Lecture slides prepared by Dr Lawrie Brown (UNSW@ADFA) for “Data and Computer Communications”, 8/e, by William Stallings, Chapter 10 “Circuit Switching and Packet Switching”.
Circuit Switching and Packet Switching

*He got into a District Line train at Wimbledon Park, changed on to the Victoria Line at Victoria and on to the Jubilee Line at Green Park for West Hampstead. It was a long and awkward journey but he enjoyed it.*

— *King Solomon's Carpet*, Barbara Vine (Ruth Rendell)

This quote is from the start of Stallings DCC8e Ch10 hints at the issue of choosing an appropriate path for information to flow over through a network. The chapter begins with a general discussion of switched communications networks. The remainder of the chapter focuses on wide area networks and, in particular, on traditional approaches to wide area network design: circuit switching and packet switching.
For transmission of data beyond a local area, communication is typically achieved by transmitting data from source to destination through a network of intermediate switching nodes; this switched network design is typically used to implement LANs as well. The switching nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. Stallings DCC8e Figure 10.1 illustrates a simple network. The devices attached to the network may be referred to as stations. The stations may be computers, terminals, telephones, or other communicating devices. We refer to the switching devices whose purpose is to provide communication as nodes. Nodes are connected to one another in some topology by transmission links. Node-station links are generally dedicated point-to-point links. Node-node links are usually multiplexed, using either frequency division multiplexing (FDM) or time division multiplexing (TDM). In a switched communication network, data entering the network from a station are routed to the destination by being switched from node to node. For example, in Figure 10.1, data from station A intended for station F are sent to node 4. They may then be routed via nodes 5 and 6 or nodes 7 and 6 to the destination.
Each station attaches to a node, and the collection of nodes is referred to as a communications network. Some nodes connect only to other nodes (e.g., 5 and 7 on previous slide). Their sole task is the internal (to the network) switching of data. Other nodes have one or more stations attached as well; in addition to their switching functions, such nodes accept data from and deliver data to the attached stations. Usually, the network is not fully connected; that is, there is not a direct link between every possible pair of nodes. However, it is always desirable to have more than one possible path through the network for each pair of stations. This enhances the reliability of the network.

Two different technologies are used in wide area switched networks: circuit switching and packet switching. These two technologies differ in the way the nodes switch information from one link to another on the way from source to destination.
Communication via circuit switching implies that there is a dedicated communication path between two stations. That path is a connected sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Communication via circuit switching involves three phases:

1. **Circuit establishment** - Before any signals can be transmitted, an end-to-end (station-to-station) circuit must be established.

2. **Data transfer** - Data can now be transmitted through the network between these two stations. The transmission may be analog or digital, depending on the nature of the network. As the carriers evolve to fully integrated digital networks, the use of digital (binary) transmission for both voice and data is becoming the dominant method. Generally, the connection is full duplex.

3. **Circuit disconnect** - After some period of data transfer, the connection is terminated, usually by the action of one of the two stations. Signals must be propagated to the intermediate nodes to deallocate the dedicated resources. Circuit switching can be rather inefficient. Channel capacity is dedicated for the duration of a connection, even if no data are being transferred. For a voice connection, utilization may be rather high, but it still does not approach 100%. For a client/server or terminal-to-computer connection, the capacity may be idle during most of the time of the connection. In terms of performance, there is a delay prior to signal transfer for call establishment. However, once the circuit is established, the network is effectively transparent to the users.
Circuit switching was developed to handle voice traffic but is now also used for data traffic. The best-known example of a circuit-switching network is the public telephone network (see Stallings DCC8e Figure 10.2 above). This is actually a collection of national networks interconnected to form the international service.

A public telecommunications network can be described using four generic architectural components:

- **Subscribers**: The devices that attach to the network, typically telephones, but the percentage of data traffic increases year by year.
- **Subscriber line**: The link between the subscriber and the network, also referred to as the *subscriber loop* or *local loop*, mostly using twisted-pair wire.
- **Exchanges**: The switching centers in the network. A switching center that directly supports subscribers is known as an end office.
- **Trunks**: The branches between exchanges. Trunks carry multiple voice-frequency circuits using either FDM or synchronous TDM.
Subscribers connect directly to an end office, which switches traffic between subscribers and between a subscriber and other exchanges. The other exchanges are responsible for routing and switching traffic between end offices. This distinction is shown here in Stallings DCC8e Figure 10.3. To connect two subscribers attached to the same end office, a circuit is set up between them. If two subscribers connect to different end offices, a circuit between them consists of a chain of circuits through one or more intermediate offices. In the figure, a connection is established between lines a and b by simply setting up the connection through the end office. The connection between c and d is more complex. In c’s end office, a connection is established between line c and one channel on a TDM trunk to the intermediate switch. In the intermediate switch, that channel is connected to a channel on a TDM trunk to d’s end office. In that end office, the channel is connected to line d.

