Frame Relay Networks

Frame relay networks are a form of connection-oriented, fast packet network. They are based on the older X.25 networks and also intended to be public data networks. Some frame relay networks might also be considered as broadband networks, but few frame relay networks fit the definition perfectly. However, most frame relay networks are still fast enough in terms of network nodal processing delays and stability to deal quite well with compressed voice and video applications along with data. In a very real sense, frame relay is the result of years of effort to enable public packet-switching networks such as X.25 to handle *packetized voice* (and today also video).

This chapter will establish the overall structure of a frame relay network. The chapter will emphasize not only the physical structure of the network, but also the need to maintain an adequate quality of service (QoS) for voice and video services while at the same time handling extremely bursty data applications. This QoS discussion introduces the concepts of routing and switching. There is much talk today about "Layer 3 switching" in routers and "adding routing" to a fast packet switch in ATM or frame relay. This chapter is the place to explore the relationship between routing and switching once and for all. The chapter will also detail exactly how frame relay evolved from X.25 and how it still retains some evidence of this process.

The chapter will end with a look at frame relay connections. Both permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) are discussed. The need for both PVCs and SVCs is examined, along with consideration of the frame relay protocols that need to be implemented to allow for SVC service.

The Frame Relay UNI and NNI

A basic frame relay network is composed of three simple elements. The elements are the access link, the port connection, and the associated virtual connections (which are almost all PVCs today). The access link and UNI arrangement is such an important piece of the frame relay network service that the access link will be discussed more fully in the next chapter. This chapter will emphasize the frame relay switch port connection and virtual connections (or circuits).

Although the two elements of port connection and virtual connections will be discussed separately, the port connection and virtual connections have no meaning unless used together in a frame relay network. Think of the port connection as a hardware aspect of frame relay and virtual connections as a software aspect of a frame relay network. It takes both to do anything useful.

The relationship between the frame relay access link (UNI), port connection, and virtual circuits is shown in Figure 3.1. The access link runs between the customer premises frame relay access device, or FRAD, and the frame relay network service provider's switch. The port connection is the actual physical connection on the switch to which the FRAD device is attached by the access link. Finally, the virtual connections (PVCs) are what allow all user traffic from the CPE to be sent into the network on the same access link, yet be delivered almost anywhere in the world. In frame relay, the virtual connections are identified by a Data Link Connection Identifier (DLCI) which will be discussed further later on.

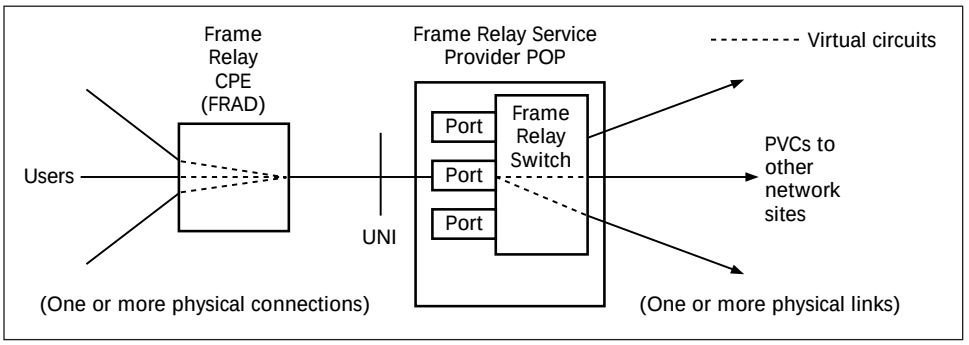

Figure 3.1 A typical frame relay network user connection.

The port connection forms the user entry point into a frame relay network. The port connection is usually associated with a single customer site, but not always. In other words, it is possible to have two sites linked to the frame relay network through one port, or even one site linked through two ports, but neither of these situations is common. In most cases, one site gets one port, no matter if there are many users, applications, or protocols sharing the network.

The key with frame relay is that many logical connections will share a single physical port connection. These logical or virtual connections (the PVCs) will carry traffic to many remote locations. And the nice thing about frame relay is that all of a particular site's traffic—regardless of originating users, application, or protocol—will use the same PVC to send and receive traffic to a particular site in the vast majority of cases.

In spite of all this connectivity, there is no dedicated bandwidth allocated to these individual users, applications, or protocols on the frame relay network. Dedicated bandwidth (all the bandwidth, all the time) is a characteristic of private line networks, but not of frame relay. Instead, the port connection on the frame relay switch will dynamically allocate the frame relay network capacity to meet the changing needs of all the users on the network, not just this one port. This is simply a way of saying that in frame relay, there is no dedicated bandwidth, but there is dedicated *capacity* on the network. The idea of dedicated capacity will be more fully explored in Chapter 4.

Capacity is determined based on the port speed. It determines the total amount of information a user sends into the network in a given unit of time (usually a full second). For example, a port speed of 64 kbps effectively allocates a capacity of 64,000 bits each second to *all* of the users attached to that port connection. There is no possible way that any user or application could ever send more than this number of bits into the network in a second.

Most frame relay service providers allow port connection speeds as a set of multiples of 64 kbps (56 kbps in some cases). These speeds are essentially based on something called fractional T1 (FT1) speeds. While it is not important to know exactly what this means, it is important to know what speeds are represented.

These speeds are usually 56/64 kbps, 128 kbps, 256 kbps, 384 kbps, 512 kbps, 768 kbps, 1024 kbps, and 1536 kbps. Some providers do not support all of these speeds and some support other speeds, but this set is very common.

Figure 3.1 showed only one site using a frame relay network across the UNI. To be more complete, there would need to be at least two sites and UNIs linked across the network. There must be at least one switch, and usually there are many. The switches link to each other over a network node interface, which is undefined in frame relay. Almost any protocol and hardware can be used, as long as it is supported by the switch vendor(s) of both switches and provides adequate Quality of Service (QoS) to users. Ironically, one of the most common uses of ATM today is to provide such a backbone for

frame relay switches to connect. When ATM forms the frame relay backbone, it is known as cell-based frame relay and has many benefits for service providers and users alike. The relationship between ATM and frame relay will be explored more fully later in Chapter 12.

For the sake of completeness, an entire (but very small) frame relay network is shown in Figure 3.2. Note that ATM is used as the backbone technology, but this is just one of the possibilities. In this case, the frame relay switches themselves become the end devices on the ATM network.

Public network standards typically spend a lot of time detailing exactly what should happen in terms of the software and what the hardware arrangements from the CPE to the network switch port should be. The CPE to switch interface defines the UNI and is clearly a key part of any network architecture. Without firm and complete UNI specifications, CPE devices could not be a varied as they are in terms of vendors and optional capabilities, and service providers could only support a small subset of all possible CPE configurations. The frame relay UNI allows this interoperability to take place.

But the frame relay network node interface is another story. Normally, the network node interface can be abbreviated NNI, but this is not a wise idea when speaking of frame relay networks. The acronym NNI does in fact exist in frame relay, but means *network-to-network interface.* So the frame relay NNI means something entirely different than the acronym NNI does in other technologies, such as ATM. Actually, ATM was the first major network architecture to define a standard network node interface. Most other network architectures, especially public network architectures, never bothered to define a standard network node (switch-to-switch interface). The reason is very simple: Such a standard interface on public networks was not felt to be in the best interests of the network.

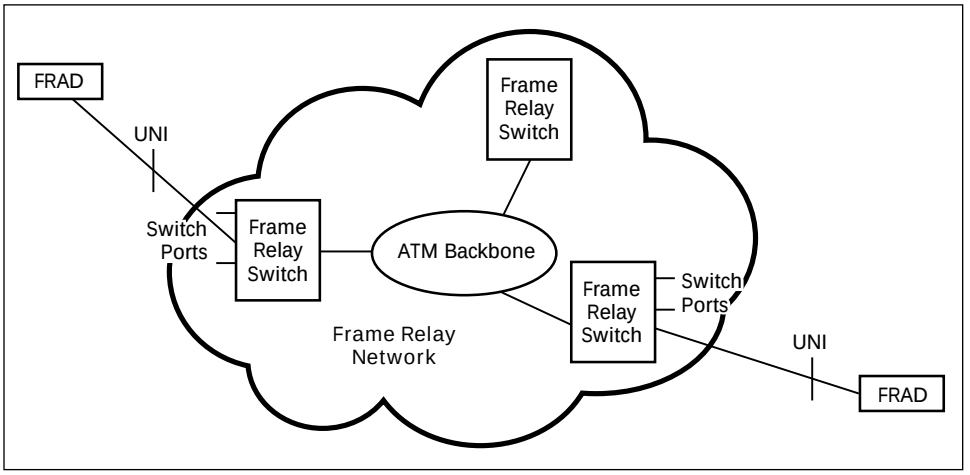

Figure 3.2 A frame relay network.

