CHAPTER

7

Not to Be Forgotten

If you would not be forgotten as soon as you are dead and rotten, either write things worth reading or do things worth the writing.

— Benjamin Franklin

We are now at the end of the ''Networking Nuts and Bolts'' part of this book. So far we have discussed most of the predominate standards that are implemented in the majority of networks. We have discussed the popular LAN and WAN standards that you will most likely be involved with should you continue in your quest of network knowledge. What you have seen in this section of the book is only a portion of the technologies that are available and/or implemented in many networks.

This chapter is going to provide an overview of some of the other standards and processes that are available and, for the most part, in use (if only in a small percentage of networks). The way we see it, it just wouldn't be a good networking book if these weren't at least mentioned.[1] Some of the technologies in the following pages are of a dying breed, whereas others are just starting to grow. Whatever their status, these are standards that have been replaced by other standards, enhanced by revisions to the original standard, developed to support proprietary hardware and/or software products, or developed to support a new technology.

When a standard is placed on the road to becoming obsolete,[2] it is normally due to technology advancements that the standard cannot support. This does not mean you cannot use the standard, but it does mean there will be no further advancements to the standard and, for the most part, what you see is what

[1] Although there are many good networking books out there that deal with even a single protocol.
[2] The process of retiring a standard is known as placing it into an ''end-of-life'' status.

you get (WYSIWYG).[3] Some of the standards we will discuss are proprietary but are often implemented as the standard of choice, and some are newer technologies that are just experiencing ''startup growth'' and will probably prove themselves to be a major part of networks in the next decade.

At the end of the chapter, we have provided an introduction to the structure of a datagram — what it is, how it works, and why it is important. This is to ensure that we keep that network knowledge flowing.

7.1 Can't Get Enough of Those LAN Technologies

In the last chapter, we discussed Ethernet, which is the most popular of LAN protocols in use today. Because of the advancements and cost savings offered by Ethernet, many other protocols have been retired (or are not as commonly used as Ethernet). In this section, we discuss a few LAN protocols that were once on the cutting edge, and may still be out there serving in some capacity.

RANDOM BONUS DEFINITION

100BASE-T — The term used to describe baseband Ethernet transmission of 100 Mbps.

7.1.1 Attached Resource Computer Network

In Chapter 1, we defined a LAN as a data network that covers a small geographical area. This normally ranges from an area with just a few PCs to an area about the size of an office building or a group of buildings. Attached Resource Computer Network (ARCnet) is a protocol that was once very popular in LANs, and has even found a purpose in today's Ethernet world. ARCnet is now used as an embedded standard to serve networks that control automation services, transportation, robotics, gaming, and other similar network types.

Developed by the Datapoint Corporation in the late 1970s, ARCnet was designed to use token-passing bus technology over coaxial cabling. The physical topology of ARCnet is a star/bus topology (see Figure 7-1). ARCnet touted speeds of up to 2.5 Mbps[4] and distances of up to four miles. ARCnet is considered the first truly commercially available LAN. Due to the low cost of the infrastructure and the simplicity in implementation and maintenance, ARCnet was very popular when it first arrived.

[3]Pronounced ''wizzy-wig.''

[4]A later version of ARCnet was released in the early 1990s and was called ARCnet plus. It could operate at speeds of up to 20 Mbps. By the time ARCnet plus had come out, however, Ethernet was quickly becoming the standard of choice.

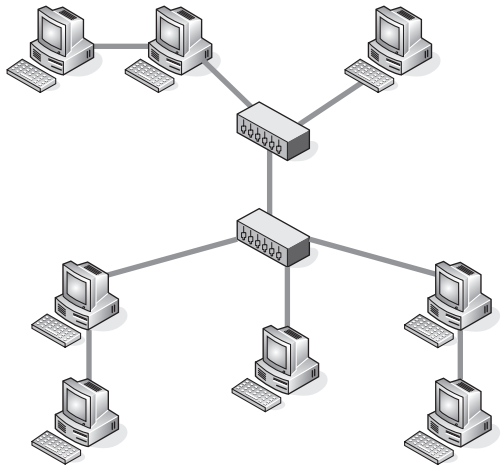


Figure 7-1 An example of an ARCnet topology

ARCnet doesn't have all the bells and whistles that are offered in networks today. It is a very simple technology that is easy to implement and run. A big drawback with ARCnet is that when an interface is brought into the network, the address of the interface has to be set by whoever is installing it. Most of the time, the address is set by jumpers or switches on the resource interface module (RIM)[5] itself.

ARCnet was designed to give Datapoint nodes the capability to share resources over the token bus, thus increasing the overall power of the attached nodes. Datapoint had originally intended to keep what became known as ARCnet fully proprietary because if the public bought their gear, they could tout resource sharing as a selling point.

Datapoint had some problems with the design of the RIM chip, so they eventually contracted with Standard Microsystems Corporation (SMSC). SMSC successfully built the chip specifically for Datapoint, and in the final negotiations got the approval to sell a version of the chip to other vendors — and ARCnet was born.

7.1.2 StarLAN

StarLAN technology is, for the most part, the predecessor to what we all know as Ethernet. Often referred to as *1BASE5* and developed in the early 1980s by AT&T, StarLAN provided a way

POP QUIZ

What was the name of the company that developed ARCnet?

[5]The RIM is basically the ARCnet-supported NIC card.

for nodes to communicate with one another over a telephone line. StarLAN operated at 1 Mbps and eventually supported speeds of 10 Mbps.[6] 1BASE5 actually came out after coaxial cabling came out supporting 10 Mbps. This is part of the reason that StarLAN never really got deployed in most LANs. Once 10BASE-T came out, the only time StarLAN was used was when someone needed a low cost infrastructure and speed was not a concern. Figure 7-2 shows an example of the StarLAN topology.

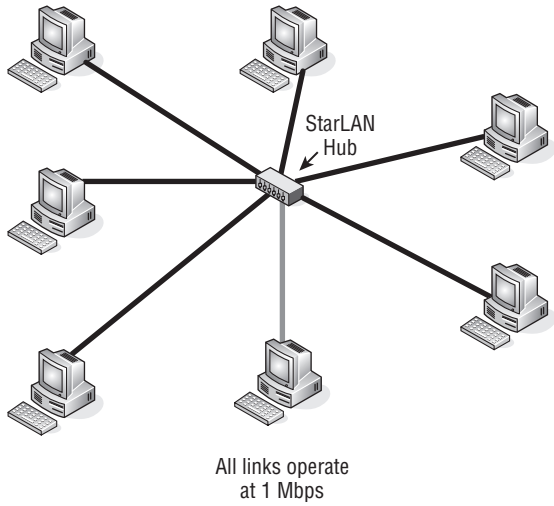


Figure 7-2 The StarLAN topology

StarLAN networks used UTP as a transmission medium and typically connected nodes to one another through at least one hub. StarLAN was able to also connect to multiple nodes without a hub by daisy-chaining them one by one upon the shared medium. The maximum number of nodes in a daisy-chain configuration was 10. Figure 7-3 shows an example of daisy-chaining.

7.1.3 Token Ring

Token Ring network technology was developed by IBM in the late 1970s. IBM submitted the proposed standard to the IEEE LAN standards committee, which adopted the proposal and used the standard as the basis for the IEEE 802.5 standard. Token Ring topologies are a star physical topology and a ring logical topology, as shown in Figure 7-4.

[6]By this time, however, 10BASE-T was out, which rendered this advancement moot.

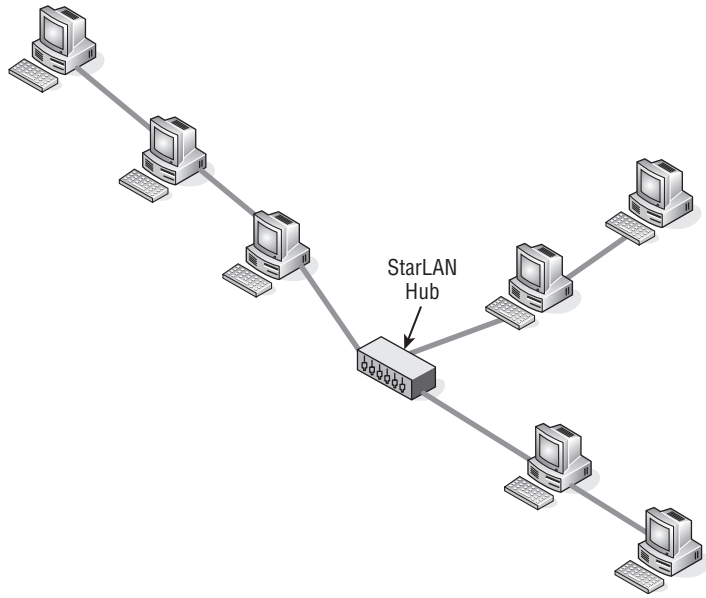


Figure 7-3 Including a daisy chain in a StarLAN configuration

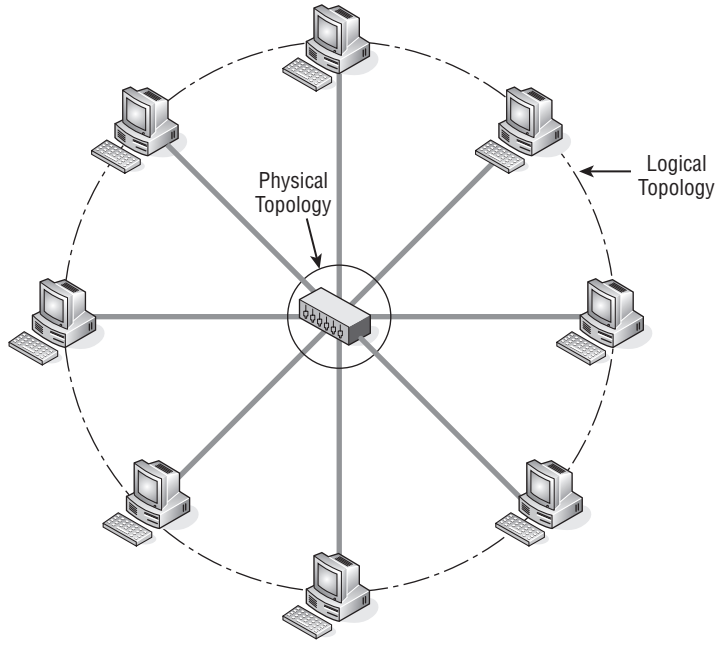


Figure 7-4 A Token Ring topology

Token Ring networks pass a signal, known as a *token*, from one node to the next. The node that you receive the token from is the *upstream neighbor*. The node that you pass the token to is the *downstream neighbor*. Each node receives the token, takes action, and then passes the token to the downstream neighbor (see Figure 7-5).

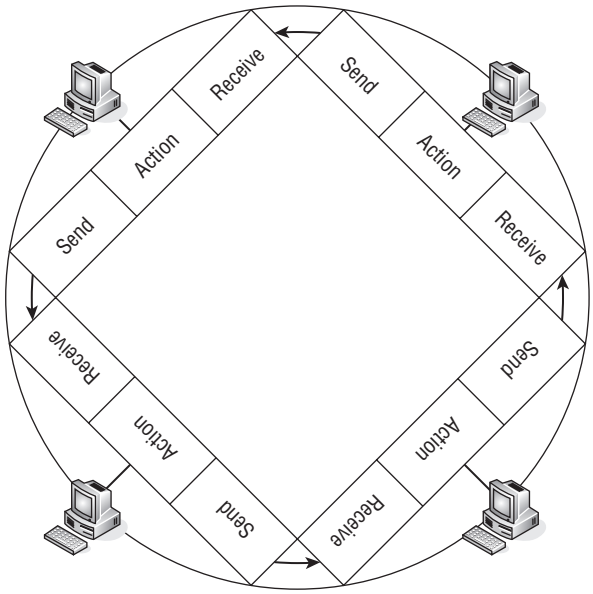


Figure 7-5 Token Ring operations

POP QUIZ

What technology is also known as 1BASE5?

The actions that are taken are determined by whether the node has control of the token. If a node controls the token, it transmits the token onto the ring to the downstream neighbor, which receives the token and then passes it on the ring to its downstream neighbor. The data is captured by each node, and once the token has made it back to the originating node, that node will remove it from the ring, thus freeing the ring up for the next token to be passed.

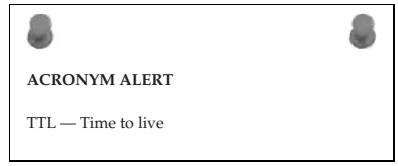
The original Token Ring supported speeds of 4 Mbps and later came to support 16 Mbps. It didn't take long for networks to upgrade to support the higher speed, especially as the demands on the LAN grew. There is an 802.5 approved standard for Token Ring, allowing up to 100 Mbps speeds, but this never really became popular.[7]

[7] Anyone care to guess why?

