion must be kept under control. As in more conventional networks this can be done with certain types of traffic, by traffic shaping. Here, it must be ensured that users do not exceed the limits agreed on their access to the network, by, for example, attempting to send at greater than the PCR. To do this the network monitors traffic and discards cells which arrive much too early,

usually earlier than $\frac{1}{PCR} - V$, where V is the CDVT (note that the user is expected to observe the agreed CDVT for sending, just as the provider is for delivering). Essentially, the network expects cells to arrive at a certain rate, but is willing to accommodate slight increases up to a point. If that point is exceeded, the offending cell may be discarded or tagged by setting its CLP, marking it for discard if trouble arises further along the path. If a cell is discarded or marked in this way, the monitoring will behave as if the cell never existed (Figure 9.17). The approach is known as the *generic cell rate algorithm* (*GCRA*).

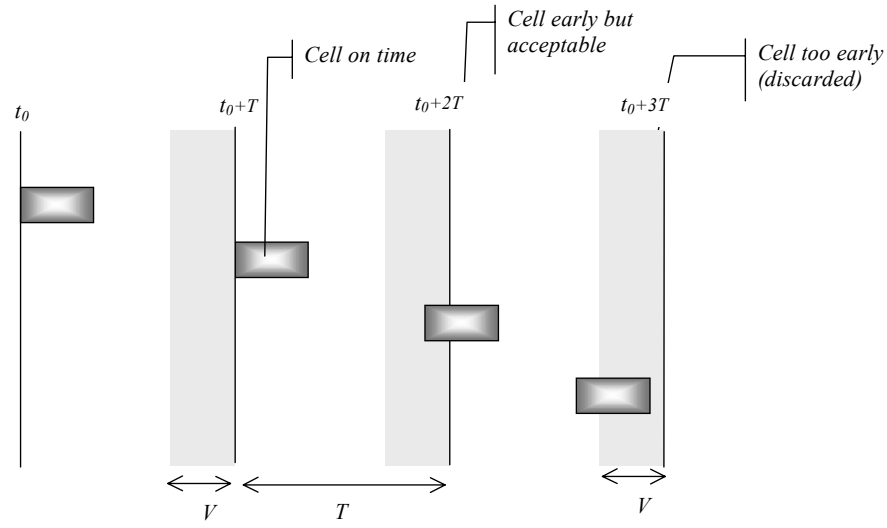


Figure 9.17 Cell discard as PCR is exceeded

The GCRA limits the load a circuit can put on the network and is an example of what is more generally known as *leaky bucket* traffic shaping. This allows traffic to progress at a strictly determined maximum rate (the rate of the leak), but provides a limited capability to absorb increases in this rate for a time (the bucket capacity): if the excess traffic exceeds a certain level, load shedding occurs. Traffic which exceeds the bucket capacity may be discarded at once or marked for preferential discard later Such techniques are clearly contingent on the careful monitoring of the QOS parameters of each connection, a process called *usage parameter control*.

Traffic shaping, however, only guarantees that a VCC or VPC will not exceed its agreed limits. To prevent congestion an ATM network may have to refuse requests for new connections if it cannot allocate the required resources. For CBR connections, in particular, the network must reserve necessary bandwidth along the path at set-up time. For other types of traffic, bandwidth, in terms of link and switch capacity, may be shared statistically, by, for example, allowing the sum of the required maximum bandwidths for all the VCCs on a given VPC to exceed the capacity of the VPC. If the VCCs are carrying traffic other than CBR, then it may be reasonable to assume that all

VCCs will not transmit at their maximum rates simultaneously—unfortunately, if this assumption proves incorrect, cells may have to be discarded.

In conventional congestion control, the common approach, when traffic shaping fails, is to ask senders to slow down (this technique is used in frame relay, for example). However, this is not possible with traffic like CBR and VBR that must meet delivery deadlines. For traffic in these classes, the senders cannot slow down, and the network's only solution is to begin load shedding. UBR traffic is low priority and will be discarded without notification if congestion is suspected. However, ABR traffic can be slowed down when necessary, and ATM networks implement a congestion control system, relying on feedback to senders, for this category.