Such an attitude sounds quite odd given the current climate and push toward standardization at all costs. But this attitude grew out of the voice network and the philosophy was later applied to X.25 packet-switching networks and frame relay, among other types of networks. The approach to standardization on the public voice network did not emphasize interoperability. Instead, the approach emphasized innovation. The feeling was that if standards are too strictly defined, no one will ever do anything radically different, since obviously this new approach would not fit in with the currently defined standard. If standards are more loosely defined, then innovation can proceed with less concern for interoperability.

Consider the PSTN as an example. Once people could buy their own telephones, the interface from telephone to voice switch was fully and strictly defined, right down to the voltages. But there was still no standard way for the voice switches to talk to each other. Each central office or local exchange switch vendor had its own, proprietary way of signaling and trunking between switches, and each felt that its way was the absolute best possible way of performing this task. This situation encouraged vendors to freely innovate and explore other methods of switch interfaces, since the only concern was for backward compatibility with their older products, at least until the older switches could be phased out.

But what about interoperability? Proprietary voice switch interfaces meant that a multivendor environment was difficult to achieve. If it had to be done, vendor A's switch had to translate everything into vendor B's talk or vice versa before any interswitch communication could take place. And this translation process is exactly what was done. At first, it would seem a chaotic situation, especially to those used to a standards-dominated world. What saved the PSTN was the fact that there were only about a half dozen public voice switch vendors, so multivendor translation was not as big a problem as it would be in the LAN world with 60 or more Ethernet hub vendors.

Large public networks like the PSTN were seldom multivendor environments anyway. There were few alternatives, as just mentioned, and no one cared to build a network where intervendor finger-pointing between the switch vendors at each end of the link slowed troubleshooting and repair times to a crawl. The customers (and regulators) would not stand for it. So most large public networks standardized on one vendor or another, and that was that.

The proprietary approach was extended to X.25 public data networks, then to frame relay as well. So frame relay switch vendors are free to innovate any way they choose on the network node switch-to-switch interface. The only real requirement is that the two ends of the link understand each other.

Ironically, in spite of the lack of standards for use on the network node interface between frame relay switches, there is one standard that is forbidden for use between frame relay switches. This is the frame relay UNI. The precise reasons for this are beyond the scope of this discussion, but this prohi-

bition revolves around the fact that two frame relay switches are total peers, and the UNI requires one end of the link to be CPE. The UNI relationship is a peer one with regard to data transfer, but not so with regard to network management and so forth.

How did ATM become a common backbone for frame relay networks? One major reason is that the pressure today in the industry is not toward innovation but toward multivendor interoperability. So proprietary interfaces, while tolerable, are not always the first choice. Also, as services grow, there are more network nodes than ever. Multivendor environments are more common in the data world, where a huge switch has 256 ports, not 10,000 or even 40,000 as on a large voice switch. Therefore, if no standard network node interface exists, it might still be a good idea to use something else that is standard to tie all of the nodes together. That is one role of ATM in a frame relay network. ATM provides the standard network node interface that frame relay lacks. Each frame relay switch essentially becomes a *user* on the ATM network.

There is much more to the relationship between frame relay and ATM than just an ATM backbone for frame relay switches. But the positioning of ATM as backbone for frame relay is enough for now.

In spite of the previous discussion, there is an NNI acronym in frame relay; NNI means Network-to-Network Interface. The frame relay NNI is used whenever two different service provider's frame relay networks need to communicate. After all, they might be using different switch vendor's products and proprietary interfaces will not work. And although multivendor public network environments are not all that common, they are not particularly rare either. For instance, a service provider might change switch vendors at some point. It would hardly be possible or intelligent to discard the previous vendor's equipment. The vendors might be isolated by area or function, but the switches must still communicate when required. Translation can be used in this situation, but there are potentially more frame relay switch vendors than voice switch vendors. And the more vendors, the more the need for standard interoperability between them. Some standard way must be found in order to allow two different frame relay networks, or portions of networks, to communicate.

The relationship and uses of the frame relay UNI, NNI, and interswitch interfaces are shown in Figure 3.3. Note the presence of a simple router as a FRAD. The possible FRAD configurations are a major theme of this chapter and the next. The other FRADs in the figure have multiple ports that connect other devices, probably routers, but also other things, especially IBM SNA components.

The figure shows a private frame relay network as well as public frame relay. Nothing prevents an organization from buying and installing its own internal frame relay network. Only leased private lines are needed to tie the switches together and link the FRADs to the switch ports. But if the private network needs to access users on a public frame relay network, there must be

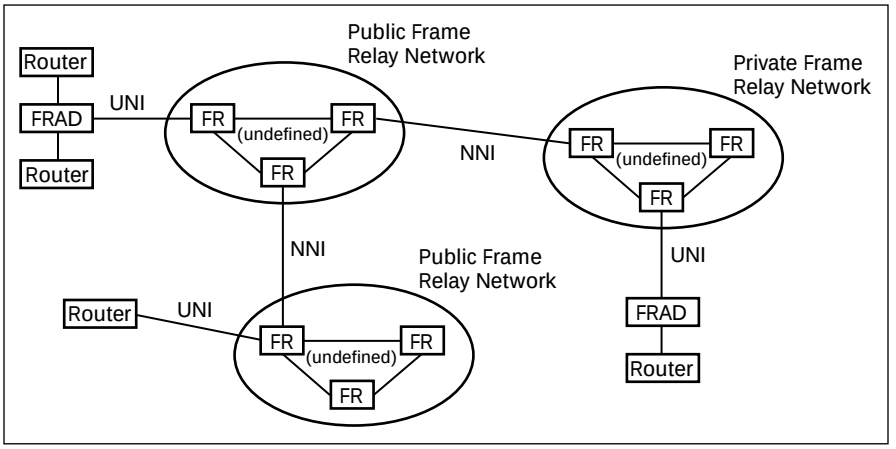

Figure 3.3 Frame relay UNIs and NNIs.

a standard interface between them if the switch does not understand each and every proprietary protocol in use. This is one role of the NNI.

The primary role of the NNI is shown in the figure also. The two public frame relay networks could belong to two frame relay service providers, perhaps a LEC and an IXC. Alternatively, the two public frame relay networks could belong to the same service providers, and could even service the same geographical area. But in this case, the two clouds contain all of vendor A's switches in one cloud and all of vendor B's switches in the second cloud. This is a job for the NNI as well. Both uses are common.

The Frame Relay Access Device

The Frame Relay Access Device (FRAD) is the user's view of the frame relay network. Users do not see the frame relay switch, nor do they usually even see the UNI. What users see is the FRAD. And even then, the FRAD might very well be the same familiar router down the hall. But in many cases, the FRAD is a special frame relay network device that terminates the UNI at the customer site; it is the CPE of the frame relay network. A lot of times, evaluating a frame relay network is really a two-step process. First, examine and choose the service provider. Then, examine and choose the FRAD vendor(s) (it is much easier to mix and match FRAD vendors than frame relay switch vendors).

A quick definition of a FRAD is easy to provide. A FRAD has at least one port capable of supporting the hardware needed for the UNI link and the software needed for understanding frame relay protocols, and one or more non-frame relay ports, usually LAN ports. So a FRAD has at least one UNI and one

or more non-UNI ports. Many FRADs, especially smaller, less expensive models have exactly one UNI port and perhaps four non-UNI ports, all usually just 10Base-T Ethernet. More expensive FRADs have more sophisticated UNI options and configurations, including dial backups, and a wider range of non-UNI ports and/or more of them as well.

The point of this section is to describe the different types of FRADs that can be found on the premises of a typical frame relay customer. This does not mean that several criteria for deciding which FRAD is right for which situation are not covered here, it just means the emphasis is on description, not selection.

It is possible to divide all of the FRADs marketed by vendors into roughly the following categories and subcategories:

- Software FRADs (Routers)
  - FRADs with only basic features
  - FRADs with more advanced features
- Hardware FRADs
  - FRADs (Traditional FRADs)
  - M-FRADs (Multiservice FRADs)
  - V-FRADs (Voice FRADs)

So FRADs fall into two major categories, software FRADs and hardware FRADs. In each major category, several variations are possible. Each of the major types of FRAD is discussed in a little more detail here. Several of the more advanced features will be explored in more detail in later chapters. This section only indicates support options.