7.1.3.1 Token Ring's Modus Operandi

In a Token Ring environment, only one node can transmit data from itself at a time. The originating node is given the token in order to pass it on to the network. The node sets the Token bit from a 0 to a 1, which transforms the Token into a datagram known as a *frame*. The data is passed from node to node around the ring. Each node inspects the frame and forwards it to the downstream neighbor.

Once a node inspects the data frame and recognizes its own address as a destination address, the node retains a copy of the data and sends the data on to the next node in line. The data continues around the ring, inspected by all nodes, and then returns to the originating node, which retrieves the frame from the token and sends a new token[8] on to the next node. Once the token arrives at a node that wants to send data, the process begins again.



7.1.3.2 Token Ring Media

Token Ring originally operated on STP cabling but converted to UTP cabling in the 1990s. This was greatly appreciated by the networking community, as it offered a cheaper and less bulky medium.

MMF[9] cabling was supported officially in 1998 when an approved amendment was written into IEEE 802.5, although in actuality a lot of networks were using it already. Token Ring 100 Mbps operation is conducted on the exact twisted pair specification that is used for 100 Mbps Ethernet.

7.1.3.3 The Format of the Token Ring Frame

Token Ring uses one of three frame types. *Token frames* have the token bit set to 0 and have no data. *Token data frames*[10] have the data payload contained within the frame (the token bit is set to 1). The abort frame carries no data and is used to stop its own transmission of data, or used to clear up data that is on the line.

The fields contained within the token frame are fairly simple to understand, as shown in Figure 7-6.

[8]Sending a ''new'' token simply means that the token bit is set back to 0, indicating an available token.
[9]Quick refresher: In Section 3.2.1.3 we discussed the two types of optical fiber, multi-mode fiber (MMF) and single-mode fiber (SMF).
[10]Also known as a token command frame.

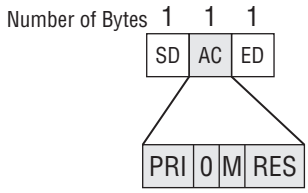


Figure 7-6 An empty Token frame

- **SD** (start of frame delimiter) — This field lets the receiving node know when the frame begins.[11]
- **AC** (access control) — There are four subfields in the access control field, all used to transmit information to the access control process within Token Ring.
  - **PRI** (priority bits) — The priority bits show the priority level of the frame.
  - **0** (token bit) — This bit differentiates the frame type. In Figure 7-6, the token bit is set to 0, identifying it as a token frame.
  - **M** (monitor bit) — The monitor bit is used by a node that is known as an active monitor node. This bit is used to detect various errors.
  - **RES** (reservation bits) — The reservation bits are used by a node to announce that it has data to send and needs to use the token as soon as it is available. Reservations are based on the priority level that has been set.
- **ED** (end of frame delimiter) — This field lets the receiving node know when the frame ends.[12]

POP QUIZ

What is the signal called that is passed in Token Ring from one node to the next?

The token data frame format is pretty much an extension of the token frame format. The first two fields are identical, but the third field is moved to the end of the frame (where it belongs). Several fields are in between that contain the data and the information that a node will need to send and receive frames on the Token Ring.

[11] There has to be something identifying the beginning of the frame.
[12] When you have to be clued in when the frame starts, there has to be some way to let you know that the frame is complete.

Figure 7-7 shows the fields contained within the token data frame.

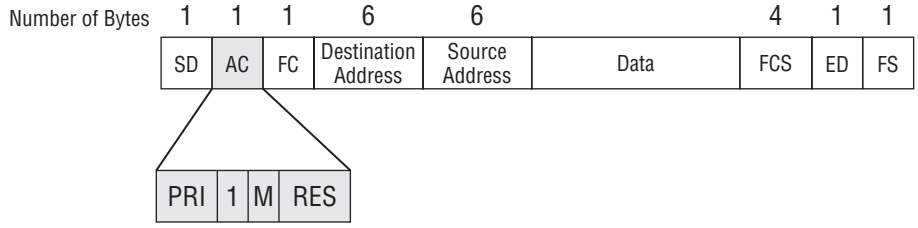


Figure 7-7 Token frame with data attached

- **SD** (start of frame delimiter) — This field lets the receiving node know when the frame begins.
- **AC** (access control) — There are four subfields in the access control field, all used to transmit information to the access control process within Token Ring.
  - **PRI** (priority bits) — The priority bits show the priority level of the frame.
  - **1** (token bit) — This bit differentiates the frame type. In Figure 7-7, the token bit is set to 1, identifying it as a token data frame.
  - **M** (monitor bit) — The monitor bit is used by a node that is known as an active monitor node. This bit is used to detect various errors.
  - **RES** (reservation bits) — The reservation bits are used by a node to announce that it has data to send and needs to use the Token as soon as it is available. Reservations are based on the priority level that has been set.
- **FC** (frame control) — The frame control field is used to separate network management data frames from user data frames.
- **Destination Address** — This field contains the 6-byte network address of the node the frame is destined for.
- **Source Address** — This field contains the 6-byte network address of the node the frame originated from.
- **Data** — This field contains the data from the upper layer protocol that is being transmitted. There is a certain limit on the amount of data that can be included in the frame. At 4 Mbps, the limit is 4,528 bytes. At 16 Mbps, the limit is 18,173 bytes. At 100 Mbps, the limit is 18,173 bytes.
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field.

- **ED** (end of frame delimiter) — This field lets the receiving node know when the frame ends.
- **FS** (frame status) — This field is used by the originating node to detect whether there were any errors during transmission. This includes: if the destination node copied the data; if there were any errors encountered; and even if the destination node recognized itself as the destination node.

RANDOM BONUS DEFINITION

trunk — A name defining a bundle of links, also known as *aggregate links*.

7.1.4 Fiber Distributed Data Interface

The *Fiber Distributed Data Interface (FDDI)* is a LAN[13] and/or MAN technology. FDDI[14] was the first such technology that could operate at 100 Mbps. FDDI is an ISO standard and is fully compatible with the IEEE 802 standards.

Although FDDI could function as a LAN technology, it is cheaper and easier to use 100 Mbps Ethernet. When FDDI was developed, it was intended to provide higher speeds in LANs than the quickest rate that was available at the time: 16 Mbps Token Ring or Ethernet. FDDI is sometimes used to connect server farms and multiprocessors to the network. Most often you will find FDDI deployed within the backbone of the network, providing quick connectivity between other networks.

7.1.4.1 FDDI Does What FDDI Does

FDDI was designed to operate over shared fiber media. The fiber connected nodes in a ring similar to the IEEE 802.5 Token Ring standard configuration. The difference is that FDDI uses a dual-ring topology over a shared fiber medium.[15] Data traffic on a FDDI ring flows in a counter-rotating manner. This means that data on one of the rings goes in one direction while the other ring carries traffic in the opposite direction. The ring that actively carries data is the primary ring

POP QUIZ

What information is contained in the Destination Address field in a Token Ring frame?

[13]Most networks use FDDI at the MAN levels.
[14]Pronounced ''fiddy.''
[15]There is a newer standard for FDDI that allows the use of twisted pair cabling instead of fiber. This is called the *Copper Distributed Data Interface (CDDI)*, discussed in Section 7.1.4.1.2.

and the other is the secondary ring, which remains in an inactive status until needed. Figure 7-8 shows an example of the FDDI topology.

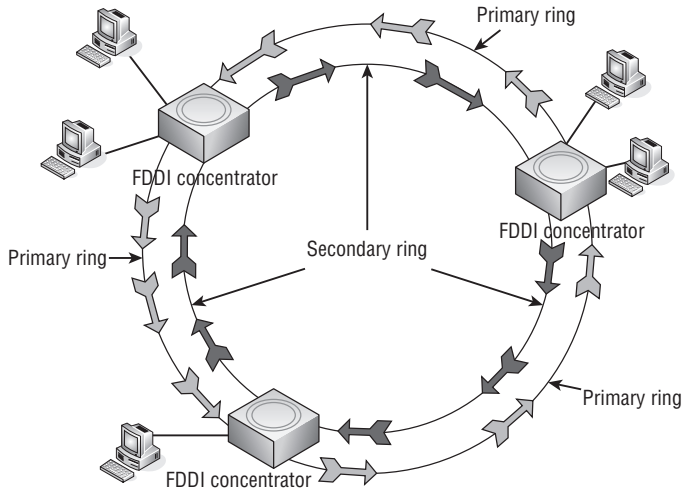


Figure 7-8 FDDI topology

Notice that unlike Token Ring, which connects to a central MAU, there are concentrators[16] that connect nodes to the FDDI topology. We will discuss the different concentrator types in Section 7.1.4.2. Other nodes that can be used within a FDDI ring are servers, routers, switches, and so on. As long as the node is able to support FDDI, it can be used for its intended purpose on the FDDI ring.

The FDDI protocol supports optical fiber (FDDI) as well as copper cables (CDDI)[17] as a shared medium. The operations provide the FDDI functions, with the difference being the medium type used. Both have advantages and disadvantages, which we will discuss in the next two sections.

7.1.4.1.1 Fiber Distributed Data Interface

FDDI is the FDDI protocol over fiber optic cabling. Both MMF and SMF optical fiber medium types are supported in a FDDI environment.

[16]Refer to Section 3.3.3.1 if you do not remember what purpose the concentrator serves in a network.

[17]The official name is *twisted pair physical medium dependent (TP-PMD)*; however, CDDI seems to be gaining in popularity. CDDI is a Cisco term, while TP-PMD is the ISO term. It seemed to us that it is easier to refer to this as CDDI for the purposes of this book, but you may need to know both acronyms when working in a professional environment (you don't want to get caught saying, ''Huh?'' when someone asks you if your TP-PMD is running). As has occurred many times in the history of networking, terms come and go. What is important is that you understand what they are referring to.

There are advantages in using optical fiber as the primary transmission medium:

- Performance
  - Greater distances
  - Faster transmission speed
- Reliability
- Data security

Each advantage is due to the actual medium itself. Optical fiber uses light instead of electricity to carry data. This prevents the leaking of electrical signals, thus improving performance and the reliability of the transmission of data. This also increases security as there is no way to tap into the fiber optic cable. This ensures that, for the most part, only the individuals that are intended to see the data will see the data.

7.1.4.1.2 Copper Distributed Data Interface

Copper Distributed Data Interface (CDDI) is the FDDI protocol over twisted pair media instead of fiber. CDDI is officially known as *twisted pair physical medium dependent (TP-PMD*) and is also known as *twisted pair distributed data interface (TP-DDI*). CD*TP-PMD*DI uses both STP and UTP cable types.

The main advantage with copper is that it is cheaper and easier to install and maintain than fiber. Because copper cannot transmit the distances that fiber can, it is often used to connect nodes to the concentrator in the FDDI environment. Figure 7-9 shows an example of this.

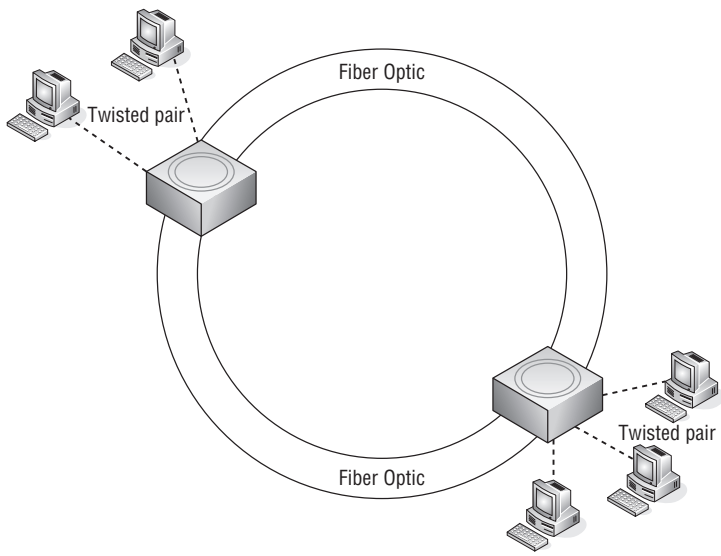


Figure 7-9 FDDI and CDDI together

7.1.4.2 FDDI Node Types

One of the really neat things about FDDI is there are options for how you can configure it. Will you use fiber or copper? How many nodes and concentrators should be supported? What types of concentrators should you use? FDDI offers a lot of choices for you.

The four main node types in the FDDI environment are:

- **Single attachment station (SAS) —** Connects to the FDDI ring through a single connector. The connector has an input port and an output port. Data is received on the input port and is sent to the downstream neighbor via the output port. The SAS connects to a concentrator and then to the primary ring only.
- **Single attached concentrator (SAC) —** Like the SAS, the SAC concentrator connects to only the primary ring. The connection is made through another concentrator.
- **Dual attachment station (DAS)** — Connects to the FDDI ring through two connectors (each with an input and an output port). Can connect directly to the ring or through a concentrator.