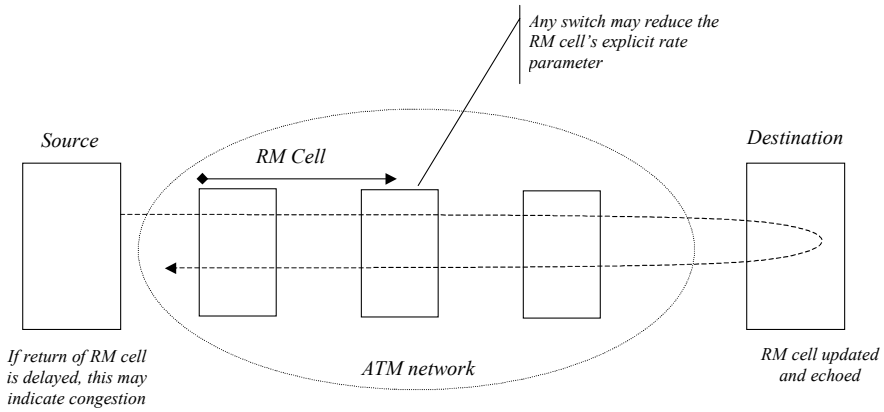


Figure 9.18 RM cells in ABR congestion control

The control mechanism relies on special OAM cells called *resource management* or *RM* cells, issued by the ATM Layer (see Figure 9.18). These are sent periodically on each VCC by each source host, then updated and echoed by the destination. In the RM cell, the sender indicates the cell rate at which it would like to send (up to its agreed PCR): this quantity is called the *explicit rate*. Any switch, or the destination, either on the outward or return trip, may lower this rate as the RM cell passes, although not below the MCR which the network has guaranteed. By the time the cell returns to the sender the rate has been reduced to a value acceptable to all points on the route. Incidentally, the congestion control mechanism for ABR traffic can also make use of the second PTI bit, which is used in much the same way as the FECN bit in frame relay. Of course, it is the destination, rather than the sender that sees this bit, but it can be echoed when the next RM cell arrives

9.4.3 ATM in WANs and LANs

The initial motivation for the development of ATM came from the recognition, in both the telecommunications and computer industries, that integrated wide area networks with much higher bandwidths than N-ISDN would soon be needed. ITU have been developing B-ISDN standards since the late 1980s, and envisage an integrated network, using ATM over a SDH or SONET carrier, offering bearer services in a manner similar to N-ISDN, but of much greater performance.

The *ATM Forum*, a cross-industry body established in 1991, now guides development, as opposed to standardisation, of ATM, a process, which, as we have seen, is still continuing. The adoption of B-ISDN technologies will undoubtedly be a slow and expensive process, and the ultimate goal of a universal ATM network which would entail replacement of the entire telephone system, may never be viable. However, ATM technology can also be used to support high-performance LANs and MANs, and it may be that this role will see the most widespread initial deployment. Even here, however, there is competition from more conventional technologies which have proved more adaptable than was at first suspected, and despite the efforts involved in its development, ATM is not guaranteed a future role as broad as was envisaged by its creators. As always, much will depend on the skill with which the technology is marketed and the extent to which the unwieldy international standardisation process holds it back. With these comments in mind, we will now look briefly first at the proposed B-ISDN, then at the emerging ATM LANs.

B-ISDN is ATM over SDH and we have already seen how these technologies merge. However, there are two issues that we have not yet tackled. First of all, how does the user interface to B-ISDN, and secondly, at the levels above AAL, what services will be offered. Tackling the former issue first, the B-ISDN interface bears a close resemblance to that of N-ISDN as shown in Figure 9.19.