Software FRADs (Routers)

Most frame relay services are used to link organizations' LANs together. Even when frame relay is used for support SNA, PBX voice, or some even more exotic services, the basic LAN connectivity *private line replacement* role of frame relay is still present. The device that is most often used to act as the external *gateway* between LANs is the router. In fact, *gateway* was the older term used for router until a company called cisco (properly spelled with a lower case "c," but seldom seen that way) essentially invented the router and, more importantly for the LAN interconnection industry, the *market* for routers. This is not necessarily an endorsement of cisco routers, but more of an acknowledgment that cisco outsells every other router vendor put together.

Routers are the network nodes of the Internet and have become the common term for the network nodes for LAN interconnections of any type, from leased private lines to virtual private networks. Older LAN connectivity schemes used *bridges,* but these devices had such limitations when compared

to routers that once the pricing was right, people took advantage of the benefits of routing almost immediately.

Note that the function of routers on LAN internetworks and the Internet is exactly the same as the role of switches in a public data network. Both are network nodes. This is not a coincidence. Some would even claim that there is basically no difference at all between routers and switches. More on this subject will be said later in this chapter, where the position is developed that there are still some consistent differences between routers and switches, but the differences are becoming less and less over time.

Because both routers and frame relay switches are network nodes, it might seem logical that they should be able to interface directly. And they can. But since the router is not a frame relay switch, but a CPE device, the interface between them must be the UNI. Since the router is also a device that has one or more non-UNI ports for LAN attachment, it is easily seen that the router fits the one-UNI, one-or-more non-UNIs definition of a FRAD and can perform the same function as a FRAD. When a router performs this function, it is sometimes known as a software FRAD; that is the term used here.

Why *software* FRAD? Because a router typically has at least one serial WAN port that runs an appropriate WAN protocol such as PPP (Point-to-Point Protocol) at the frame level (Data Link Layer 2) that the serial router port on the other end of the link understands. If the serial WAN port is now to be used as a frame relay UNI, PPP will no longer do. The protocol that runs at the frame level now must understand frame relay frames and nothing else. This is not usually a problem. All major router vendors (a phrase that can just mean "cisco," but in this case means almost everybody) support and bundle the frame relay protocol on their serial WAN ports.

The use of a router as the CPE on a frame relay UNI is quite attractive. This is especially true if the frame relay network is replacing an existing private line network between the routers and many frame relay networks do. What usually happens is that the customer can terminate service on all of the other links, which are usually 56-64 kbps private lines, and keep one to form the UNI. The router port is reconfigured as a frame relay UNI, restarted when service begins, and the UNI is up and running. What could be simpler? This *graceful migration path* aspect of frame relay must not be underestimated.

It should be noted that with the exception of the frame structure, the UNI link performs exactly as it did before it became a frame relay UNI. That is, the link carrier frames are represented as bits. But now the link terminates at a local frame relay switch instead of at another router port hundreds or even thousands of miles away. Because private lines are paid for by the mile, not only are there fewer links in the frame relay network, but also they are a fraction of their former length (and price).

The UNI still requires a standard Digital Service Unit/Channel Service Unit (DSU/CSU) arrangement on the premises if the UNI is a digital link such as a

56-64 kbps DS-0, which the vast majority of UNIs are. The DSU/CSU forms the boundary between the service provider network and the CPE (the router or FRAD). The position and function of the DSU/CSU on a frame relay UNI is shown in Figure 3.4.

The DSU/CSU takes bits having one type of coding suitable for short distances and some media sent on the serial port, and converts them to and from bits that are suitable for longer distances and other media. The DSU/CSU is strictly a Physical Layer 1 device in the OSI RM. The function and position of the DSU/CSU is worth mentioning because the CPE is customer property and cannot be directly managed or configured by the service provider in most cases. But the DSU/CSU can be directly managed, since it is seldom changed. This makes the DSU/CSU an attractive place to try to manage frame relay UNIs; several service providers have attempted to do just that.

Software FRADs do not even have to be routers, but most of them are. Almost any device that has a serial port can be used as a FRAD, as long as the vendor provides or supports software to generate frame relay frames and understands the switch at the other end of the UNI (there is more to frame relay and networking in general than just data transfer). Bridges, mainframe Front-End Processors (FEPs), minicomputers, and other equipment can be software FRADs, and many are, especially IBM SNA FEPs. But SNA and frame relay are a topic for a later chapter.

Previously, software FRADs were distinguished by the presence or absence of advanced features. In the case of software FRADs, *advanced features* can translate to many things that hardware FRADs offer routinely and easily. Most routers that function as software FRADs, as convenient as this is when frame relay is used for LAN interconnectivity and private line replacement (which is just about always), are unable to provide more than simply a way to package

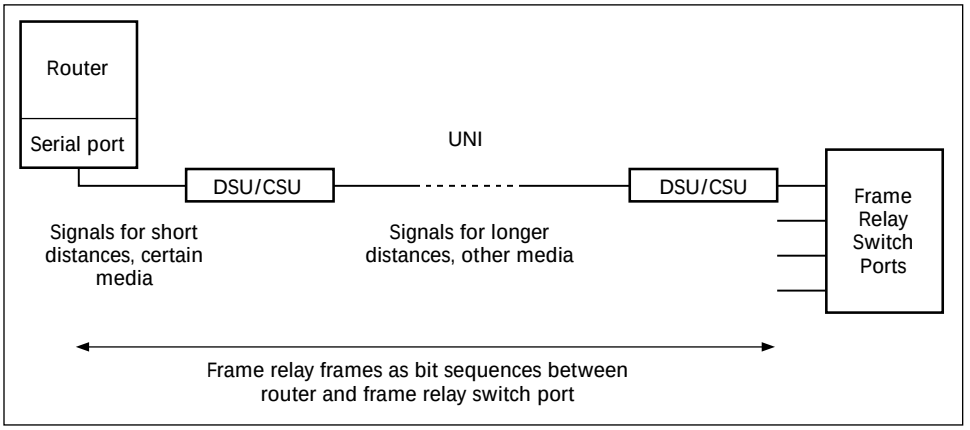

Figure 3.4 The DSU/CSU on a frame relay UNI.

LAN frames or packets into frame relay frames on one side of the network and haul them out again on the other.

However, it has already been pointed out that there is more to networking than simple data transfer. The basic frame relay functions will transfer data across the network, but how will the routers detect congestion on the frame relay network? How will the routers deal with missing frame relay frames that are dropped because of errors? Most software FRADs do not deal with these issues; in fact, they are not really worried about these issues at all. Routers simply route. Let the end systems worry about what to do about errors and congestion. The problem is that routers *are* the end systems to the frame relay network. But simply putting frame relay software in a router will not necessarily make the router a particularly good FRAD.

Nevertheless, the use of a router as a software FRAD is common and accepted. There are several benefits to this use of routers. First, the frame relay software is usually bundled with the router, at least all but the very low-end routers. So no extra hardware is needed and the cost of a separate FRAD is not incurred. One router is typically the gateway off the premises for network traffic, so making this the gateway to the frame relay network makes sense also. Certainly this is the simplest configuration. Finally, routers have been around for a number of years; there is widespread understanding and support for routers in the networking community.

Sometimes, the limitations of software FRADs only become apparent when the frame relay network becomes so successful that frame relay network access must be expanded to more users, perhaps all of them. The biggest limit is that there can only be one frame relay UNI per router, unless definite steps are taken to provide more than one UNI. Many customers, familiar with private line environments, do not think about having more than one UNI per router. Also, many router-based FRAD implementations are not totally compliant with frame relay specifications. The router might be able to handle basic frame transfer, but not much else. There are usually few really advanced frame relay features on the router acting as a FRAD, which is only understandable. Routers are built and marketed as routers, not as FRADs, after all. Finally, because routers are basically connectionless, *best-effort* packet delivery platforms, there is little quality of service (QoS) support in a software FRAD. In this context, QoS means that the application is able to obtain the bandwidth, error rate, and delay characteristics that the application needs to function from the underlying frame relay network. QoS in networks, in general, and frame relay QoS, in particular, will be defined more fully in a later section of this chapter.

In fairness to the router industry, it should be noted that nothing prevents a router from becoming as good a FRAD as anything else, except perhaps in the area of support for other services. That is, voice telephony ports on a router will be rare for the time being. The strength of software FRADs depends on the amount of effort put into them by the individual router vendor.