POP QUIZ

What does the acronym *FDDI* stand for?

- **Dual attached concentrator (DAC)** — A concentrator that connects to both rings.

7.1.4.3 The FDDI Frame Format

The FDDI frame format is very similar to the format of a Token Ring frame. FDDI uses either token frames or token data frames. Figure 7-10 shows an example of a token frame.

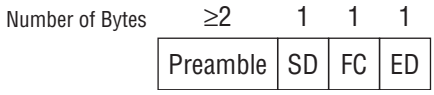


Figure 7-10 An empty token frame

- **Preamble** — Provides a vehicle to ensure the receiving node is synchronized to receive the frame.
- **SD** (start of frame delimiter) — This field lets the receiving node know when the frame begins.

- **FC** (frame control) — This field is used to separate network management data frames from user data frames.
- **ED** (end of frame delimiter) — This field lets the receiving node know when the frame ends.

The token data frame format is pretty much an extension of the token frame format. The first two fields are identical, but the third field is moved to the end of the frame (where it belongs). There are several fields in between that contain the data and the information a node needs to send and receive frames on the Token Ring.

Figure 7-11 shows the fields contained within the token data frame.

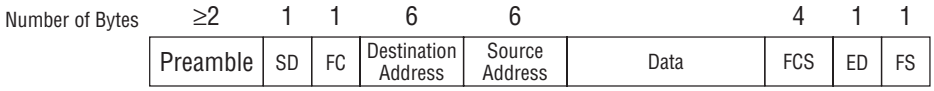


Figure 7-11 A token frame with data attached

- **Preamble** — Provides a vehicle to ensure the receiving node is synchronized to receive the frame.
- **SD** (start of frame delimiter) — This field lets the receiving node know when the frame begins.
- **FC** (frame control) — This field is used to separate network management data frames from user data frames.
- **Destination Address —** This field contains the 6-byte network address of the node the frame is destined for.
- **Source Address —** This field contains the 6-byte network address of the node the frame originated from.
- **Data** — This field contains the data from the upper layer protocol that is being transmitted. There is a certain limit on the amount of data that can be included in the frame. At 4 Mbps, the limit is 4,528 bytes. At 16 Mbps, the limit is 18,173 bytes. At 100 Mbps, the limit is 18,173 bytes.
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field.
- **ED** (end of frame delimiter) — This field lets the receiving node know when the frame ends.

POP QUIZ

What are the four main node types in the FDDI environment?

- **FS** (frame status field) — This field is used by the originating node to detect whether there were any errors during transmission. This includes: if the destination node copied the data; if there were any errors encountered; and even if the destination node recognized itself as the destination node.

7.2 As If You Haven't Had Enough of These Sweet Protocols

It was tough to decide what to include in this section. There are a lot of protocols and other services that you will need to know. For one thing, you will probably come across some, if not all, of them at some point. Additionally, many of the protocols were built upon some networking original protocols, so understanding their function and structure is helpful in understanding the more advanced protocols that have come out in recent years.

The information in this section should really help you start piecing out how things are connected in today's networks. It should also help you better understand the next two parts of this book (especially when you will be tasked to design your own network).

This section is fairly long, but it simply made sense to put it all in here. After reading through this chapter, if you like what we did, you can thank author Jim. If you don't like it, it was author Rich's idea.

7.2.1 Digital Equipment Company Network

The Digital Equipment Company (Digital)[18] developed and released the first version of the *Digital Equipment Company Network (DECnet)* protocol in the mid-1970s. For years, Digital had been developing a series of minicomputers that were known as the *programmed data processor (PDP)*[19] series. DECnet was developed to allow two PDP series 11 (PDP-11) nodes to connect to one another over a point-to-point link and share resources.

[18]Many people in the industry refer to the Digital Equipment Company as ''DEC'' (pronounced ''deck''), but the official ''short name'' is Digital.

[19]Digital decided to use the term programmed data processor (PDP) instead of what it truly was — a computer. This is because computers were known to be complicated and very expensive. To thwart the negative press the computer had developed, the term PDP was used and sold to a market that could not afford a computer.

AN UNRELATED MOMENT OF PAUSE

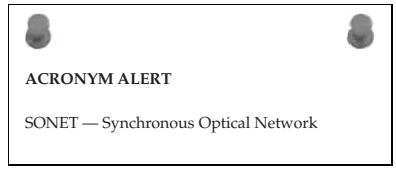
A reporter was given an opportunity to do an exclusive interview with a network engineer who had been sent to the International Space Station to upgrade the network.

Reporter: "So, how do you feel now that you have been there for 3 days?"

Engineer: "Lady, how would you feel if you were stuck in space, floating inside a grouping of about 120,000 parts all bought from the lowest bidder?"

DECnet is not in and of itself a complete single standard; it's a suite of protocols. As with most protocols that continue to have an end-user demand, DECnet has undergone several updates to the original protocols. Following is a brief overview of the DECnet phases:

- **DECnet phase I** — Allowed two PDP-11 series to communicate with one another.
- **DECnet phase II** — Increased support to networks of up to 32 nodes. The nodes did not have to be identical, but were requested to be able to interoperate with each other. Communication between nodes was done via a point-to-point link. File sharing was an important upgrade during this phase.
- **DECnet phase III** — Increased support to networks of up to 255 nodes. Communication was handled via point-to-point link, as well as multidrop links. Support was added to allow DECnet networks to communicate with networks of other types. Routing and network management were also supported at this phase.
- **DECnet phase IV** — Increased support of networks of up to 63 areas, supporting up to 1023 nodes each. Phase IV included Ethernet support as well as some hierarchical routing standards. Also, a client was developed for Microsoft DOS and some Windows platforms that allowed workstation support of the DECnet protocol.
- **DECnet phase V** — IOS standards were rolled into this phase, moving the protocol from a proprietary standard to an open standard. The name phase V was later changed to *DECnet/OSI*, identifying the compatibility with other OSI standards. Eventually, some TCP/IP protocols were added and the name was changed to *DECnet-Plus*.



DECnet phase IV introduced a layered network architecture that is similar to the architecture outlined in the OSI reference model. The DECnet layered

model is known as the *digital network architecture (DNA)*. In the DNA model, each layer serves the layers above it and requests services from the layer beneath it. The structure and purpose of the DNA model are much like the OSI model, each layer being responsible for a function to support the protocol. Each layer is mostly based on the proprietary protocol, so some of the upper layers share functions within individual substandards.

The DNA changed as well when DECnet phase V came about, due to the multiple open standard support that was now part of the protocol. Most of the upper layers support both the proprietary and the open standards that became part of the protocol suite.

Note that you don't have to know all the proprietary standards in the protocol suite; know only that it operates in a hierarchical manner.

7.2.2 Xerox Network Systems

Xerox Network Systems (XNS), developed by the Xerox Corporation[20] in the late 1970s and early 1980s, was a suite of protocols that supported a variety of functions. Although it was never a true competitor to TCP/IP, XNS was adopted by many vendors to run within their LANs.[21]

XNS also utilized a reference model that roughly matched the OSI reference model. There were a total of five levels[22] in the XNS reference model:

- **Level 0** — Roughly corresponded with the OSI Layers 1 and 2.
- **Level 1** — Roughly corresponded with the OSI Layer 3.
- **Level 2** — Roughly corresponded to the OSI Layers 3 and 4.
- **Level 3** — Roughly corresponded to the OSI Layers 7 and 7.
- **Level 4+** — Roughly corresponded to the OSI Layer 7.

XNS used a routing protocol called the *Internet Datagram Protocol (IDP)*, which was responsible for datagram delivery within a network as well as an addressing scheme for the routing of said datagrams. Because the format of the IDP packet differed[23] from some other routing protocols, we wanted to break down the packet for you in Figure 7-12 so you can see the fields that are contained in the packet.

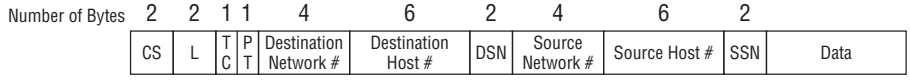


Figure 7-12 The IDP packet format

[20]That was pretty obvious, wasn't it?
[21]XNS was modified for several of these companies to suit the needs of their particular network.
[22]Not layers.
[23]For one thing, the IDP network address contains the following: a 4-byte network number, a 6-byte host address, and a 2-byte socket field.

- **CS** (checksum) — Used to determine the integrity of the packet upon receipt by the destination.
- **L** (length) — Identifies the length of the packet.
- **TC** (transport control) — This field actually contains two subfields. The first subfield identifies the current hop count for the packet. The other subfield identifies the maximum time the packet can live on the network.
- **PT** (packet type) — Identifies the format of the packet.
- **Destination Network #** (destination network number) — The 4-byte destination network identifier.
- **Destination Host #** (destination host number) — The 6-byte destination host identifier.
- **DSN** (destination socket number) — The 2-byte destination socket identifier.
- **Source Network #** (source network number) — The 4-byte source network number.
- **Source Host #** (source host number) — The 6-byte source host identifier.
- **SSN** (source socket number) — The 2-byte source socket identifier.
- **Data** (data) — The payload!

POP QUIZ

What are DECnet's five phases?

7.2.3 Internetwork Packet Exchange

The *Internetwork Packet Exchange (IPX)* protocol is normally found within networks with nodes running the Novell NetWare operating system. Novell NetWare was built to support the protocols that were a part of the XNS protocol suite. IPX is a datagram protocol used to route packets within a network. It is connectionless-oriented protocol (IP, for example) and therefore does not have to ensure a connection before it puts the packet onto the transport medium.

IPX uses a distance-vector protocol (RIP, for example), making routing decisions based on hop counts. IPX RIP works similarly to RIP, but instead of using a hop count for distance determination it uses what is known as a *tick*. A tick is simply a measure of time ($1/18^{th}$ of a second) delay that is expected for a particular distance on the medium. If there are two routes to the destination and the ticks are the same on each path, the route with the lowest hop count is the one that will be chosen.

IPX uses an IPX address for host/node identification. There are two parts to the IPX address. The first part of the IPX address is the *network* number: the remaining part is known as the *node* number. The network number is 4 bytes long (that's a total of 32 bits for those of you who are counting).[24] The node number is 6 bytes long (48 bits), which happens to match the length of the MAC address of the NIC. Why does it match? Because the MAC (IEEE 802) address is the number that is used for the node number part of the IPX address. Figure 7-13 is an example of the IPX address.

Figure 7-13 The IPX address

RANDOM BONUS DEFINITION

workgroup switch — A switch used within a single department or workgroup.

Because the node has its own MAC address, the only requirement you need to have an IPX address assigned to the node is to plug it into an interface to the network. The node will send out a broadcast letting the network know it has joined the network. The appropriate router will then assign the network number to the node. The node now has identification and can send and receive IPX datagrams. IPX is simple to implement — it is basically plug and play.

By now you have to be asking if there is anything complicated about IPX. The answer is no, but there is something you need to know about the IPX datagram format: there is not just a single datagram format. Why? Originally, IPX frame formats served well on the early Ethernet networks within a single network. But as networks grew and as LANs began communicating with one another, other standards were introduced and existing standards were improved, and IPX could not support communication with nodes outside of their known network number–which is why four Ethernet frame formats are used.[25]

Novell proprietary frame format — This is the original frame format that was used. It is often referred to as *802.3 raw* (minus the LLC [802.2]). Figure 7-14 is an example of this.

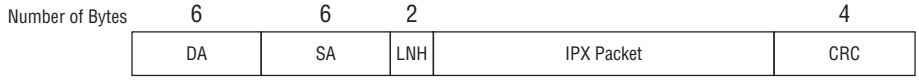


Figure 7-14 The 802.3 raw frame format

[24]If you are counting, or even thought of counting, then you get extra credit! Great job!
[25]The router is responsible for translating and reformatting different formats so the destination can understand the information within the frame.

- **DA** (destination address) — The 6-byte destination MAC address.
- **SA** (Source address) — The 6-byte source MAC address.
- **LNH** (length) — This field identifies the amount of data contained in the data payload field.
- **IPX Packet** — This is the IPX datagram portion of the frame. The following subfields are part of the IPX packet:

  CS (checksum) — This field is normally not used. If it is used, then it is not compatible with the Novell proprietary format.

  PL (packet length) — The length of the IPX packet.

  TC (transport control) — The hop count (this is an incrementing field).

  PT (packet type) — Identifies the format of the data in the payload portion of the packet.

  DNN (destination network number) — The 4-byte destination network identifier.

  DHN (destination host number) — The 6-byte destination host identifier.

  DSN (destination socket number) — The 2-byte destination socket identifier.

  SNN (source network number) — The 4-byte source network number.

  SHN (source host number) — The 6-byte source host identifier.