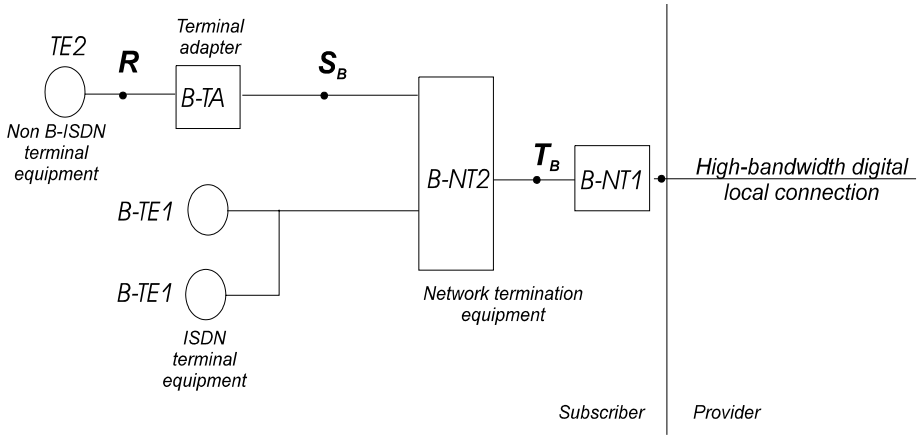
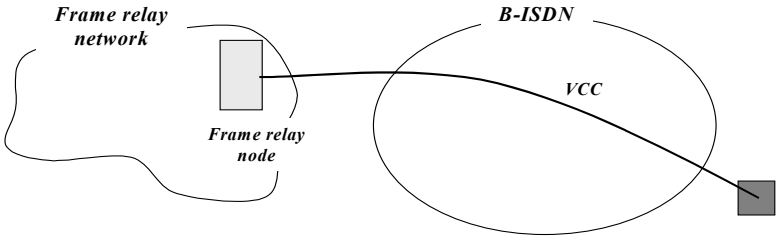


Figure 9.19 B-ISDN subscriber reference points

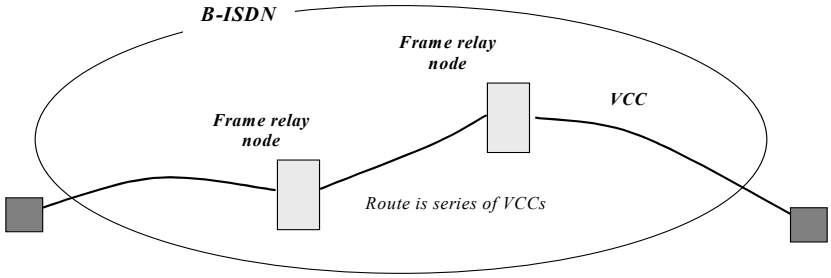
Comparing Figure 9.19 with Figure 9.1, the roles of the various functional units and reference points will be fairly obvious. As before, the *B-NT1* unit provides a minimal physical level transmission facility and handles physical level OAM. The optional *B-NT2* unit may be a PBX or other local switching facility, or it may be some kind of LAN interface. If it is a PBX, for example, it will include the functionality of all ATM layers, with buffering of cells, signal protocol handling, usage parameter control (traffic monitoring) and so on. The interfacing of B-ISDN TEs and non B-ISDN equipment is entirely analogous with N-ISDN.

B-ISDN will offer services that are a superset of those of N-ISDN. For example, consider how a B-ISDN might offer a frame relay service. Recall that frame relay is

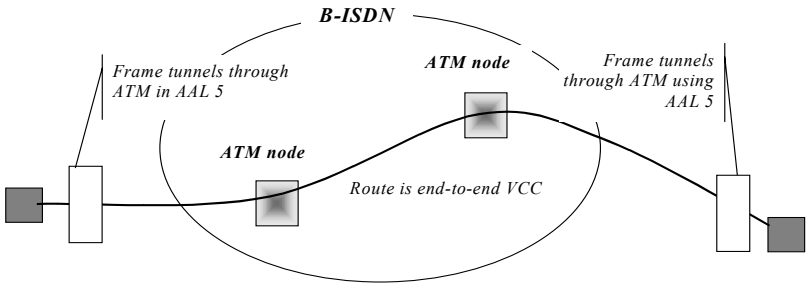
actually three things: a network interface specification, like X.25; a service offered to any compliant host; and a switching architecture. Viewed in this way, one can see that frame relay can be offered to B-ISDN subscribers in three distinct ways, as follows.



a) Separate frame relay network



b) Integrated frame handlers



c) ATM switching architecture

Figure 9.20 Frame Relay on B-ISDN

1. The frame handlers are not part of the B-ISDN but are accessed using it. A user wishing to access a frame handler simply attaches to it using a switched B-ISDN connection. The frame relay protocols run on top of the B-ISDN connection, but are invisible to the B-ISDN itself. The destination and other frame handlers need not be attached to the B-ISDN at all (see Figure 9.20a).
2. The frame handlers are integrated into the B-ISDN, and are again accessed directly using a B-ISDN connection. However, all the frame handlers and the

destination are part of the B-ISDN network, which arranges a route to the destination just as it would for an ATM VCC. The frame handlers, unlike ATM switches, will have an AAL (see Figure 9.20b).