Hardware FRADs

Many of the basic features of FRADs have been covered in the software FRAD section. A lot can be covered by contrast rather than detailed descriptions. That is not to say that hardware FRADs are not as important a topic as using routers as FRADs; it is simply a reflection of the natural tendency to use the familiar router in a role that can also be served, and sometimes better served, than a separate, dedicated frame relay network access device.

Today, the FRAD marketplace is broken up loosely (very loosely) into three major categories: "traditional" FRADs, multiservice FRADs, and voice FRADs. The term "traditional" refers to FRAD as a stand-alone hardware device which might have very advanced state-of-the-art capabilities. In fact, the perspective employed here on these divisions is not based on any formal definitions at all, only the author's individual view of the marketplace. Formal definitions may evolve, but for now the dividing line between the various categories of FRADs remains quite fluid.

Traditional FRADs

It is surely a measure of how far hardware FRADs have come that it is now necessary to call the simplest packaging of FRAD capabilities a *traditional* FRAD. Although the leading edge of the market has progressed far beyond the simple packaging of the basic FRAD, such traditional FRADs remain in heavy use on many frame relay networks. The basic package should remain common in low-end FRAD offerings for many years to come.

The term "traditional" applied to hardware FRADs has nothing to do with frame relay features that determine frame relay standards compliance or support for options. All hardware FRADs offer such compliance (at least they should) with frame relay standards; option support is another issue altogether. Rather, the term "traditional" applies to a FRAD that only supports data services and treats all data traffic identically in the FRAD itself (i.e., there are no priorities for individual virtual circuits).

The major features of a traditional hardware FRAD are shown in Figure 3.5. The main difference between a hardware FRAD and a software FRAD, such as a router, is support for more types of non-UNI ports on the FRAD than only LAN ports, although such a FRAD might still have nothing but LAN ports, but generally more than one. Even X.25 or telex ports can be accommodated on some models of this type of FRAD.

One of the distinguishing characteristics of a traditional FRAD is that traffic from all non-UNI ports is treated exactly the same inside the FRAD itself. In other words, if two frame relay frames containing client/server LAN traffic are already waiting to be sent on the UNI into the frame relay network, and a port connected to an IBM AS/400 generates a delay-sensitive unit of SNA traffic rep-

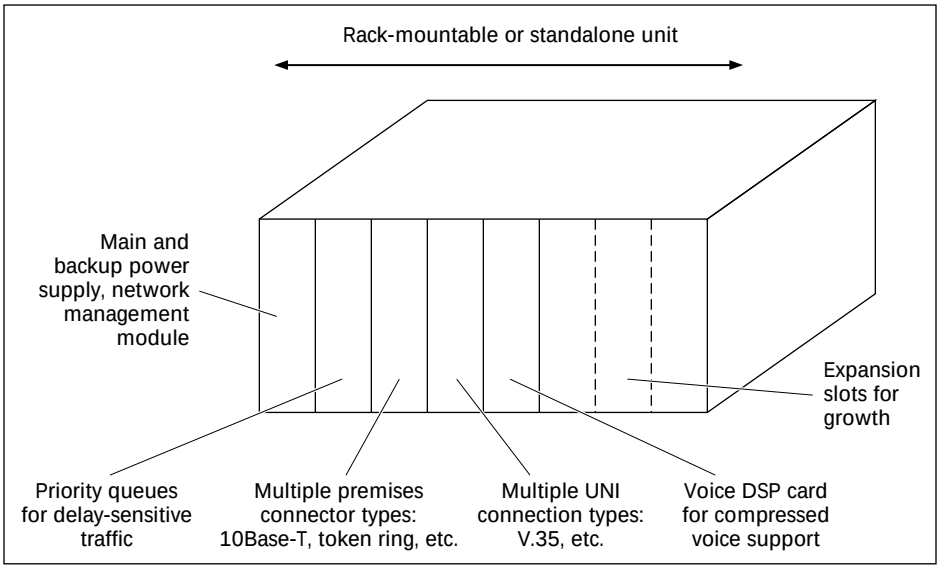

Figure 3.5 The stand-alone FRAD with advanced state-of-the-art features.

resenting a financial transaction, there is no way for the SNA traffic to *leap frog* and thus gain preference in the output queue over any other traffic in the FRAD.

So from the traditional FRAD perspective of the frame relay network (which is basically all frame relay network levels since the FRAD is the interface to the frame relay network), all traffic is created equal. Under light loads, traffic moves through quickly, transactions and e-mail alike. Under heavy loads, traffic moves through more slowly, perhaps slow enough to result in SNA session restarts. SNA session restarts slow transaction processing to a crawl and make more work for all the components of the network, frame relay and non-frame relay components alike. Perhaps if some way to distinguish bulk e-mail traffic from delay-sensitive SNA transactions within the FRAD were possible, the frame relay network users would be a much happier group. E-mail could easily wait while SNA session traffic was delivered more quickly.

Perhaps it would be better if there was a way in the FRAD itself to acknowledge inherent differences in the type of service that an application needs and prioritize the virtual connections on the network. This FRAD would not only prioritize connections, but also dynamically allocate more or less bandwidth to an application as the application uses the frame relay network. This type of FRAD is sometimes called the M-FRAD, or multiservice FRAD.

Multiservice FRADs

The first thing that should be said about M-FRADs is that they are not necessarily the same as *multiservice access concentrators*. M-FRADs support frame

relay UNIs, pure and simple. Multiservice access concentrators generally support both frame relay and ATM UNIs. M-FRADs still support delay-sensitive traffic streams like voice and video only if these services are delivered to and originate from LAN-attached PCs. So multiservice support in an M-FRAD still revolves around *data* packet delivery. Multiservice access concentrators have voice and video support, but the voice and video support is usually handled by specialized ports with not only non-UNI interfaces but non-LAN ports as well. This is not to say that a multiservice access concentrator with a frame relay UNI and only LAN ports cannot be used in the same fashion as an M-FRAD. It simply acknowledges that the multiservice access concentrator is a more general device while the M-FRAD is a more specialized device.

The key feature and benefit of an M-FRAD is that the device can prioritize the frame relay service given to individual connections based on the needs of the traffic on the connection. Typically, this need is for data traffic priorities, but nothing prevents M-FRADs from supporting voice traffic as well—usually by encapsulating digital voice inside data packets. It is hard to be precise when there are no accepted definitions and a large measure of common sense is needed when evaluating these types of FRADs.

Before M-FRADs, almost all FRADs had a few common characteristics. Some of these features have been discussed already. First, these traditional FRADs provided a single type of service (first in, first out) for all traffic. Next, the number of UNIs and non-UNI ports serviced by the FRAD were relatively small, so central site concentration was awkward. Finally, these FRADs all relied solely on permanent virtual circuits (PVCs) for connectivity.

Usually, the M-FRAD at least distinguishes between LAN traffic and SNA sessions, often called *legacy data traffic* by the product vendors. More sophisticated M-FRADs can give priorities to interactive client/server database access over bulk file transfers, or even SNA transactions to one mainframe over SNA transactions to another mainframe. The key is that the M-FRAD is aware not only of frame relay frames, but also the differing connection identifiers (the DLCIs) of the frame relay connections or virtual circuits.

Voice FRADs

The last type of specialized hardware that might be encountered in FRADs is voice capability. These voice FRADs, or just V-FRADs, typically have a *harmonica* style interface for hooking up 50-pair twisted-pair copper wire from an organization's PBX. These cables can carry up to 24 voice channels from the PBX into and across the frame relay network to a similar device on the other end of the PVCs used for voice. Ordinarily, these voice channels would be carried on tie-lines, which are nothing more than leased lines used for voice purposes between PBXs. Most often, the 24 voice channels would be represented by 64 kbps digital voice and the 24 channels would be carried by a 1.5 Mbps DS-1 circuit in the United States.

Naturally, if an entire DS-1 acting as a frame relay UNI were used to carry regular 64 kbps voice conversations, then when 24 people were on the telephone at one site, all data transfer would cease. This is what the V-FRAD is for. The V-FRAD will take the 64 kbps voice and further compress it to anywhere from 4 kbps to about 13 kbps, depending on the V-FRAD vendor and desired voice quality. So 24 voice compressed channels should only take up between 192 kbps and 312 kbps on the 1.5 Mbps UNI.

The voice compression is done by a special board in the V-FRAD known as a Digital Signal Processor (DSP). Although the term DSP might seem to imply that any digital signals at all could be subjected to this process, in practice only digital *voice* signals are processed by current DSPs. The compression must be done by hardware because of the delays that might otherwise be introduced by attempting to perform this task in software. For smaller installations using only a 64 kbps UNI, the DSP boards usually have individual modular jacks for handling only a few telephones instead of a whole T-1 interface.