  SSN (source socket number) — The 2-byte source socket identifier.

  Data — The payload!

  CRC (cyclic redundancy check) — This is a 4-byte value that is part of the frame check sequence (FCS), used to determine if a frame is intact at the receiving end.

802.3 frame format — This is the same format used by Ethernet, followed by the IPX data payload. Figure 7-15 is an example of this.

POP QUIZ

Which operating system uses IPX?

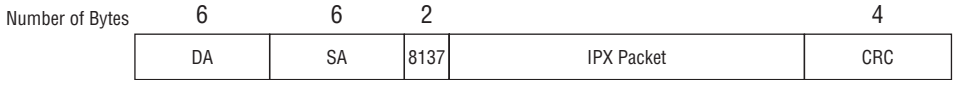


Figure 7-15 The 802.3 frame format

- **DA** (destination address) — The 6-byte destination MAC address.

- **SA** (source address) — The 6-byte source MAC address.
- **LNH** (length) — This field identifies the amount of data contained in the data payload field.
- **IPX Packet** — This is the IPX datagram portion of the frame. The following subfields are part of the IPX packet:

  CS (checksum) — This field is normally not used. If it is used, then it is not compatible with the Novell proprietary format.

  PL (packet length) — The length of the IPX packet.

  TC (transport control) — The hop count (this is an incrementing field).

  PT (packet type) — Identifies the format of the data in the payload portion of the packet.

  DNN (destination network number) — The 4-byte destination network identifier.

  DHN (destination host number) — The 6-byte destination host identifier.

  DSN (destination socket number) — The 2-byte destination socket identifier.

  SNN (source network number) — The 4-byte source network number.

  SHN (source host number) — The 6-byte source host identifier.

  SSN (source socket number) — The 2-byte source socket identifier.

  Data (data) — The payload!

  CRC (cyclic redundancy check) — This is a 4-byte value that is part of the frame check sequence (FCS), used to determine if a frame is intact at the receiving end.

802.3 with 802.2 frame format — The header of this format is the same format used by IEEE 802.3, then comes the LLC header, and finally the IPX data payload. Figure 7-16 is an example of this.

RANDOM BONUS DEFINITION

access priority — The priority used to determine access privileges on a shared LAN segment.

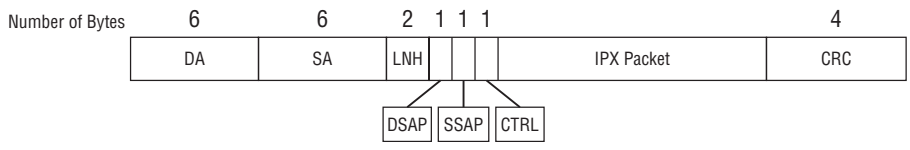


Figure 7-16 The 802.3 with 802.2 frame format

- **DA** (destination address) — The 6-byte destination MAC address.
- **SA** (source address) — The 6-byte source MAC address.
- **LNH** (length) — This field identifies the amount of data contained in the data payload field.
- **DSAP** (destination service access point) — This field identifies which service access points[26] the LLC information should be delivered to.
- **SSAP** (source service access point) — This field identifies the service access point the data originated from.
- **CTRL** (control) — This field contains information used by the LLC on the receiving node that identifies the LLC frame type.
- **IPX Packet** — This is the IPX datagram portion of the frame. The following subfields are part of the IPX packet:

  CS (checksum) — This field is normally not used. If it is used, then it is not compatible with the Novell proprietary format.

  PL (packet length) — The length of the IPX packet.

  TC (transport control) — The hop count (this is an incrementing field).

  PT (packet type) — Identifies the format of the data in the payload portion of the packet.

  DNN (destination network number) — The 4-byte destination network identifier.

  DHN (destination host number) — The 6-byte destination host identifier.

  DSN (destination socket number) — The 2-byte destination socket identifier.

  SNN (source network number) — The 4-byte source network number.

  SHN (source host number) — The 6-byte source host identifier.

  SSN (source socket number) — The 2-byte source socket identifier.

  Data (data) — The payload!

  CRC (cyclic redundancy check) — This is a 4-byte value that is part of the frame check sequence (FCS), used to determine if a frame is intact at the receiving end.

Sub-network Access Protocol (SNAP) frame format — Uses the IEEE 802.3 standard header, LLC header, SNAP header, and finally the IPX data payload. Figure 7-17 is an example of this.

[26] A *service access point (SAP)* is a label that is assigned to endpoints in a network.

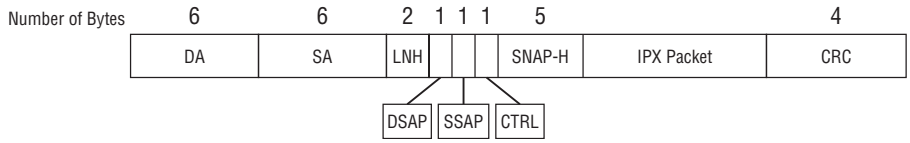


Figure 7-17 The SNAP frame format

- **DA** (destination address) — The 6-byte destination MAC address.
- **SA** (source address) — The 6-byte source MAC address.
- **LNH** (length) — This field identifies the amount of data contained in the data payload field.
- **DSAP** (destination service access point) — This field identifies which service access points that the LLC information should be delivered to.
- **SSAP** (source service access point) — This field identifies the service access point that the data originated from.
- **CTRL** (control) — This field contains information used by the LLC on the receiving node that identifies the LLC frame type.
- **SNAP-H** (Sub-network Access Protocol[27] header) — There are two subfields contained within this):

  VC (vendor code) — This identifies the vendor code of the source.

  ET (ether type) — This identifies the version of Ethernet being used.
- **IPX Packet** — This is the IPX datagram portion of the frame. The following subfields are part of the IPX packet:

  CS (checksum) — This field is normally not used. If it is used, then it is not compatible with the Novell proprietary format.

  PL (packet length) — The length of the IPX packet.

  TC (transport control) — The hop count (this is an incrementing field).

  PT (packet type) — Identifies the format of the data in the payload portion of the packet.

  DNN (destination network number) — The 4-byte destination network identifier.

  DHN (destination host number) — The 6-byte destination host identifier.

  DSN (destination socket number) — The 2-byte destination socket identifier.

[27]SNAP is an extension of LLC.

SNN (source network number) — The 4-byte source network number.

SHN (source host number) — The 6-byte source host identifier.

SSN (source socket number) — The 2-byte source socket identifier.

Data (data) — The payload!

- **CRC** (cyclic redundancy check field) — This is a 4-byte value that is part of the frame check sequence (FCS), used to determine if a frame is intact at the receiving end.

All of you Token Ring fans, don't fret. IPX also can be encapsulated and transmitted on a Token Ring network. Figure 7-18 shows the format of the Token Ring frame.

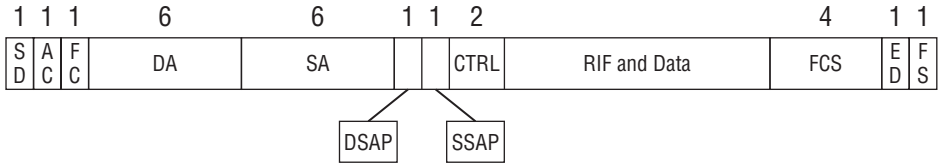


Figure 7-18 The IPX Token Ring frame format

- **SD** (start of frame delimiter) — This field lets the receiving node know when the frame begins.
- **AC** (access control) — There are four subfields in the access control field, all used to transmit information to the access control process within Token Ring.
- **FC** (frame control) — This field is used to separate network management data frames from user data frames.
- **DA (**destination address) — This field contains the 6-byte network address of the node the frame is destined for.
- **SA (**source address) — This field contains the 6-byte network address of the node the frame originated from.
- **DSAP** (destination service access point) — This field identifies which service access points[28] the LLC information should be delivered to.
- **SSAP** (source service access point) — This field identifies the service access point that the data originated from.
- **CTRL** (control) — This field contains information that is used by the LLC on the receiving node that identifies the LLC frame type.

[28] A *service access point (SAP)* is a label assigned to endpoints in a network.

- **RIF** (routing information) — This field assists in ensuring the Token Ring frame is sent in the correct direction.
- **Data** — The payload!
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field.
- **ED** (end of frame delimiter) — This field lets the receiving node know when the frame ends.
- **FS** (frame status) — This field is used by the originating node to detect whether there were any errors during transmission. This includes if the destination node copied the data, if there were any errors encountered, and even if the destination node recognized itself as the destination node.

POP QUIZ

True or false: IPX is not supported on a Token Ring network.

7.2.4 Point-to-Point Protocol

The *Point-to-Point Protocol (PPP)* is really not a protocol at all; rather, it is a suite of protocols that work to allow IP data exchange over PPP links. Prior to the release of PPP, the standard that was being used for IP serial link transmission was the *Serial Link Internet Protocol (SLIP)*. SLIP did a decent job of transmitting the IP data, but it wasn't reliable, wasn't secure, and really wasn't able to support the performance demands of end users. Additionally, SLIP was used in LANs where the cabling wasn't long at all — SLIP just couldn't support communication over longer distances. PPP was developed to address these issues, as well as support serial communication for many network layer protocols, not just IP.

To support the multiple protocol datagrams, PPP uses the following three main components:

- PPP encapsulation method
- PPP Link Control Protocol (LCP)
- PPP Network Control Protocol (NCP)

7.2.4.1 PPP Encapsulation Method

PPP specifies a frame format that is to be used to encapsulate higher layer data. The format is based on the format used for the *High-level Data Link Control (HDLC) protocol*. HDLC is a synchronous Data Link layer protocol developed by the ISO and used as a reference for the PPP standard.

7.2.4.2 PPP Link Control Protocol

LCP is the foundation protocol of the PPP protocol suite. It is the big kahuna in PPPland, supervising all the other protocols to ensure that they are performing the actions they are responsible for. LCP controls the PPP links. The processes involved in setting up and negotiating the rules for a link, managing the activity on the link, and closing the link when the data transmission is complete are all functions overseen by LCP.

7.2.4.3 PPP Network Control Protocol

NCP is the control protocol that ensures the correct Layer 3 protocol is being used. NCP establishes which network layer protocol is required and then it sets the parameters needed to ensure that data can be recognized and understood at the endpoint. PPP supports multiple NCPs running on the same link, regardless of the type or which of the Layer 3 protocols is being supported.

7.2.4.4 Please, Tell Us More

PPP has to set up a PPP link in order to communicate to the destination. The first node will test the link by sending an LCP frame. Once LCP has set up the link and all of the session parameters have been negotiated between the endpoints, NCP frames are then sent to set up and configure the parameters for the particular NCP type to be used. Once all these steps have occurred, packets can be sent. The link remains established until it is no longer needed or something external[29] causes link failure.

7.2.4.5 PPP Frame Format

We previously mentioned that PPP was designed based on the HDLC protocol. The frame format is the same for PPP and HDLC; however, PPP does not use all the fields. Therefore, some fields are set to a standard number for PPP.[30] Figure 7-19 depicts the PPP frame format.

- **Flag** — The PPP Flag field is always set to binary 01111110. This field indicates the start point and end point of the frame.

[29] In other words, PPP didn't do it.
[30] Why reinvent the wheel?

- **BA** (broadcast address) — This field is set to binary 11111111.
- **CTRL** (control) — This field is used by HDLC and is used for certain control parameters. The PPP control field is always set to binary 00000011.
- **Protocol** — This field identifies the protocol type for the information contained in the data payload.
- **Data** — The payload!
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field.
- **Flag** — The PPP Flag field is always set to binary 01111110. This field indicates the start point and end point of the frame.

POP QUIZ

What serial transmission standard was used before PPP came out?

Figure 7-19 The PPP frame format

7.2.5 X.25

X.25 is a Network layer protocol standard that is maintained by the International Telecommunication Union – Telecommunication standardization sector (ITU-T). Used within packet-switched networks, X.25's purpose in networking is to provide the rules on how connections between nodes are set up and maintained. X.25 protocols[31] allow communication between different networks, regardless of what equipment and protocols they are running. Communication between the networks is actually handled through an intermediary (more on this in a little bit) at the Network layer. X.25 is a reliable connection-oriented standard of protocols.

X.25 uses the following three main types of nodes (see Figure 7-20):

- **Data terminal equipment (DTE)** — Nodes that communicate on the X.25 network (these are the computers and nodes that connect the user to a network). Think of the DTE as the user nodes.

[31] Did you notice the *s*? Yep — it's a suite of protocols, not really a single protocol.

- **Data circuit-terminating equipment (DCE)**[32] — A network access point (normally a modem or packet switch that is the interface to the *cloud*).[33] Think of the DCE as the network nodes.
- **Date switching exchange (DSE)**[34] — The nodes that are in the cloud. These nodes are responsible for passing data from DTE to DTE.