3. Finally, the network can do away with the idea of a frame relay switching architecture completely, providing only a frame relay interface and service, but using ATM directly to switch the frames from source to destination. In this case, there are no frame handlers, and frames are simply carried using a protocol like AAL5. In effect LAP-F is tunnelled though AAL 5 (see Figure 9.20c).

Frame relay is a connection-oriented service, but except in the last implementation, the frame relay connections are maintained by the frame relay handlers and the ATM VCCs are simply used as vehicles to connect these together. If VCCs can be opened and closed quickly enough it may be possible to create them on demand, and still provide the frame relay layers with the illusion of connectivity. However, in reality, opening circuits on demand will typically add an undesirable overhead to the sending of a frame, and this can only be avoided if semi-permanent VCCs are kept open between handlers and subscribers. Since it is not reasonable for any user to have a large number of permanent VCCs open to different destinations, this may favour an implementation such as (2) above.

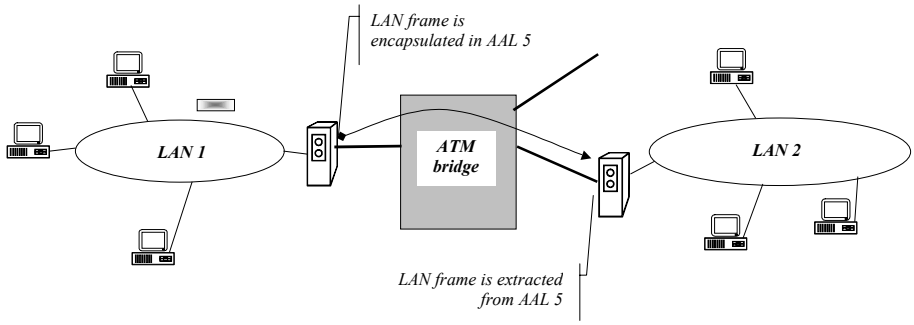


Figure 9.21 Bridging LANs with ATM

In other applications, however, a genuine datagram service is required, for example, if IP packets are to be transmitted over B-ISDN. This is best achieved by quickly establishing a VCC, transferring the datagram, then closing the VCC. The connectionless service, rather like the frame relay service, may be offered by a separate network to which the B-ISDN is connected, or by integrated connectionless routers. For example, IETF has studied this issue with regard to the transport of IP datagrams over ATM networks in particular, the results of the deliberations being published in *RFC 1483* and *RFC 1577*. ATM hosts using IP are grouped together into a *logical IP subnet* (*LIS*), containing a limited number of hosts in order to control the number of permanently open VCCs needed. One problem, however, is that a sender of an IP packet must have a way of associating the IP-number of a target with its B-ISDN number. This is analogous to the association of IP number and LAN address solved by ARP, and is tackled in a similar manner. Each LIS has a special server,

called an *ATMARP* server (ATM-ARP), which will provide the B-ISDN (ATM) number for a given IP-number. Inter-LIS communication may be achieved via IP routers which are members of two or more LIS sub-networks, just as in standard IP routing between subnets. However, this is actually quite wasteful when one considers that physically all hosts are attached to a single ATM network. Instead, it is possible to use another approach based on *next hop resolution protocol* (*NHRP*). Briefly, this uses *NHRP servers* (*NHSs*) to resolve IP address into ATM addresses. Each LIS will have its own NHS, and operating just like an ATMARP server. However, unlike an ATMARP server an NHS can resolve IP addresses outside its own LIS, by cooperating with NHSs in other LISs. To facilitate such cooperation, NHSs will have permanent connections to each other across the ATM network.