Voice over frame relay is facing a real challenge from Voice over IP (VoIP) proponents and equipment. However, doing VoIP with adequate quality typically means that the organization must use a *managed* Internet Service Provider's service. This usually translates to the ISP providing a separate access line and backbone network in the form of routers connected by leased lines. In other words, the VoIP in this case has nothing to do with the Internet, other than the fact that the ISP also happens to provide Internet access. The attraction of doing voice over frame relay is that the voice is more intimately tied in with the basic service. That is, the voice over frame relay is delivered over the same network, from UNI to backbone, as the data service. This is not often true with VoIP today.

More will be said about voice over frame relay in Chapter 9.

The discussions in this section should be used for information purposes, not as a blueprint for one FRAD or another. As time goes on, all FRADs will develop support for multiservice priorities and support for nonpacket-based telephony. So the distinctions between FRAD, M-FRAD, and V-FRAD will blur over time. In some truly state-of-the-art packages, the differences between M-FRAD and V-FRAD have already begun to be merged into the same device, just with different boards for the different functions.

Figure 3.5 shows what passes for a state-of-the-art hardware FRAD today. Some of the features have yet to be discussed in detail, but this is the place to deal with the overall features that a frame relay customer should expect to find or at least be available for the CPE device at the customer end of the UNI. The chassis is a standard rack-mountable or standalone unit whose cost will vary widely based on features and functions supported. All include a main and backup power supply, and redundancy is always advisable for installations using voice over frame relay (otherwise voice communication is cut off with loss of building power). Typically, any network management capabilities are also built into this main system board, but there are exceptions. In fact, the

network management capabilities are the main distinction between software and hardware FRADs, as will become apparent. If the unit supports multiple queues for giving priority to one traffic stream or physical port over another, there is typically a separate board for that function, although some units combine this capability with the main system board.

The rest of the slots are configurable on a mix-and-match basis depending on number and type of premises connection, and number and type of UNI connections (some FRADs can support multiple UNIs). For the premises side of the network, the connectors almost universally include one or more 10Base-T LAN connectors, often support one or more token ring connectors, and might include more exotic LAN and device connector types such as Fibre Channel. For the WAN side of the network, the FRAD supports multiple UNI connector types, usually depending on the speed of the UNI itself. The most common is the V.35 connector for a 56-64 kbps UNI, but 1.5 Mbps and 45 Mbps UNI are also supported, of course. Usually there is only one UNI connector, but more are possible. There would also be a voice DSP board for compressed voice at 8 kbps or so, depending on the compression method used. The remaining slots (if any) would be used for expansion. And, of course, there is no requirement for the DSP board (for instance) to be present until there are plans for voice support, and so on for the other optional services.

More details on FRADs are available from the individual vendors, from various trade magazines who periodically review such devices, and from the Frame Relay Forum (FRF). The FRF is a vendor consortium interested in vendor interoperability of frame relay devices; it issues implementation agreements (IA) covering a wide variety of frame relay topics. More information on the frame relay forum will be found in the bibliography to this book.

Regardless of the future of FRADs, all FRADs share a common purpose. The FRAD exists to allow users to access the frame relay network. The frame relay network exists to give each user the quality of service (QoS) that he or she needs to allow applications to function as designed. Because a lot of time will be spent going over a frame relay network's techniques for delivering the proper QoS needed, this is a good place to say a few words about what QoS on a network is precisely.

Quality of Service and Networks

Having the proper quality of service (QoS) on a network is a lot like having nice weather on a vacation. Not only does the term "nice weather" mean different things to different people, it means different things depending on what the vacation is all about. Obviously, nice weather for skiing in the Rockies is not the same as nice weather for sunbathing in the Bahamas. So it is with networks. The QoS needed for bulk file transfers like remote server backups is not the same as the QoS needed for packetized voice. Also, people being what

they are, no one complains about too much nice weather or excellent QoS. But everyone notices when the weather or QoS fails to live up to their expectations. But the network, like the travel agent, is always held responsible if the weather or QoS disappoints a user expecting one thing and given another.

Analogies are nice tools for comparison, but they can only be pushed so far before they either become inadequate or just annoying. This section will say no more about vacations, especially since no one has ever confused setting up a network with taking a vacation.

There is no official definition of QoS. For the purposes of this book, QoS will be defined as *the ability of a network to deliver to each and every user application that specifies a series of QoS parameters the correct amount of network resources needed to deliver that QoS*. This definition sounds complicated, but it really is not. All it really means is, as an example, that if a user tells the network that this application needs a delay of 20 ms across the network, plus or minus 1 ms, the network will make sure that this happens. If not, then the user has a legitimate complaint and there might be a rebate on the monthly bill or some other penalty involved. Guaranteeing QoS is not easy on the part of the network. The network must not only look around (using whatever mechanism employed for this purpose) and see if this 20 ms delay is even possible to deliver, but also make sure that no other applications granted a QoS in the future are allowed to affect the QoS just given out. In other words, the network cannot suddenly stop 20 ms delay because a whole raft of other users are now demanding 20 ms delays also.

The delivery of QoS is so difficult to accomplish consistently on large public networks that the Internet as structured today cannot do it at all, and networks such as ATM, designed for QoS delivery from the ground up, can only deliver QoS under certain circumstances. This non-QoS support is the main reason that the Internet, and IP networks in general, are considered to be *unreliable*. It seems odd that a network like the Internet, characterized by dynamic rerouting around network node (router) failures while switched networks drop connections when switches fail, is considered unreliable while switched networks that drop connections are considered *reliable*. But this is only because the term "unreliable" when used in an Internet or IP context simply means that the network is *unreliable* when it comes to delivering user-desired QoS parameters such as stable delays (or guaranteed delivery!). The circuit-switched PSTN, although perhaps failure-prone, is much more *reliable* at delivering the QoS parameters that voice (especially) requires in terms of stable delays.

There is no real agreement as to exactly what parameters go into QoS. This sounds odd, but it is true. Most would agree that at least bandwidth, delay, delay variation (jitter), and error performance (in terms of cell/packet/frame loss) belong on the list of QoS parameters. Some add at least one or sometimes even two more. Reliability concerns, this time in the sense of network

availability, have become more acute with the recent widely publicized outages of portions of the Internet and public frame relay networks. In fact without reliability, the ability of a network to deliver any other QoS parameters becomes pretty much moot. In some routing protocols, reliability is a metric that can be maximized when routing decisions are made.

There is also a good argument for adding security to the list of QoS parameters. The venerable IP protocol has had a bit configuration available in the IP packet header telling routers to "maximize security" when routing the packet for more than 20 years. Router vendors have never implemented this option, but that does not mean it is unimportant. It could even be argued that given today's dependence on the Internet and other public networks for commerce and finance, if security is not a QoS parameter, it soon must be.

So a comprehensive list of QoS parameters will have not four, but six items. These are:

1. Bandwidth (number of bits per second the application needs, e.g., 8 kbps).
2. Delay (maximum amount of time it can take to reach destination, e.g., 20 ms).
3. Delay variation or *jitter* (amount of time the delay is allowed to vary, e.g., 1 ms).
4. Errors or information loss (percentage of cells/packets/frames the network can lose, e.g., 0%).
5. Reliability (annual percentage of time that the network must be available, e.g., five nines or 99.999%).
6. Security (degree of protection afforded to information on the network, e.g., double encryption).

The actual values of these parameters will differ from application to application and the ability of a given network architecture to support them will differ from network to network. For instance, delays are so variable on the Internet that it makes absolutely no sense for applications to specify delay variations limits.

What has all of this discussion of QoS have to do with frame relay? Primarily, to make the point that frame relay occupies a sort of *halfway* point between network architectures with no QoS delivery mechanisms at all like the Internet and other IP-based best-effort networks, and network architectures that were invented specifically to deliver precise QoS performance to all applications, such as ATM. This means that while there is no mechanism in frame relay for an application using a frame relay PVC to inform the frame relay network of its QoS needs, there are some basic bandwidth reservation mechanisms built into frame relay. Even a multiservice FRAD can only guarantee that some PVCs will receive priority queuing over other PVCs, not that the network delay will be lower than X at all times. This is a form of *relative QoS,* not the kind of *absolute QoS* that ATM can deliver.