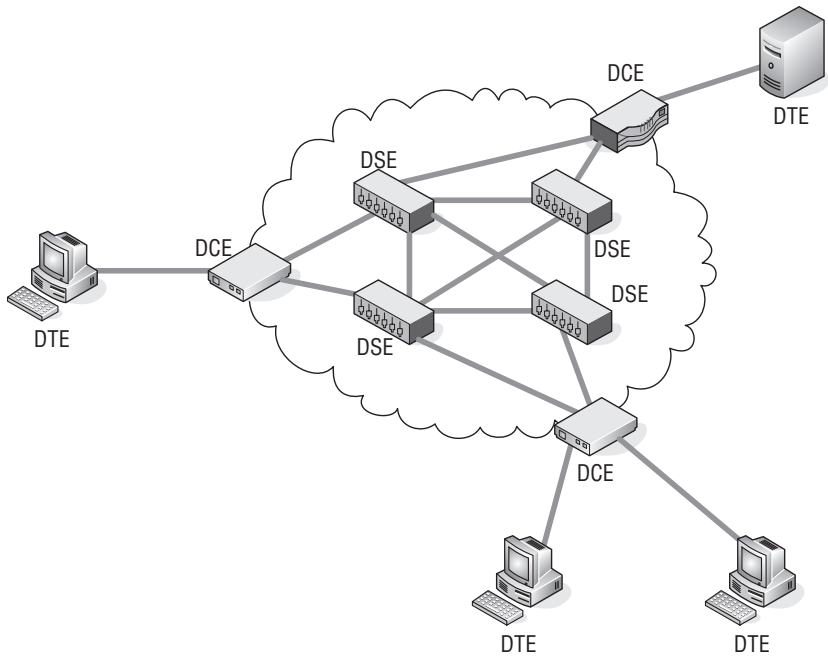


Figure 7-20 Deployments of the X.25 node types

In X.25 data transmission operations, every DTE must have an association with a DCE. Don't confuse DTE and DCE as being single standalone network nodes. DTE and DCE are actually the functions performed. As a matter of fact, a single node can provide multiple functions (for instance, a node can be both a DCE and a DSE.

DCEs and DSEs are the nodes that route the packets through the cloud to a destination. Each and every packet that is transmitted may take a different

[32]Also known as data communications equipment and data carrier equipment.

[33]*Cloud* is a term that defines the WAN infrastructure. Normally networks connect using a communication protocol (such as X.25). There is usually a switch that is the interface to the cloud. Once a packet hits the cloud, the provider is responsible for routing data to a destination. What goes on in the cloud stays with the cloud — meaning the endpoint networks don't necessarily care how the provider is getting the data there, just as long as it gets there.

[34]Also known as packet switching exchange (PSE).

path to get to the destination DCE and ultimately the destination DTE. Usually, the DTE connects to the DCE over some type of network, but two nodes can be connected directly. When there is a direct connection between nodes, then one of the nodes has to perform the functions of a DCE.

The DTE is responsible for serving multiple sessions over a single connection to the DCE. Each and every session first needs to be connected to the DCE. Once the connections are established, the transmission of data can occur. Figure 7-21 is a basic diagram that depicts the session setup and processes.

RANDOM BONUS DEFINITION

broadcast address — The well-known multicast address defining all nodes.

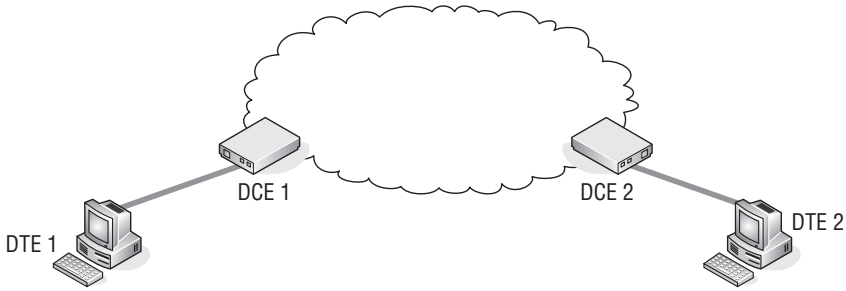


Figure 7-21 A basic X.25 network

A session can be established in one of three ways (refer to Figure 7-21):

- The DTE can send a message to the DCE, letting the DCE know it has data to transmit. For instance, DTE 1 contacts DCE 1 and lets the DCE know it has data to transmit to DTE 2. This is known as a *switched virtual circuit (SVC)*.
- A DCE can receive a message from another DCE, letting the DCE know that a DTE is requesting to send data to another DTE. For instance, DCE1 informs DCE 2 that DTE 1 wishes to pass data to DTE 2.
- The session can be left up at all times. In this scenario, as far as the DTEs are concerned, they can just pass the data to the destination DTE whenever they have data to send. No session setup is required. This is known as a *permanent virtual circuit (PVC)*.

THE X.25 PAD

Some DTEs (for instance, dumb terminals) are not complex enough to understand full X.25 functionality. Therefore, they need a little assistance in communicating with the DCE. X.25 also supports a node type that performs just this function (helping the little guy out).

The packet assembler/disassembler (PAD) is a node between the DCE and the DTE that is used to assemble packets, disassemble packets, and buffer data until the DTE is ready to receive.

X.25 was developed and used before the OSI reference model was developed. To understand the protocol X.25, all you have to know is that (with only a few exceptions) operations can be mapped to the functions of the lower three levels (Physical, Data Link, and Network) of the OSI reference model. The three levels of the X.25 suite are as follows:

1. **Physical level** — This level corresponds to the OSI Physical layer. This includes defining all of the electrical and mechanical functions that are used by the physical medium. Some X.25 protocols operating at this level include:
   - V.35
   - X.21bis
   - RS232
2. **Link level** — This level corresponds to the OSI model's Data Link layer. Functions that are performed at this level are the framing of packets, numbering packets, receipt acknowledgment, flow control, error detection, and recovery, etc. The X.25 protocol that operates at this level is *Link Access Procedure, Balanced (LAPB)*.
3. **Packet level** — At this level, data is exchanged between X.25 nodes. The protocol that is used at this level is the *Packet Layer Protocol (PLP)*.

RANDOM BONUS DEFINITION

routing — The passing of data among various networks.

7.2.5.1 X.25 Operations

When an X.25 session is established, the session is assigned a virtual circuit number that is known to only the DTE and its associated DCE. The virtual circuit number is what is used to route the packets to the destination. The

virtual circuit number is normally a shorted number, so the route lookup process is shorted (fewer bits and bytes to look at).

The virtual circuit is nothing more than a path to a destination. A virtual circuit number reinforces the existence of a reliable path from one DTE to another DTE. As mentioned previously, there are two types of virtual circuits: *switched virtual circuits (SVC)* and *permanent virtual circuits (PVC)*. The SVC is a circuit that is established as needed between DTEs. Each time a DTE needs to send data, the SVC will have to be set up before communication occurs and closed when the session terminates. The other type of virtual circuit, the PVC, is set up only once. It is used between DTEs that have a constant need to send data to other DTEs.

Additionally, X.25 supports what is known as *multiplexing*, which means that it can carry multiple sessions over a single physical line. Each session would maintain its own virtual circuit, which will identify the destination DTE. Multiplexing is used when a single DTE has several processes that need to communicate with multiple destinations. Once data arrives at the destination, it is demultiplexed and sent to the appropriate DCE to be passed to the endpoint DTE. Figure 7-22 shows an example of how this works.

ACRONYM ALERT

ATP — AppleTalk Transaction Protocol

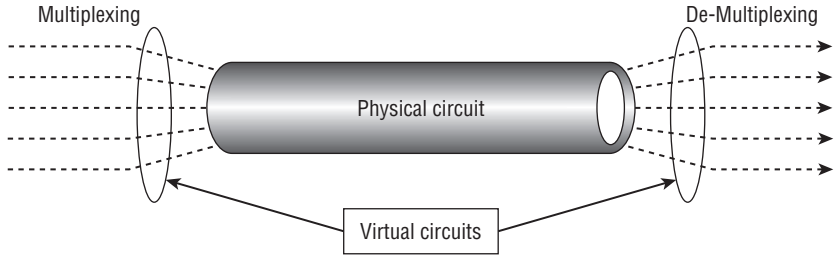


Figure 7-22 A multiplexing example

7.2.5.2 Link Access Procedure, Balanced

The *Link Access Procedure, Balanced (LAPB)* is the X.25 Data Link layer protocol that ensures reliable, error-free packet framing and data communication management. LAPB employs the use of three message frame types:

- **Information frame type** — Frames of this type are known as *I-frames*. I-frames are used to pass upper layer data and some control data. I-frames perform packet sequencing, flow control, and error detection and recovery.

- **Supervisory frame type** — Frames of this type are known as *S-frames*. S-frames are used to pass control data, such as transmission requests, status reporting, I-frame receipt acknowledgements, and termination requests.
- **Unnumbered frame type** — Frames of this type are known as *U-frames*. U-frames are used to pass control data, such as session setup, error reporting, and session termination.

LAPB frames include a header, the PLP data that is being passed to the other end, and a frame trailer. Figure 7-23 shows the format of the LAPB frame.

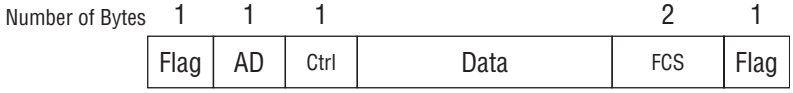


Figure 7-23 The LAPB frame format

- **Flag** — The LAPB Flag field indicates the start point and end point of the frame.
- **AD** (address) — This field identifies whether the frame is carrying a response or a command.
- **CTRL** (control) — This field details which frame type (I-frame, S-frame, or U-frame) is being used, the frame sequence number, and the frame function.
- **Data** — The payload! In LAPD, this is the PLP packet.
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field. This is where error checking and data integrity are monitored.
- **Flag** — The LAPB flag field indicates the start point and end point of the frame.

7.2.5.3 Packet Layer Protocol

The *Packet Layer Protocol (PLP)* is the X.25 Network layer protocol that is used to direct the flow of packets between two DTE nodes over a virtual circuit. PLP can run in conjunction with other protocol standards (for instance, ISDN interfaces on a WAN or LLC within a LAN). There are five defined modes of operation for the PLP:

- **Initial session setup mode** — Used to set up an SVC or PVC between DTE nodes.

- **Data transfer mode** — Used to transfer data between DTEs.
- **Idle mode** — Used by SVCs to keep a session active when no data is being transmitted at the time.
- **Session termination mode** — Used to terminate a session and to clear the SVC.
- **Re-initialization mode** — Used to synchronize data transmission between a DTE and its associated DCE.

7.2.6 Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) is a standard maintained by the ITU-U. Its function is to pass fixed-size datagrams known as *cells* over an ATM network. ATM is a connection-oriented standard, which means the connection is up between nodes before data can be transmitted.[35] Unlike pure packet-switched networks (IP, Ethernet, X.25, etc.), where the frames are of variable lengths, ATM provides *cell-relay* (transmission of data that is encapsulated into a fixed length cell) services on a packet-switched network.

ATM uses nodes that are called *ATM switches*[36] for the transfer of cells within a network. An ATM switch is not a switch in the Layer 2 meaning of the term. It is actually more like a router in functionality.

7.2.6.1 ATM Generic Cell Format

ATM cells are a fixed 53 bytes in size (see Figure 7-24). The first portion of the cell is the header information and is 5 bytes long. The remaining 48 bytes are for the data payload. ATM cells are perfect for passing large amounts of data (streaming video, for example). The fixed length cells do not require the delays that can occur in synchronous data transmission because the variable length packets can cause long upload and download times. Asynchronous transmission, on the other hand, is a steady stream of cells.

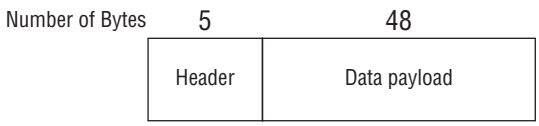


Figure 7-24 The ATM cell format

[35]Repetition – repetition - repetition.
[36]Often nodes are tagged with the word *switch* by the marketing folks out there. It's a buzzword that is often used to impress the customer base.

7.2.6.2 An Overview of ATM Operations

ATM is efficient and reliable. It offers transmission delay (there is no time lapse waiting for your turn), guaranteed to serve constant streams of data and patient enough to wait until data is ready to be passed.

ATM networks contain nodes that are called ATM switches, as well as endpoint nodes that support ATM. ATM switches are responsible for passing data traffic to destination ATM switches and/or ATM endpoint nodes. Endpoint nodes are responsible for interfacing other network types to the ATM network. Examples of endpoint nodes include (see Figure 7-25):

- ATM channel service unit/data service unit (CSU/DSU)
- LAN router
- LAN switch
- LAN workstation

POP QUIZ

Which protocol operates at the packet level of the X.25 model?