As mentioned above, many of the hopes for ATM lie in LAN and MAN, as much as in WAN, applications. Physically, ATM LANs and MANs will rely on ATM switch technology to implement connectivity. However, realistically, such ATM switches must coexist with IEEE 802 LANs such as Ethernet, and, in practice, will use AAL 5 to carry IP, IPX or CLNP packets. ATM can therefore be used to implement individual LANs and MANs, or as a bridging technology between conventional LANs (Figure 9.21). In the latter case, ATM switches forward LAN frames and must adopt similar strategies to 802 bridges, building spanning trees to allow the correct routing from one LAN to another. Of course an ATM network can also be very effectively used as a backbone to connect LANs remotely.

It is often desirable, in all these situations, to carry 802 protocol traffic transparently over ATM based structures of these types. The key standard for this is *LANE* (*LAN emulation*), developed by the ATM Forum, which allows a switched ATM network to appear to its hosts as an emulated 802 LAN (*ELAN*). As with direct implementation of IP over ATM, LANE involves the mapping of emulated-network addresses (in this case MAC addresses) to ATM addresses. The server playing the analogous role to an ATMARP is called a *LAN Emulation Server* (*LES*). In LANs, a broadcast capability is expected by many applications, and, of course, ATM is inherently a switched technology (although multicasting is possible). The LANE solution is another server called the *Broadcast/Unknown Server* or *BUS*, which has a connection open to every host, and can relay a packet received from one to all. The BUS is also used to broadcast packets whose destinations are not yet in the LES tables (c.f. ARP). Of course, an ELAN looks to the network layer, just like a real 802 LAN, so IP can be run over an ELAN, as an alternative to the LIS model discussed above. Inter-ELAN traffic relies on the use of routers.

If this all seems a little messy to you, then you are right. The way in which MAC and network protocols are handled over ATM networks is currently a bit of a patchwork. However, work is progressing on a unifying standard known as *MPOA* or *Multiprotocol Over ATM*, which will extend LANE (LANE v2 to be exact), to support traffic between ELANs using a similar approach to NHRP. It is intended that the MPOA model can deliver MAC or packet-level protocols across ATM networks.

An interesting question, which some sceptics have posed, is whether ATM is becoming over-complex, perhaps even in danger of incipient “OSIification”. When standards in a technology are churning over and being delivered piecemeal, there is a danger that potential adopters will be discouraged, believing that the technology is too

unstable. This may be true, and ATM may lose market share in LAN and short distance backbone applications to more dynamic, if technically less sophisticated, technologies like Gigabit Ethernet, because of it. However, no other networking technology currently challenges the wide area and long distance backbone performance of ATM, and, for this reason especially, the future of the technology looks assured.

9.5 SUMMARY OF CHAPTER 9

- Telecommunications providers conceived (narrowband) ISDN as a way of providing integrated digital voice and data services to subscribers.
- An ISDN network primarily supports circuit-switching using ISDN exchanges.
- Customers connect to ISDN using a digital pipe which can carry several types of channel, notably *D*, *B* and *H*.
- Basic rate ISDN (2B+D) may be provided over most local loop lines, so long as suitable exchange equipment is installed, and line inductances are removed.
- Primary rate ISDN operates over a T1 or E1 line to give respectively (23B+D) or (30B+D).
- At the customer's premises, the digital line from the local ISDN exchange is terminated by network terminating equipment. The NTE to exchange line runs at 160kbps in BRI, using quaternary signalling and echo-cancellation or TCM.
- ISDN terminal equipment connects to an S-interface of the NTE. This is a shared bus operating at 192kbps to transfer synchronous frames.
- Non-ISDN equipment interfaces to the NTE using special terminal adapters.
- ISDN supports circuit-switched, semi-permanent and packet-switched connections.
- The D-channel is used to establish circuit-switched connections, with data then travelling over a B-channel. The D-channel uses its miniature protocol stack, called the *C-plane*.
- ISDN may support packet switching integrated into the ISDN network, or simply by providing access to a PSPDN.
- Frame relay was conceived as a streamlined service and internal implementation to deliver packets, or rather frames, over ISDN.
- Frame relay is connection-oriented, and designed to carry asynchronous data, with no specific isochronous support.
- Frame relay networks rely on *frame handlers* which act like efficient cut-down packet switches, but operating at the data-link level using the protocol LAP-F. There is no internal flow control. If faults are detected, or in the presence of congestion, frames are discarded.
- Frame relay is supplied by non-ISDN networks as well. It is popular as a backbone technology for interconnecting LANs, although at lower rates than SMDS.
- The SONET and SDH synchronous hierarchies are now used within the telephone network, to attain higher rates and more flexible demultiplexing than PDH.