But arguments that frame relay has no QoS guarantees and seem to put frame relay into the same category as the Internet or other IP networks are just wrong. This line of thought emphasizes the lack of a complete set of explicit QoS parameters that is present (but not always used) in ATM networks. It is true that the best that can be said for frame relay QoS is that the QoS is *probabalistic* and not *deterministic* as in ATM. So a frame relay network might be able to *probably* deliver frames in under 20 ms (for example) 99.4 percent of the time. While very good QoS performance, it is not ironclad. The 0.6 percent of a year that the delay is over 20 ms works out to 52.56 hours. If the QoS is not met for several 8-hour days when critical business activities are scheduled, this 99.4 percent might be of little consolation to the user. Yet the service provider has met the letter of the Service Level Agreement (SLA).

Frame relay service providers routinely use terms and conditions like "delay is less than 40 ms 99.5 percent of the time" and "PVC available will be 99.99 percent annually" (this works out to less than an hour of downtime a year). Some SLAs are quite explicit: "Delay is 10 ms plus 0.05 ms per 100 km of route miles from source to destination." All of these conditions require that some mechanism is put in place to verify the QoS level available to each and every application to verify compliance and detect violations. This is one of the reasons that frame relay, although a public network service, allows users to have ways of gathering more performance information about their portion of the network than ever before (how else could a customer ever determine route miles on the network?). The good news is that in most cases all of the network management mechanisms work extremely well. More details on SLAs, frame relay network management, and related topics will be discussed in Chapter 7.

It might be a good idea to close this section with a look at the service guarantees that a frame relay service provider would typically offer as opposed to the service guarantees that a typical Internet service provider would offer. These examples come from no specific source or service provider. However, they are certainly representative of the types of figures one would expect to see in a *virtual private network* proposal for a network based on frame relay as opposed to one based on the Internet or the IP protocol in general.

This comparison in made in Table 3.1. Note that service providers commonly distinguish between *network* availability and user or application availability. This is an attempt to say that just because an individual user or site has a bad month or year, overall the network is doing just fine. Also, the Internet column has the absolute *best* guarantee from any number of widely-known business Internet service providers.

So when it comes to QoS, frame relay is not the best network architecture available, but neither is it the worst. Frame relay QoS mostly provides only absolute bandwidth guarantees, but bandwidth is probably the most critical of the six QoS parameters when it comes to correct day-to-day user application operation.

Table 3.1 QoS Levels in Frame Relay and on the Internet or with IP

PARAMETER	FRAME RELAY	INTERNET, TYPICAL	INTERNET, BEST
Delay	60 ms one way	No guarantee	150 ms or less one way
Errors (Loss)	99.99%[1]	No guarantee	Individual case basis
Reliability			
Network	99.99%	99%	100%[2]
User/application	99.9%	No guarantee	100%[2]
Penalty?	Yes, detailed	No	Almost same as frame relay

[1] This applies only to traffic which conforms to the committed information rate (CIR).
[2] The ability of any service, let alone the Internet or IP, to be 100% reliable is remote. 99.7% or 99.5% are more often the best.

Private Routers and Public Switches

Frame relay is typically a public network service. The essence of a public network service is that the service provider owns and operates the network nodes. The basic network nodes in the public frame relay network are called *switches*. A lot of the reasoning behind the use of the term *switch* for a frame relay network node is historical. Traditionally, the network nodes for services provided by the telephone companies (itself an increasingly historical term in the days of deregulation) have been called switches. So there are central office switches, ISDN switches, and X.25 packet switches. Today, the term *switch* has come to mean *any network node whose primary method of operation involves setting up connections as paths for information to follow from source to destination*. So today there are Ethernet switches, Layer 3 switches, and the like.

Private networks also have plenty of network nodes. The essence of a private network is that the end user's organization owns and operates the network nodes. The basic network nodes in private networks today have a variety of names. In a small LAN, the network nodes are called *hubs*. When a private network is used to connect LANs, the network node used to connect these LANs was most often a *bridge* in the past, but it is the *router* today.

At first, these characterizations of public switch and private router seem obviously wrong. Is not the network node of the public Internet the router? And is not a Fast Ethernet (100 Mbps) hub called an Ethernet switch? The

answer is yes to both questions. But this does not mean that the origins of these terms are not correct, only that their current usage has little to do with their original context.

The Internet network node is called a router because a company named cisco decided that this is what the device should properly be called. Until then, Internet routers were called *gateways;* this term can still be seen in various Internet acronyms such as IGP (Interior Gateway Protocol) and BGP (Border Gateway Protocol) that apply exclusively to routers. One of the reasons for the change is that the OSI RM defined a *gateway* as a network connectivity device that operated at all seven layers of the OSI RM. But Internet gateways (routers) operated only at the bottom three layers of the OSI RM. So the change was made, and successfully, largely due to cisco's enhanced standing in the field it basically created single-handedly. (One of the reasons that bridges faded as LAN interconnection devices is that many routers could also function as bridges if so configured. Once called *brouters,* the bridging capabilities of all modern routers is a given today and so the term was mercifully dropped.)

The Ethernet *switch* or switching hub is called a switch because LAN equipment manufacturers were looking for a term to distinguish how these LAN network nodes operated internally from other types of LAN hubs. The term *gateway* did not apply and the term *router* was already taken. Ironically, the most descriptive term and accurate term for what a switched Ethernet hub does, a simple *bridge,* was avoided since by then everyone knew that a router was a more advanced network device than a bridge (and this was true). The only term left that had ever been applied to network nodes at all was switch. So the very private LAN hub that employed bridging between each individual LAN port became known, for better or worse, as a LAN *switch.* And no matter how much more accurate the term *single port bridging hub* might be, LAN *switch* it remains and will remain.

The term *Layer 3 switch* applied to what otherwise appears to be an ordinary router is simply a repetition of this naming crisis. Routers operate at Layer 3. But this device is radically different, so what do we call it? Well, Layer 3 switch is not taken, and it certainly points out the router relationship (Layer 3). In this instance, the term *router switch,* which is basically the same as *Layer 3 switch* was avoided as too confusing.

So frame relay network nodes are switches and users' LANs use routers at the ends of leased lines to connect their LANs. But a private router can be a software FRAD. And frame relay switches can be used to link customer's routers to public Internet routers. Given the converging terminology previously noted (router switch), does all of this mean there is no difference at all today between switches and routers? Not at all. And because the frame relay switch and customer router as FRAD have such a close relationship, being at either end of the UNI, this relationship is worth exploring in a little more depth.

What Is a Router and What Is a Switch?

As has already been established, the network nodes in a frame relay network are called switches. On the Internet, the network nodes are called routers. More accurately, these are IP routers, since IP is the OSI RM Layer 3 protocol used in these routers. Why should any of this matter? The answer to this question is of vital importance for organizations building networks for their users and for the service providers who build the infrastructures that link the users together. If frame relay and other technologies such as ATM are to survive and prosper in the world of the Internet, the position of public switches in relation to IP routers must be considered.

Switches and routers can be compared in a number of ways. It is important to realize that even though this section emphasizes the differences between switches and routers, both are still network nodes that can be used in a wide variety of networks and under a wide range of circumstances.

Switches usually (there are exceptions) have the following characteristics. They are hardware-based; processing happens very quickly at the chipset level with a minimum of added overhead processing. Switches were created by the telcos for use on a public WAN and standards are governed by the ITU. The tables that are used for routing traffic through the switch are set up by a *signaling protocol* when the connection between the users is initially made. So switches are typically connection-oriented and no data transfer takes place until this connection between users is set up. Connections might be of the permanent virtual circuit (PVC) type or switched virtual circuit (SVC) type (*on-demand connections* is a more accurate, but seldom used, term). Both PVCs and SVCs are connections in either case, no matter how they are established.

All data units are distributed from input ports to their proper output ports by a simple, quick lookup in the *routing* table of a connection identifier, which makes the hardware implementation so attractive. This simplicity is a result of the connection-oriented approach of the switch environment. The connection identifiers often have what is called *local significance only,* which means they can be used over and over on the network as a whole and thus must be changed as the data units flow from node to node across the network. Examples of networks that use switches with these characteristics as network nodes include X.25, frame relay (naturally), ATM, ISDN, and many other mostly public network services.

The behavior of routers can be contrasted with switches almost point by point. Routers usually have the following characteristics, but as with switches, with some exceptions. Routers are mostly software-based (but this is changing) and processing happens more slowly at the CPU level with some added overhead processing. Routers were created for use on the public Internet (and called *gateways* until cisco popularized the name *router*), and router stan-

dards are governed by Internet organizations. The *switching* tables (the contrasting terms are used intentionally here) that are used for routing traffic through the router from input port to output port are created by a routing protocol that periodically contacts neighboring network nodes (other routers) for the purpose of exchanging this routing information. Routers are typically connectionless and data transfer between users can take place at any time without establishing connections between the routers. Note that there may be permanent or switched (on-demand) connections that exist end-to-end between users or applications, but there are no logical connections at all between the routers themselves, just physical links.