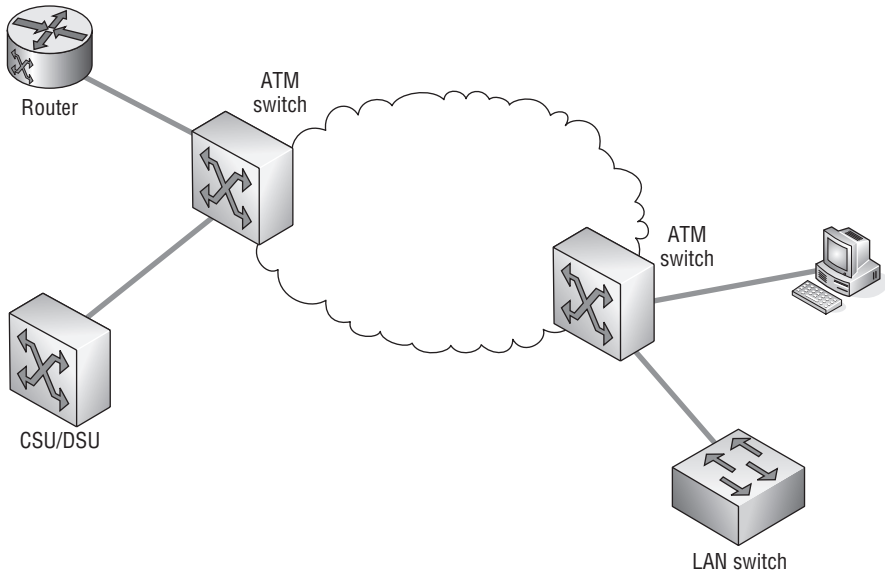


Figure 7-25 An ATM network

7.2.6.2.1 ATM: Virtual Paths, Circuits, and Channels

Closely emulating the virtual circuit concept that is used in X.25, ATM uses what are known as *virtual path identifiers (VPI)* and *virtual circuit identifiers (VCI)*[37] for the routing of cells in an ATM environment. The VPI/VCI pairing

[37] Also known as a *virtual channel identifier*. A channel is basically the same thing as a circuit.

is found in the ATM header and is used to map sessions that are active at any given time. The VPI is used by ATM switches to keep track of the paths to a destination. The backbone switches do not care about the VCI; it's the interfacing nodes (nodes that are outside of the backbone) which include that in path definition. When a switch includes the VCI in its switching decisions, it considers the VPI/VCI pair as a single number.

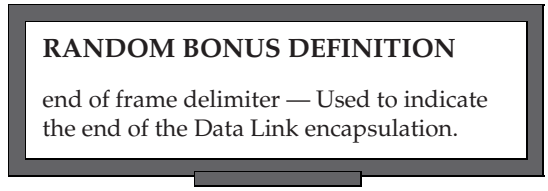
Different types of VPIs and VCIs are used in an ATM network:

Virtual circuit types

- Permanent virtual circuit (PVC) — This is a static virtual circuit.
- Soft permanent virtual circuit (SPVC) — This is a dynamic PVC.
- Switched virtual circuit (SVC) — This is an ''as needed''[38] virtual circuit.

Virtual path types

- Permanent virtual path (PVP) — This is a static virtual path.
- Soft permanent virtual path (SPVP) — This is a dynamic PVP.



The VPI and VCI sessions are identified in the header of the ATM cell. THE VPI is a 12-bit identifier[39] and the VCI is a 16-bit identifier. Virtual circuits must be set up before any data transmission can occur. A virtual path is a group of virtual channels, which are bundled together and transmitted across the ATM network over a shared virtual path. Even though there may be multiple virtual circuits between ATM switches, the VPI and VCI pairing is used only by the endpoint nodes that are involved in the session (see Figure 7-26). Notice how this ATM multiplexing is very similar to the multiplexing processes in X.25.

7.2.6.2.2 ATM: Link Interface Types

There are two primary types of link interfaces used in an ATM environment. *The network-network interface (NNI)* and the *user-network interface (UNI)*. The UNI is the link that connects ATM endpoint nodes to an ATM switch. The NNI is the connection between ATM switches through the cloud.

[38]This could also be ''on demand.''

[39]Four bits of this can be used for *generic flow control (GFC)*, when the communication is taking place between an endpoint node and an ATM switch.

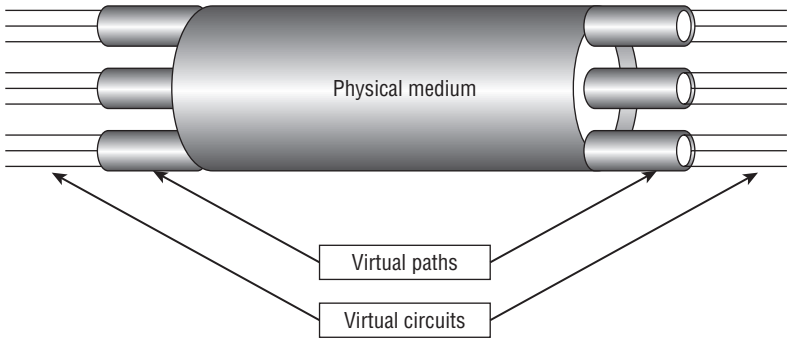


Figure 7-26 ATM multiplexing

Both interface types can be broken up into public UNIs and NNIs or private UNIs and NNIs. Private interface types are used to connect nodes within an ATM topology that is specific to their organization. The public interface types are used to connect nodes on a public network (available to everyone).

7.2.6.2.3 ATM Cell Header Format

The format of the cell header that is used in the ATM cell is determined by the interface type being used. The UNI header (see Figure 7-27) is used for communication between an endpoint node and an ATM switch, while the NNI header (see Figure 7-28) is used for communication between ATM switches.

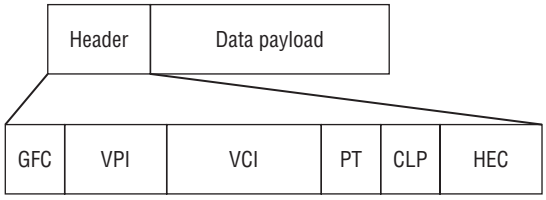


Figure 7-27 The UNI header format

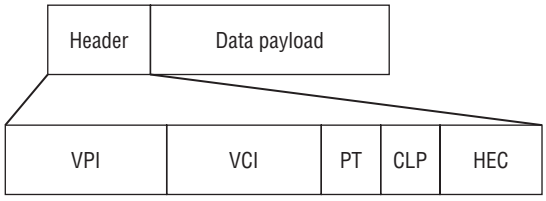


Figure 7-28 The NNI header format

UNI header

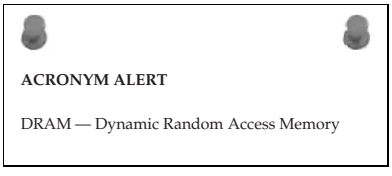
- **GFC** (generic flow control; 4 bits) — Used to assist in identifying the nodes that are part of a shared ATM interface.
- **VPI** (virtual path identifier; 8 bits) — Used to identify the VPI portion of the VCI.
- **VCI** (virtual circuit identifier; 16 bits) — The circuit number used to associate the session's virtual circuit.
- **PT** (payload type; 3 bits) — Identifies the data type in the data payload portion of the ATM cell.
- **CLP** (cell loss priority; 1 bit) — Often referred to as the *discard bit*, set by the sending node for cells that can be discarded if link congestion occurs. Also can be sent by nodes if there is a connection that is exceeding the bandwidth allotment for its session.
- **HEC** (header error control; 8 bits) — The checksum algorithm used for the information contained within the header only for error detection and control.

NNI header

- **VPI** (virtual path identifier; 12 bits) — Used to identify the VPI portion of the VCI.
- **VCI** (virtual circuit identifier; 16 bits) — The circuit number that is used to associate the session's virtual circuit.
- **PT** (payload type; 3 bits) — Identifies the data type in the data payload portion of the ATM cell.
- **CLP** (cell loss priority; 1 bit) — Often referred to as the *discard bit*, set by the sending node for cells that can be discarded if link congestion occurs. Also can be sent by nodes if there is a connection that is exceeding the bandwidth allotment for its session.
- **HEC** (header error control; 8 bits) — The checksum algorithm used for the information contained within the header only for error detection and control.

7.2.6.3 ATM Reference Model

ATM is a protocol suite whose functions are described by a reference model. The ATM reference model uses layers that correspond to the Physical layer and a portion of the



Data Link layer of the OSI reference model. The layers that are part of the ATM reference model are as follows (see Figure 7-29):

- **ATM adaptation layer (AAL)** — Comparable to the functions of the OSI reference model's Data Link layer. This layer is responsible for sorting higher layer data from the ATM processes. This layer combines its services with the service of the ATM layer.
- **ATM layer** — Comparable to the functions of the OSI reference model's Data Link layer. This layer handles the relay of cells through the ATM environment. This layer is also responsible for cell multiplexing.
- **Physical layer** — Responsible for transmission of data on the medium.

POP QUIZ

What are the three virtual circuit types used in ATM?

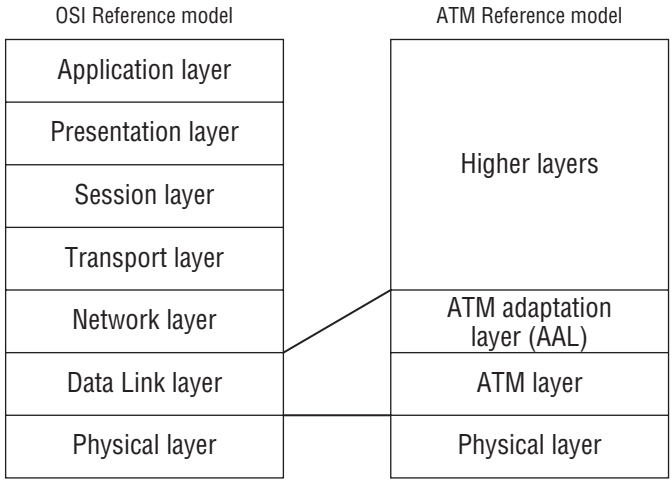


Figure 7-29 A comparison of the OSI and ATM reference models

7.2.6.4 Traffic Management

Several classes of service are defined for user data that is passed within an ATM network. These are as follows:

- **Constant bit rate (CBR)** — Data is passed constantly. The bandwidth required to pass the data is always available.

- **Variable bit rate (VBR)** — Data is passed often. The bandwidth required to pass the data is available, but there are limits on the amount of data that can be passed. The following two types of VBR are used:

  Variable bit rate real-time (VBR-rt) — This is used to pass real-time application data.

  Variable bit rate non-real-time (VBR-nrt) — This is used to temporarily store data in a queue when there is not enough available bandwidth to pass all of the data. It is used with applications that send data, but is not real-time.

- **Available bit rate (ABR)** — Data is passed when bandwidth is available. ABR supports congestion feedback so the sending node will know when there is too much congestion to pass data.
- **Unspecified bit rate (UBR)** — Data is passed if there is bandwidth available, and is dropped if there isn't any available bandwidth. There are no guarantees about delivery.

7.2.6.5 ATM Adaptation Layer Types

The AAL provides interface types that support the service class type that it is assigned to. The type of AAL to be used is determined by the sending node and the type announced when the initial call setup is sent. The AAL types are:

- **AAL1** — Supports CBR transmissions.
- **AAL2** — Supports VBR transmissions.
- **AAL3/4** — Supports both connectionless and connection-oriented data transmission. This AAL type is used to transmit switched multimegabit data services (SMDS)[40] packets.
- **AAL5** — Supports both connectionless and connection-oriented data transmission. This AAL type is used to transmit non-SMDS packets.

RANDOM BONUS DEFINITION

network layer — Layer 3 of the OSI reference model.

[40]SMDS is a connectionless telco service that supports various protocols and functions needed to transmit data over a high-performance packet-switched network. This protocol is outside of the scope of this book, so this footnote should provide all the information that you will need — a basic definition pertaining to the service.

AN UNRELATED MOMENT OF PAUSE

By now, we felt that you might be in need of a study break. To make your break a bit more enjoyable, here is a great peanut butter cookie recipe. Make a couple of batches to enjoy while you continue on with this book. If you are hyper-motivated, you can reread the section on X.25 while the cookies bake. That section is a good lead-in to the next section, "Frame Relay."