- A major problem for carriers is providing high-rate access for domestic subscribers. A number of technologies such as ADSL have been developed to yield such high-rate access over ordinary local loops.
- ATM is a standard form of cell relay using 53-byte cells routed through high-speed ATM switches. It is suitable for all types of data transport including isochronous.
- ATM is connection-oriented with end-to-end VCCs and VPCs.
- Like ISDN, ATM uses a separate C-plane mini-stack for control, and also provides a management mechanism in a management plane.
- ATM may be carried over SDH or over a no-carrier cell-based interface. It may also be used over PDH and FDDI.
- The transport of ATM cells is handled by the ATM layer. Over this is an ATM adaptation layer, which conditions data of various types for ATM transport.
- Traffic from different sources has different characteristics and requires different quality of service. The most challenging types are CBR and RT-VBR which originate from isochronous data flows such as uncompressed and compressed audio and video.
- To provide appropriate quality of service for these types, ATM networks must carefully control congestion, shaping traffic and discarding cells as necessary.
- ATM can be applied as a wide area service, as in the proposed B-ISDN networks, where it can be used to carry many types of traffic including frame relay or even IP.
- ATM can also be adapted for use in LANs and MANs, either as a total implementation technology or in a role for bridging between existing conventional LANs.

9.6 EXERCISES

9.1. In TI.601 the signal levels are used to encode bits in the order, 10, 11, 01, 00, rather than the more natural 11, 10, 01, 00. Why might this be?

9.2. If a subscriber line is longer than 5km, what type of equipment would be required to extend the range of the 2B1Q signal?

9.3. If a LAN is to be used in the NT2 function, it must have certain properties. Outline what these would have to be.

9.4. In the S-bus contention resolution algorithm, the wire-AND property ensures that if there is a collision, exactly one of the TEs will always fail to notice, and will gain uninterrupted control of the bus. Explain why.

9.5. In a frame relay network, a connection is routed through 5 frame handlers including the entry and exit handlers. A single frame message is sent along this route from source to destination TE, and is acknowledged by the remote end, under the control of a higher layer. Compare the number of point-to-point LAP-F frames required (a frame is a PDU sent from one TE or handler to an immediate neighbour) with the number of equivalent LAP-B frames sent in an analogous X.25/X.75 exchange.

9.6. Summarise all possible options for switched and permanent connections between two TEs using a frame relay network accessed via ISDN.

9.7. If a frame relay network uses a protocol like LAP-D or LAP-F control above LAP-F, does this eliminate the efficiency gains from using the stripped down LAP-F in the first place? Explain your answer.
9.8. Discuss the ways in which frame relay diverges from the conventional wisdom concerning link level protocols. Explain why it can succeed in spite of failing to adhere to convention.
9.9. What is the actual net data rate of STS-1?
9.10. How many T1 carriers can be conveyed in a STM-1 frame?
9.11. Why is it useful to allow the SPE in an SONET or SDH frame to be arbitrarily misaligned with the frame itself?
9.12. What are the advantages of restricting the ATM error check to the cell header?
9.13. What is the probability of the HEC-based cell synchronisation method registering a false sync if it is based on a single HEC identification?
9.14. The HEC-based cell synchronisation method violates a central principle of layering. Explain why.
9.15. Is HEC synchronisation needed in an SDH interface for ATM?
9.16. What is the maximum cell rate on an ATM link using the STM-1 interface?
9.17. AAL 5 uses an unusual PDU format with all PCI in the trailer. Can you think of any problems that might arise with this approach?
9.18. Discuss the relative merits of the three ways of implementing frame relay in a B-ISDN, with fast but non-negligible VCC setup.