Because of the connectionless approach used in the router environment, the data units are distributed from input ports to their proper output ports by a set of rules which are applied to a global network address that must be present in each and every data unit. The fact that this global routing information is present in each data unit is a major reason behind the flexibility of routing and makes the software implementation so attractive. The network addresses handed out in a router environment have global significance; so they must be carefully allocated to users on the network. There can be no overlap among network addresses in a router network, a fact which alone adds administrational complexity to the network. Examples of networks that use routers with these characteristics as network nodes include IP, IPX, and a few others. Note that these network protocols started out as proprietary or closed protocols, whereas most protocols based on switches as nodes started out expressly as public and open protocols.

Today, the trend is toward convergence between switches and routers as network nodes. That is, routers have begun to take on the characteristics of a kind of *connectionless switch* routing data units with global addresses while switches have begun to take on the characteristics of *connection-oriented routers* switching data units with local addresses (connection identifiers).

As previously mentioned, some former LAN connectivity devices displaying the characteristics of both bridges and routers were called *brouters*. The term has thankfully disappeared, but perhaps the same approach could be taken with regard to network nodes that combine the characteristics of a switch and a router. This *swouter* device would do a lot of the data unit processing in hardware at the chipset level but would have many tables to look things up in as well. This device could handle connections or *flows* of IP packets. The point is clear: If this device is neither a traditional switch nor traditional router, then what exactly is it?

Switches and routers have already merged in function to the point where more and more equipment manufacturers are not calling their new products a switch or a router at all. The device might be a *packet processor* or *nodal processor*, but not merely a switch or router.

Two examples are instructive. Start with a normal, premises-based IP router. Add the hardware module that cisco calls its *route switch processor* card. Then add some software to handle IP version 6 *flows,* which are basically a type of on-demand connection between routers. If a frame relay UNI is added, the result is more than a FRAD, but something less than a full-blown public frame relay switch.

Or start with a frame relay switch in the public frame relay network. Everyone has a router, but perhaps no potential customers want to buy a FRAD or they are reluctant to change their router configuration. No problem. Just add some software to the frame relay switch to handle IP and other traditionally connectionless protocols. This frame relay switch-based software needs the IP routing tables to perform its task, of course and the IP routing protocols would need to be added to maintain the routing tables properly. Is this frame relay switch now an IP router? Or is it rather (and this seems to be what it really is) a new type of central office FRAD or CO FRAD?

It should also be noted that in a router network, each data unit is usually routed independently. But in a switch network, only the call setup message of the signaling protocol is routed independently. In a switch, all subsequent traffic follows the same path. A router can also do this, using a concept called *flows*, as previously noted.

Routing and Switching on a Frame Relay Network

All the pieces are now in place to understand both how a frame relay network operates and how this operation is an improvement on how an X.25 packet-switching network operates. X.25 networks operate by taking source X.25 packets from an end-user device and placing them inside of frames based on the Link Access Procedure-Balanced (LAPB) standard. The term *balanced* means that the same type of messages can flow from either end of the link, making them peers from the networking perspective. The LAPB frames are sent as a stream of bits from source device to network X.25 packet switch.

At the public X.25 switch, each arriving frame is checked for errors and, if none are detected, an acknowledgment is sent back periodically to the source saying, in effect, “the last x frames were good.” The sender must wait after sending “x” frames until this acknowledgment appears. There is always the possibility of a Negative Acknowledgment (NACK) appearing as well. This usually prompts the sender to resend at least one and probably more frames due to the detected error. All this occurs at Layer 2 of the OSI RM. Inside the frames are the packets or more likely a piece of a packet. There is a measure of *flow control* done at this level as well. Flow control simply means that the user premises device can never send frames and packets into the network faster than the network can handle them. One of the simplest ways to perform

flow control at this level is to merely delay an acknowledgment so the sender cannot send any more frames or packets.

The public X.25 switch now assembles the entire packet and examines the connection identifier at Layer 3. The connection identifier is looked up in a table (the switching or routing table) and the proper output port determined. There might be packet-level acknowledgments involved as well. Then the public switch repackages the X.25 packet in an appropriate frame for sending to another public X.25 switch. These frame-level procedures are not quite the same as LAPB, and are vendor-specific, but must perform the same error checking and acknowledgments sequence as on the user access link. There might be an arbitrary number of public X.25 switches to traverse until the packet arrives at the switch that has the X.25 access link to the destination.

At the destination side of the network, this entire packet-in-frame error detection, acknowledgment, and flow control procedure is repeated. So all three layers of the OSI RM are involved at each hop between nodes along the way. This link-by-link, or hop-by-hop, error and flow control is characteristic of all older network protocol such as X.25 and is shown in Figure 3.6.

There is nothing at all wrong with doing things the hop-by-hop way on a network. In fact, it saves the end devices the tasks associated with error and flow control, since the network is doing all of this on behalf of the users. With X.25

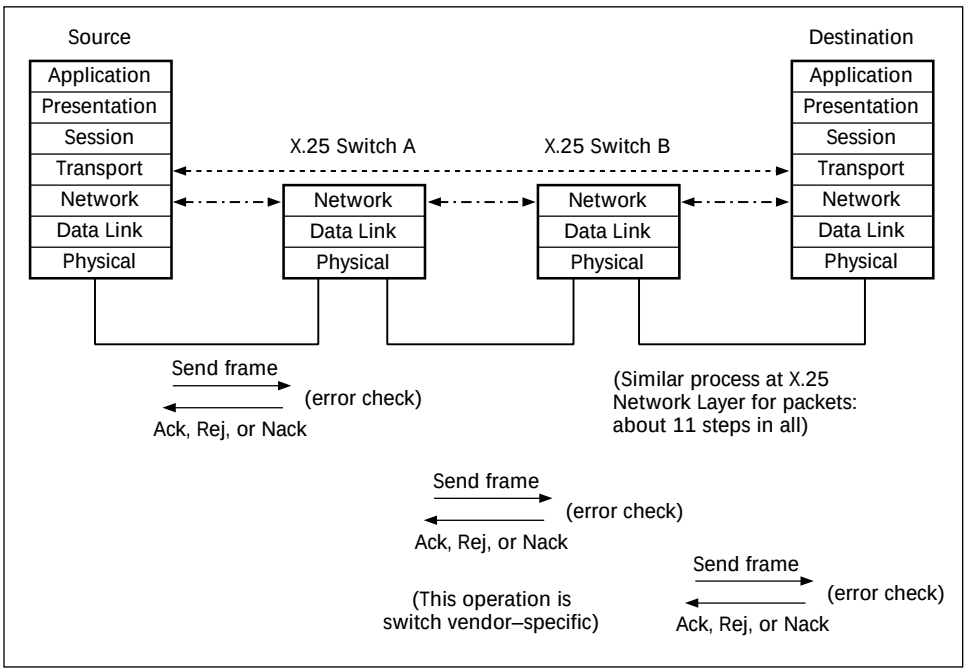

Figure 3.6 Information flow in X.25.

it was even possible to have a source sending at 9,600 bps (which shows how long X.25 has been around) and a destination receiving at 4,800 bps. The network would buffer and store the excess bits until the receiver was ready for them. Try doing that with a leased line!

However, today's source and destination PCs and routers are much more capable than they were just a few years ago. The complete Layer 2 and Layer 3 processing required in X.25 now slows the network down more than it decreases the burden on the end systems. All in all, there are some 10 decisions that an X.25 switch must make to process a packet through an X.25 network node. About six of the decisions are at Layer 2 and four of them are at Layer 3. The details are unimportant here. What is important is that most of these decisions have to do with the error and flow control that must be done hop-by-hop throughout the X.25 network.

The philosophy in a frame relay network is radically different. Instead of the network performing error and flow control hop-by-hop, the frame relay network makes these procedures the responsibility of the end-user device. (Some frame relay texts say that the end user is responsible, conjuring up the image of an office worker feverishly working to resend information through the frame relay network.) It is the end-user device, such as the host attached to the router, that performs the error and flow control tasks *end-to-end* across the frame relay network.