Ingredients:

- **1 cup firmly packed brown sugar**
- **1/2 cup peanut butter**
- **1/2 cup softened butter**
- **1 tsp vanilla**
- **1 egg**
- **1 cup sugar**
- **1 1/2 cups flour**
- **1/2 tsp baking powder**
- **1/2 tsp baking soda**
- **1/2 tsp salt**

Preparation steps:

1. **Preheat oven to 375°F.**
2. **Combine brown sugar, butter, and peanut butter in a large bowl. Beat on medium speed until well mixed.**
3. **Add egg and vanilla; continue beating until well mixed.**
4. **Reduce speed to low.**
5. **Add flour, baking powder, baking soda, and salt. Beat until well mixed.**
6. **Shape dough into 1-inch balls; roll in sugar.**
7. **Place the balls 2 inches apart onto ungreased cookie sheets; flatten balls in a crisscross pattern with fork dipped in sugar.**
8. **Bake for 8 to 10 minutes or until edges are lightly browned.**

Bon appétit!

7.2.7 Frame Relay

Frame relay is a WAN protocol that operates as a packet-switched network. Like other packet-switched network protocols, frame relay uses the following:

- Multiplexing
- Variable length datagrams

Frame relay is very similar to X.25, and is often considered the upgraded version of X.25. Because frame relay uses various WAN interface types (such as ISDN) to handle Layer 3 functions, and because communication media has improved, frame relay does not have to do the error checking and recovery that X.25 did. Because there is less chatter, frame relay is able to provide quicker and more reliable data transmission, which pretty much renders X.25 obsolete.

Frame relay services operate at the Physical and Data Link layers of the OSI reference model. Originally designed to operate over ISDN interfaces, it now supports transmission over broadband ISDN and ATM.

ACRONYM ALERT

CSMA/CD — Carrier Sense, Multiple Access with Collision Detection

7.2.7.1 Frame Relay Node Types

If you reread the section on X.25 while your cookies were baking, you will probably remember the X.25 node types are DTE, DCE, or DSE. In frame relay, you cut out the DSE and have the two node types that are used (see Figure 7-30):

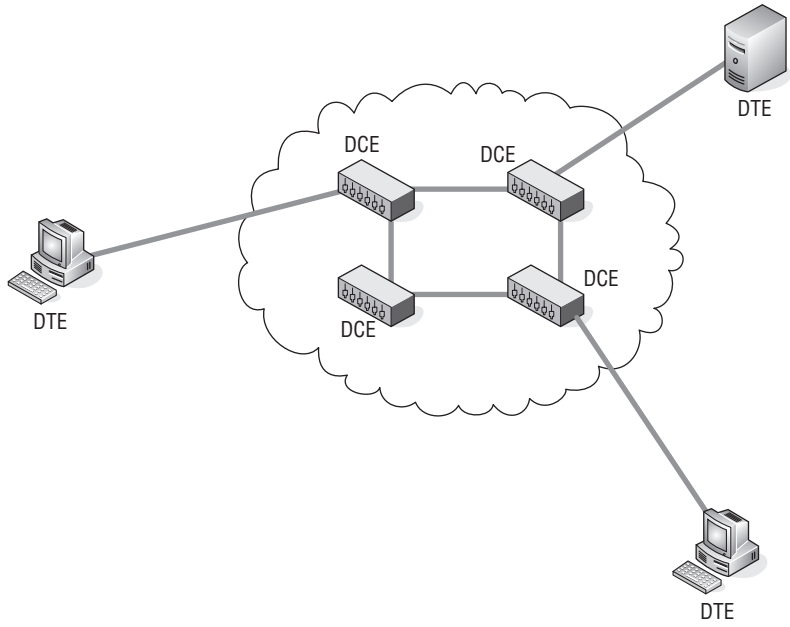


Figure 7-30 DCE and DTE relationship in a frame relay environment

- **DTE** — Nodes that communicate on the frame relay network (these are the computers and endpoint nodes that connect the user to a network). Think of the DTE as the user nodes.

- **DCE** — These are the devices that are within the cloud that transports the data over a WAN. Because the DCEs in frame relay are able to handle the clocking and packet-switching services, there is no need for an intermediary device, like the DSE in X.25

7.2.7.2 Virtual Circuits . . . Again?

Frame relay provides a connection-oriented service at the Data Link layer. Before data can be transmitted, the connection has to be up. The connection is associated with a unique data link connection identifier (see the next section). It is the DLCI that defines the virtual circuit between DTEs. Frame relay supports the multiplexing of virtual circuits to be established over a physical circuit. The frame relay virtual circuit types are:

- SVC — A temporary connection
- PVC — A permanent connection

7.2.7.3 Data Link Connection Identifier

The identifier used to define a circuit is known as the data link connection identifier (DLCI). The DLCI is a value that is normally defined and assigned by the telco provider. The DLCIs are only important to the DTEs. The DCEs normally employ various methods and routes from circuit to circuit. In other words, the DLCI is what allows the data to be passed to the endpoint nodes outside of the cloud. The DCEs make decisions based on whatever technologies are in use by the telco. Because frame relay is a multiplexing WAN protocol, there can be multiple logical circuits passing data through the cloud over a single physical circuit.

POP QUIZ

Frame relay is very similar to _____.

7.2.7.4 Feckens and Beckens[41]

As much as we all may hate to admit it, network congestion occurs more often than we would like it to. It's just a fact of life in a network. Fortunately, there are a lot of checks and balances in most networks that help to prevent errors and to detect and recover from them when they do occur.

Within the frame relay cloud (the provider's portion of the frame relay environment), there can be thousands upon thousands of transmissions passing

[41] These are another pair of fun acronyms similar to catenet (although these are still in use).

through from multiple organizational LANs. All of this data is passing through the same equipment to make its way through the cloud and to a destination. Because of all the end-user data passing through the nodes, congestion does occur.

Frame relay has a couple of functions that help detect congestion and notify the DTEs that congestion is occurring. Additionally, the frame relay header provides an address field that reserves 1 bit for the FECN and one for the BECN. These functions are:

- **Forward explicit congestion notification (FECN)**[42] bit — Within the address field of the frame relay frame header.
- **Backward explicit congestion notification (BECN)**[43] bit — Within the address field of the frame relay frame header.

In addition to the FECN bit and the BECN bit, there is also a bit that is used to indicate if the data is important or not. This field is known as the *discard eligibility (DE)* bit. If the DE bit is ''set,'' the DTE is notifying the DCEs that the frame is low priority and can be discarded if congestion is occurring. This gives the DCEs the capability to prioritize, dropping the data with less importance and only discarding the important data as a last option. The DTEs will retransmit the higher priority data if it gets notification from the DCEs that congestion is occurring.

There are two additional bits in the frame relay frame header that can be set to notify a target node that there is congestion. BECNs are sent to the sending DCEs that there is congestion and FECNs are sent to the target DCEs that there is congestion. Normally, the sending DCE will assume that there are problems if it receives so many BECNs in a certain time period (the number is set by the provider and the subscribing network). It will then cut down on the amount of data it is transmitting[44] or will stop transmitting altogether. When the DTE stops seeing the BECNs, it will return to the way it normally performs.

[42]Pronounced ''fecken.''
[43]Pronounced ''becken.''
[44]Normally, a frame relay provider will promise a minimum transmission rate for a virtual circuit. This is known as the committed information rate (CIR). Often, the provider will allow you to exceed the CIR and will try to pass the data on a best-effort basis. Should your edge router start seeing the BECNs repeatedly outside of the standards you have configured, the CIR should be checked and may need to be adjusted. It could be that multiple frames are being received by a router that has a lower CIR and cannot handle the level of traffic at the time (especially if all of the sending routers are exceeding the CIR).

7.2.7.5 Local Management Interface

For the first few years that frame relay was in use, it didn't really have any standards that ensured that the link was up between DTEs and DCEs. Several companies that were leaders in the networking and telecommunication fields banded together to come up with a signaling standard that would work with frame relay to assist in ensuring the link between a DTE and its associated DCE would remain up. What developed was an enhancement known as the *local management interface (LMI)*.

LMI is used to provide link status updates pertaining to PVCs between a DTE and the local DCE. One of the functions performed by LMI is status inquiries that are sent out periodically (normally 10 seconds) to test if a link is up. If the inquiry does not receive a reply, it assumes the link is down. These inquiries are known as *keepalives*. LMI also sends out updates pertaining to the status of all the links in frame relay network, provides information about PVC changes, and ensures that IP multicast is functioning.

7.2.7.6 Frame Relay Frame Format

The standard frame relay frame format is also known as the LMI version of the frame relay frame. Figure 7-31 shows the fields contained within the frame relay frame.

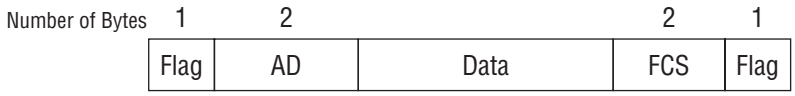


Figure 7-31 Frame Relay frame format

- **Flag** — The frame relay Flag field indicates the start point and end point of the frame.
- **AD** (address) — Included in this field is information pertaining to the DLCI. There are also 3 bits that are included in this field that are for the FECN, BECN, and the DE bit.
- **Data** — The payload!
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field. This is where error checking and data integrity are monitored.
- **Flag** — The frame relay Flag field indicates the start point and end point of the frame.

7.2.8 Integrated Services Digital Network

Integrated Services Digital Network (ISDN) is a data transport service that can be used over regular existing telephone lines. The ISDN service enables the telephone line to be digitized, allowing multiple data types to be passed over existing telephone lines. Additionally, ISDN can be used with digital telephone lines.

ISDN is a baseband transmission standard, used to operate over normal copper lines. Broadband ISDN (B-ISDN) was designed to be faster and more reliable than ISDN. B-ISDN operates over fiber optics. As fiber optics are being rolled into more and more residences and businesses, many ISDN users are using the broadband service.

ISDN provides two types of channels to be used for communication in the ISDN environment, the *B channel* and the *D channel*. The B channel is used to carry user data, whereas the D channel is used for signaling between the end user and the ISDN network. The B channel operates at 64 kbps, and the D channel operates between 16 and 64 kbps, depending on the interface rate standard that is being used.

7.2.8.1 Basic Rate Interface and Primary Rate Interface

The following two services are used in ISDN to determine bandwidth availability between a source and a destination:

- Basic rate interface (BRI)
- Primary rate interface (PRI)

RANDOM BONUS DEFINITION

modem — A node used to pass data communication over an analog communications channel.

The BRI service uses two B channels and one D channel.[45] Each B channel operates at 16 kbps. The BRI D channel operates at 16 kbps as well. The PRI service uses 23 B channels[46] and one D channel.[47] Each B channel operates at 16 kbps, whereas the PRI D channel operates at 64 kbps.

7.2.8.2 ISDN Nodes

Several node types are used in an ISDN environment. Terminals are a node type that can be either an ISDN terminal type, known as a *terminal equipment*

[45]This is referred to as *2B+D*.
[46]PRI in the United States and in Japan includes 23 B channels. Other parts of the world include 30 B channels.
[47]This is referred to as *23B+D*.

type 1 (TE1), or a non-ISDN terminal, known as a *terminal equipment type 2 (TE2)*. The next type of node is called a *terminal adaptor (TA)*, which is used to interface a TE2 with the ISDN network. The next type of node is called a *network termination device type 1 (NT1)* and *network termination device type 2 (NT2)* (or a combination of both). Most ISDN networks will use the NT1.

7.2.8.3 The ISDN Reference Model

ISDN standards span the first three layers of the OSI reference model. At the Physical layer, two different types of frames are used. Which one is used depends on whether the data is flowing from the user node (the terminal) to the ISDN network (TE frame) or from the network to the terminal (NT frame). Figure 7-32 shows the format of the TE frame, and Figure 7-33 shows the format of the NT frame.