In a frame relay environment, the packet scenario is as follows. Frame relay networks operate by taking source frame relay frames from a site CPE device (e.g., a router or FRAD) and placing them inside of frames based on the Link Access Procedure-Frame Relay (LAPF) standard. There are several levels or types of service a frame relay network can offer; the most basic is based on *LAPF core*. This is the type of frame relay network described here. The LAPF frames are sent as a stream of bits from source device to a network frame relay switch.

At the public frame relay switch, each arriving frame is checked for only two things. First, if the frame contains any errors at all, it is just discarded. No notification of this frame discard is sent to the source. Second, the arriving frame is checked to see if the Data Link Connection Identifier (DLCI) in the frame header has a routing or switching table entry. If the DLCI is not in the table, then the frame is discarded. If no errors are detected and the DLCI has a table entry, the frame is switched to the proper output port. No acknowledgment is sent back periodically to the source. So there is no flow control or error control done *within* the network. If missing frames are to be resent, it is the task of the end systems to decide if any frames are missing and what to do about the missing frames (voice frames can hardly be resent!). The same logic applies to flow control: If the destination system wants the source system to slow down, then it is the responsibility of the destination to inform the source of the need to slow down the sending process. The frame relay network will

convey this information inside the frames to the source, but the frame relay network is never aware of the contents of the frames that the network transports on behalf of the users.

There is no need for the public frame relay switch *ever* to look inside a frame and assemble or process the entire packet. The connection identifier is all that is needed to allow the frame relay switch to determine the proper output port from the switching or routing table lookup. There are no packet-level acknowledgments or flow control done in the network at all. The public frame relay switch does no re-packaging of packets; it only *relays frames* from input port to output port. The frame-level procedures used between the frame relay switches are not quite the same as LAPF core, but are vendor-specific, just as in X.25 and most other public networks. There still might be an arbitrary number of public frame relay switches to traverse until the frame arrives at the destination.

At the destination side of the network, the frame content is subjected to error detection, acknowledgment, and flow-control procedures if necessary based on the application. But this is an *end-to-end* function of the user devices, not a hop-by-hop function of the network itself. So only the bottom two layers of the OSI RM are involved at each hop between nodes along the way. This end-to-end error and flow control is characteristic of all newer *fast packet* protocols such as frame relay. The frame relay information flow is shown in Figure 3.7.

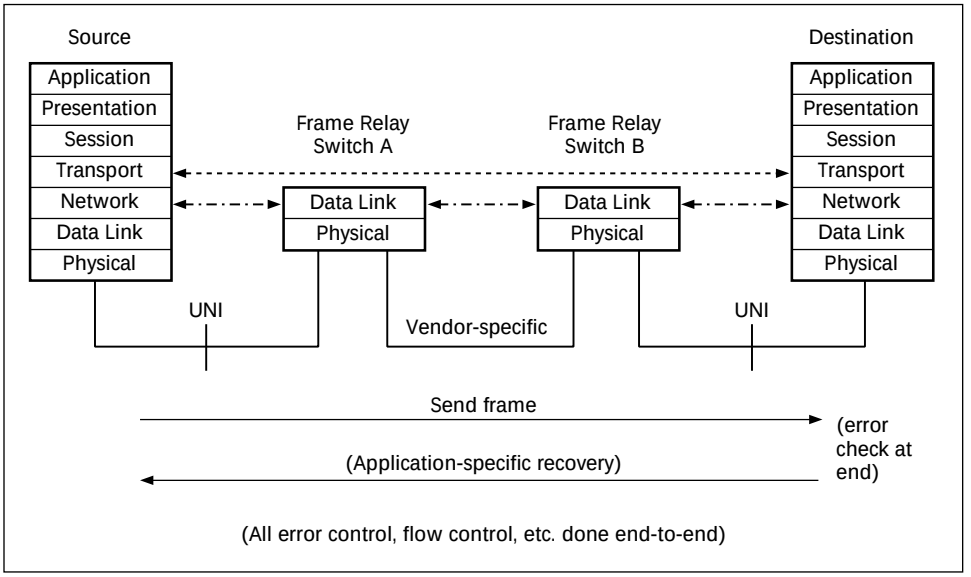

Figure 3.7 Information flow in frame relay.

Note that the end-to-end layer in a frame relay network is the Network Layer and not the Transport Layer. This means that IP packets, or any other OSI RM Layer 3 data unit, now become the end-to-end data transfer unit through the frame relay network. So any mechanisms that the TCP/IP protocol has in place to handle error control and flow control work just as before (if the network was previously run on leased lines or the like) and the frame relay network is totally transparent to the IP routers. This simple transparency is both a benefit and a liability to the network and router alike, and will be explored more fully in the next few chapters.

The Frame Relay Protocol Stack

Only a few related topics remain to give a good overall description of how a frame relay network actually works. It has already been pointed out that the vast majority of public frame relay networks (and even private ones) offer only Permanent Virtual Circuits (PVCs) for connectivity. The few frame relay service providers that do offer Switched Virtual Circuits (SVCs) are few and far between, and usually have many restrictions on the number of SVCs that can be established, where the endpoints are located, and so on. Of course, PVCs have no call setup delays while SVC signaling messages are processed by the network to determine routes and network resources, establish switch table entries, and engage billing procedures. All of these issues will be discussed more fully later.

All that remains here is to show that SVC support depends on the exact frame relay protocol stack that a frame relay service provider supports. Mention has already been made of LAPF core, the basic frame protocol run on any frame relay UNI at all. The fact is that LAPF core simply transfers frame relay frames around a PVC-defined network. That is, no SVCs are possible in a frame relay network employing only LAPF core.

In is often said, and not incorrectly, that frame relay is defined at the bottom two layers of the OSI RM. This is not inaccurate if the *data transfer* aspect of frame relay is being discussed. But there is more to networking than data transfer, much more in fact. Networks must be managed with some form of network management techniques. The techniques could be added on, but network management is more efficient and consistent if the techniques are part of the network specification itself. A network must be controlled with some form of signaling protocol so that users and network can inform each other of their intentions in terms of connectivity. This is the task of the signaling protocol.

In any case, there is no room at Layer 2 of the OSI RM for these management and control functions. Yet these functions must be performed in the

frame relay network nodes, the switches themselves. With regard to the frame relay protocol stack, these functions can be considered to be at Layer 3, the Network Layer, although most texts are fond of insisting that there is no Layer 3 in frame relay switches at all.

Here is how the frame relay protocol stack actually looks. X.25 is a fairly faithful representation of what the OSI RM should do at the Physical, Data Link, and Network layers. In X.25, the layers can be represented by V.35 or X.21 at the Physical Layer, the X.25 LAPB at the Data Link Layer, and the X.25 Packet Layer Protocol (PLP) at the Network Layer. Frame relay can perform all data transfer tasks with a subset of the full OSI RM Data Link Layer. Network management is done in frame relay as a small subset of OSI RM Layer 3 and only on the UNI. This basically means that frame relay network management on the UNI consists of a small set of messages inside special frames sent back and forth on the UNI. LAPF on its own only supports manually configured PVCs.

If SVCs are to be supported in frame relay, they can take one of two forms. The Integrated Services Digital Network (ISDN) signaling protocol specification on which frame relay signaling is based is called Q.931. Frame relay adapts and extends the Q.931 ISDN signaling protocol as Q.933. If the frame relay SVCs are established with signaling messages that form a subset of the full Q.931 ISDN call control message set, technically a Q.933 subset, this is known as *non-ISDN SVC support*. With non-ISDN SVC support, there is no relationship between a service provider's ISDN signaling network (and billing system) and its frame relay SVC offering. But at least there are frame relay SVCs. However, a service provider can make its frame relay network a part of its ISDN, with frame relay playing the same role as X.25 as a *packet-bearer service*. This requires the full implementation of Q.933, however. With ISDN-

Network	X.25 PLP	Management	Q.933 subset & Management	Full Q.933 & Management
Data Link	LAPB	LAPF	LAPF	LAPF
Physical	X.21, V.35 (56/64 kbps)	V.35 (DS-0, DS-1, etc.)	V.35 (DS-0, DS-1, etc.)	V.35 (DS-0, DS-1, etc.)
OSI-RM	X.25	PVC only (with link management)	Non-ISDN SVCs (with link management)	ISDN SVCs (with link management)

Figure 3.8 The OSI RM, X.25, and frame relay.

compliant SVC support, there is a close relationship between a service provider's ISDN signaling network (and billing system) and its frame relay SVC offering. In this case, the user can use the service provider's ISDN to establish frame relay SVCs.

The relationships between all of these frame relay protocol stack permutations are shown in Figure 3.8.