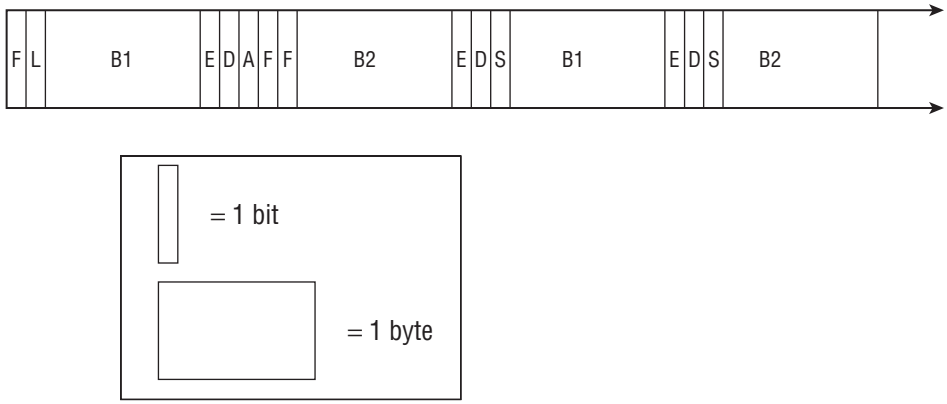


Figure 7-32 The TE frame format

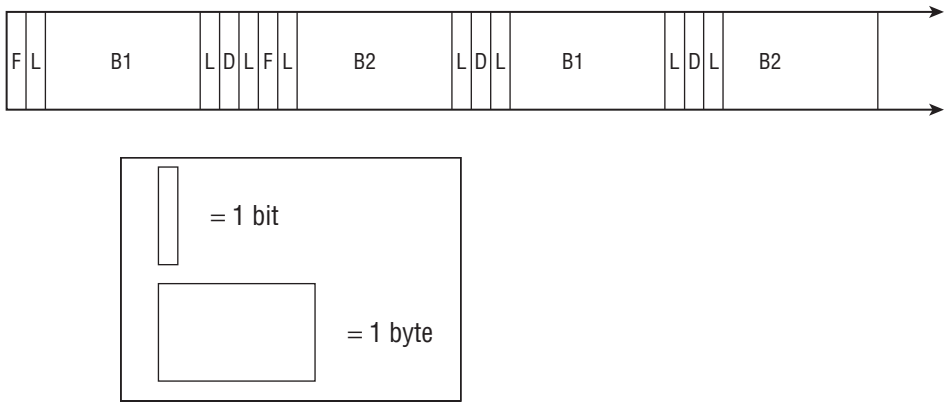


Figure 7-33 The NT frame format

- **F** — Framing bit, marks the beginning of the frame for synchronization.
- **L** — Load balancing bit. These are used to balance the frames signaling.
- **B1** — B1 channel byte. This is B channel data.
- **E** — Echo bit. Echoes D channel data when line congestion is occurring.
- **D** — D channel bit. This is D channel data.
- **A** — Activation bit. Used to activate nodes.
- **B2** — B2 channel byte. This is B channel data.
- **S** — Spare bit.
- **F** — Framing bit. When used, marks the beginning of the frame for synchronization.
- **L** — Load balancing bit. These are used when needed to balance the frames signaling.
- **B1** — B1 channel byte. This is B channel data.
- **D** — D channel bit. This is D channel data.
- **B2** — B2 channel byte. This is B channel data.
- **S** — Spare bit.

The Layer 2 protocol used by ISDN is called the link access procedure D channel (LAPD), which functions like LAPB does for the X.25 protocol. Figure 7-34 shows the LAPD frame format.

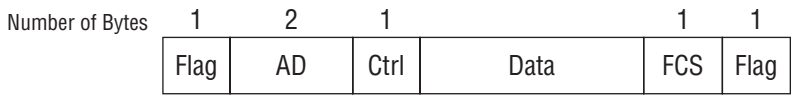


Figure 7-34 The LAPD frame format

- **Flag** — The LAPD Flag field indicates the start point and end point of the frame.
- **AD** (address) — This field identifies whether the frame is carrying a response or a command.
- **CTRL** (control) — This field details which frame type (I-frame, S-frame, or U-frame) is being used, the frame sequence number, and the frame function.
- **Data** — The payload! In LAPD, this is the PLP packet.
- **FCS** (frame check sequence) — This field is a checksum algorithm that checksums the frame from the FC field to the end of the Data field. This is where error checking and data integrity are monitored.

- **Flag** — The LAPD Flag field indicates the start point and end point of the frame.

Finally, two Layer 3 protocols are used by ISDN: ITU-T and ITU-T I.451. These protocols take care of operations at Layer 3, including setting up sessions, establishing and maintaining connections, gathering information pertaining to remote nodes, and other functions.

POP QUIZ

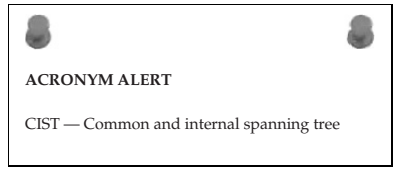
What are the four endpoint node types used in ATM?

7.2.9 AppleTalk

AppleTalk is a protocol suite developed by the Apple Computer company to be integrated with Macintosh computers to allow users to share resources on a network.

AppleTalk came into existence in the 1980s and was one of the first to implement the client/server network architecture. AppleTalk is a plug-and-play service that doesn't require any intervention on the end user's part to connect to a network. The first version of AppleTalk, known as *AppleTalk Phase 1*, was developed mainly for use in a local network segment. It was able to support a maximum of 135 client nodes and 135 server nodes. AppleTalk Phase 2 was developed to support routing outside of the local segment and could support a total of 253 nodes, regardless of whether they were clients or servers.

The services provided and/or supported by AppleTalk span all the layers in the OSI reference model. Figure 7-35 compares the OSI reference model and the AppleTalk protocols that correspond to each layer.



ACRONYM ALERT

CIST — Common and internal spanning tree

7.2.9.1 AppleTalk Physical and Data Link Layers

AppleTalk depends on the same media access protocols to exchange networking data. Each implementation has to work with the AppleTalk suite. At the Physical layer, AppleTalk data can be passed over fiber, twisted pair, and coaxial cabling. AppleTalk interacts with each implementation of a media access protocol to allow AppleTalk data to be exchanged. Following are some of the protocols used at this layer:

- **EtherTalk** — Used on Ethernet networks. The protocol that communicates between the network layer and the Physical layer is known as the *EtherTalk Link Access Protocol (ELAP)*.

- **TokenTalk** — Used on Token Ring networks. The protocol that communicates between the Network layer and the Physical layer is known as the *TokenTalk Link Access Protocol (TLAP)*.
- **FDDITalk** — Used on FDDI networks. The protocol that communicates between the Network layer and the Physical layer is known as the FDDITalk Link Access Protocol (FLAP).
- **LocalTalk** — This is the AppleTalk proprietary standard that is included with all Macintosh computers. This standard is supported on Macintosh nodes only. The protocol that communicates between the Network layer and the Physical layer is known as the *LocalTalk Link Access Protocol (LLAP)*.

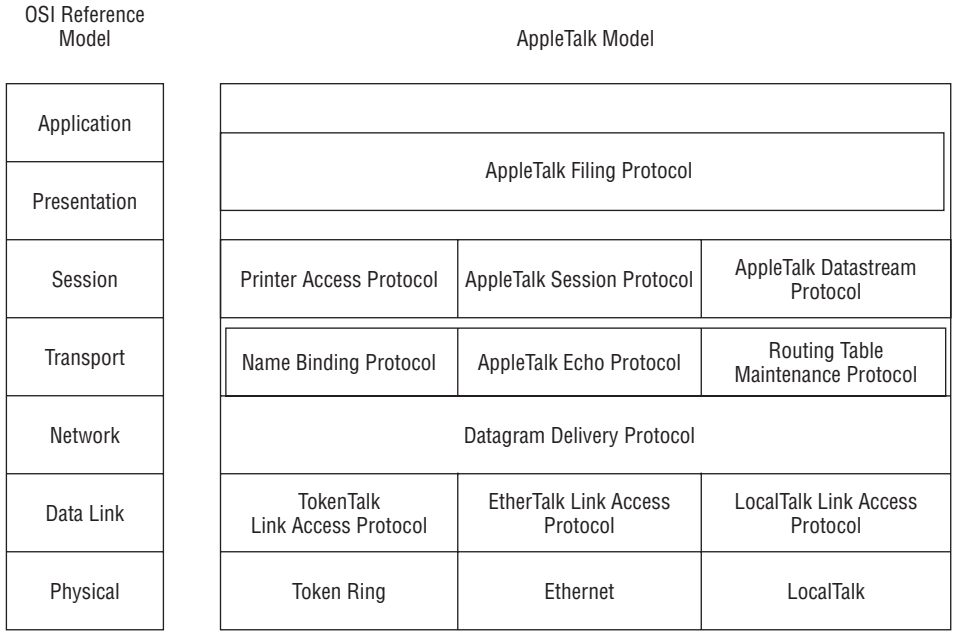


Figure 7-35 The layers of the AppleTalk model

7.2.9.2 AppleTalk Network Layer

The *Datagram Delivery Protocol (DDP)* is the protocol used by AppleTalk at the Network layer. The purpose of DDP in an AppleTalk infrastructure is to provide end-to-end datagram delivery. DDP uses sockets to identify a logical process on a node and as part of the address that is used in order to exchange datagrams. All the upper layers use sockets as well.

All AppleTalk data is formatted to be exchanged in DDP packets over an AppleTalk network. DDP has two different packet types. The short DDP packet type is not used much anymore. It was developed when AppleTalk was limited to segments only. The extended DDP packet type is what is most commonly used.[48]

Another protocol used at this layer is the *AppleTalk Address Resolution Protocol (AARP)*. Just like the Address Resolution Protocol (ARP) for TCP/IP, AARP maps network addresses to their associated data link addresses.

RANDOM BONUS DEFINITION

Layer 3 switch — A router.

7.2.9.3 AppleTalk Upper Layers

AppleTalk uses several upper layer protocols that were built off of the DDP protocol and therefore use DDP as the protocol of choice when information is being passed down to the lower layers for transport across the network.

Transport layer protocols are used for flow control, circuit management, and error checking, detection, and recovery. The AppleTalk protocols included at this layer are:

- **AppleTalk Echo Protocol (AEP**) — The service provided by this protocol is an echo request or an echo reply.
- **AppleTalk Transaction Protocol (ATP)** — Used to pass transmissions between two sockets.
- **Name Binding Protocol (NBP)** — Maintains and manages the use of host names and socket addresses for nodes within the network.
- **Routing Table Maintenance Protocol (RTMP)** — Used to maintain and manage routing information.

Session layer protocols manage communication sessions between Presentation layer processes. The protocols operating at this layer are:

- **AppleTalk DataStream Protocol (ADSP)** — A connection-oriented protocol that provides a data channel for the host nodes.
- **AppleTalk Session Protocol (ASP)** — Maintains and manages higher level sessions.
- **Printer Access Protocol (PAP)** — Maintains and manages virtual connections to printers, print servers, and other server types.

[48]The extended DDP packet is the one most commonly used in new implementations. There is really no good reason to use the short DDP packet, as you need to plan for growth and that packet type limits where your data can be transmitted.

- **Zone Information Protocol (ZIP)** — Manages network numbers and AppleTalk zone names.

RANDOM BONUS DEFINITION

internetwork — A group of networks connected to one another through a router.

The final two layers, the Application and Presentation layers, use the services of the *AppleTalk Filing Protocol (AFP)*[49]. The Presentation layer provides services that are applied to data at the Application layer. Additionally, the Application layer interacts with Macintosh applications (which the OSI Application layer does not).

7.3 Chapter Exercises

1. True or false: The only type of node that is used on a FDDI ring is a FDDI concentrator.
2. What are the three levels of operation within the X.25 protocol suite?
3. In X.25, _____ are used to pass control data, such as: transmission requests, status reporting, _____ receipt acknowledgements, and termination requests.
4. What are the three main components used by PPP?
5. What is the difference between a DTE and a DCE in an X.25 network?
6. What are the Session layer protocols that are used in the AppleTalk protocol suite?
7. What does the acronym ISDN stand for?
8. What is the frame relay local management interface (LMI) used for?
9. What is a constant bit rate (CBR)?
10. _____ is the foundation protocol of the PPP protocol suite.

7.4 Pop Quiz Answers

1. What was the name of the company that developed ARCnet?

   The Datapoint Corporation developed ARCnet in the late 1970s.

[49] AFP is a file sharing protocol.

2. What technology is also known as 1BASE5?

   StarLAN

3. What is the signal called that is passed in Token Ring from one node to the next?

   A token

4. What information is contained in the Destination Address field in a Token Ring frame?

   The Destination Address field contains the 6-byte network address of the node that the frame is destined for.

5. What does the acronym *FDDI* stand for?

   Fiber Distributed Data Interface

6. What are the four main node types in the FDDI environment?
   - Single attached station
   - Single attached concentrator
   - Dual attached station
   - Dual attached concentrator

7. What are DECnet's five phases?
   - DECnet phase I
   - DECnet phase II
   - DECnet phase III
   - DECnet phase IV
   - DECnet phase V

8. Which operating system uses IPX?

   Novell NetWare

9. True or false: IPX is not supported on a Token Ring network.

   False

10. What serial transmission standard was used before PPP came out?

    Serial Link Internet Protocol (SLIP)

11. Which protocol operates at the packet level of the X.25 model?

    Packet Layer Protocol (PLP)

12. What are the three virtual circuit types used in ATM?
    - Permanent virtual circuit (PVC) — This is a static virtual circuit.
    - Soft permanent virtual circuit (SPVC) — This is a dynamic PVC.

- Switched virtual circuit (SVC) — This is an ''as needed''[50] virtual circuit.

13. Frame relay is very similar to _____.

    X.25

14. What are the four endpoint node types used in ATM?
    - ATM customer service unit/digital service unit (CSU/DSU)
    - LAN router
    - LAN switch
    - LAN workstation

[50]This could also be ''on demand.''