Circuit-switching technology has been driven by those applications that handle voice traffic. One of the key requirements for voice traffic is that there must be virtually no transmission delay and certainly no variation in delay. A constant signal transmission rate must be maintained, because transmission and reception occur at the same signal rate. These requirements are necessary to allow normal human conversation. Further, the quality of the received signal must be sufficiently high to provide, at a minimum, intelligibility. Circuit switching achieved its widespread, dominant position because it is well suited to the analog transmission of voice signals. In today’s digital world, its inefficiencies are more apparent. However, despite the inefficiency, circuit switching will remain an attractive choice for both local area and wide area networking.
The technology of circuit switching is best approached by examining the operation of a single circuit-switching node. A network built around a single circuit-switching node consists of a collection of stations attached to a central switching unit. The central switch establishes a dedicated path between any two devices that wish to communicate. Stallings DCC8e Figure 10.4 depicts the major elements of such a one-node network. The dotted lines inside the switch symbolize the connections that are currently active.

The heart of a modern system is a digital switch. The function of the digital switch is to provide a transparent signal path between any pair of attached devices. The path is transparent in that it appears to the attached pair of devices that there is a direct connection between them. Typically, the connection must allow full-duplex transmission.

The network interface element represents the functions and hardware needed to connect digital devices, such as data processing devices and digital telephones, to the network. Analog telephones can also be attached if the network interface contains the logic for converting to digital signals. Trunks to other digital switches carry TDM signals and provide the links for constructing multiple-node networks.

The control unit performs three general tasks. First, it establishes connections. This is generally done on demand, that is, at the request of an attached device. Second, the control unit must maintain the connection. Because the digital switch uses time division principles, this may require ongoing manipulation of the switching elements. Third, the control unit must tear down the connection, either in response to a request from one of the parties or for its own reasons.
An important characteristic of a circuit-switching device is whether it is blocking or nonblocking. Blocking occurs when the network is unable to connect two stations because all possible paths between them are already in use. A blocking network is one in which such blocking is possible. Hence a nonblocking network permits all stations to connect at once and grants all possible connection requests as long as the called party is free. When a network is supporting only voice traffic, a blocking configuration is generally acceptable, because it is expected that most phone calls are of short duration and that therefore only a fraction of the telephones will be engaged at any time. However, when data processing devices are involved, these assumptions may be invalid. For example, for a data entry application, a terminal may be continuously connected to a computer for hours at a time. Hence, for data applications, there is a requirement for a nonblocking or "nearly nonblocking" (very low probability of blocking) configuration.
We turn now to an examination of the switching techniques internal to a single circuit-switching node. Space division switching was originally developed for the analog environment and has been carried over into the digital realm. A space division switch is one in which the signal paths are physically separate from one another (divided in space). Each connection requires the establishment of a physical path through the switch that is dedicated solely to the transfer of signals between the two endpoints. The basic building block of the switch is a metallic crosspoint or semiconductor gate that can be enabled and disabled by a control unit. 

Stallings DCC8e Figure 10.5 shows a simple crossbar matrix with 10 full-duplex I/O lines. The matrix has 10 inputs and 10 outputs; each station attaches to the matrix via one input and one output line. Interconnection is possible between any two lines by enabling the appropriate crosspoint. Note that a total of 100 crosspoints is required. The crossbar switch has a number of limitations:

- The number of crosspoints grows with the square of the number of attached stations. This is costly for a large switch.
- The loss of a crosspoint prevents connection between the two devices whose lines intersect at that crosspoint.
- The crosspoints are inefficiently utilized; even when all of the attached devices are active, only a small fraction of the crosspoints are engaged.
To overcome these limitations, multiple-stage switches are employed. Stallings DCC8e Figure 10.6 is an example of a three-stage switch. This type of arrangement has two advantages over a single-stage crossbar matrix:

- The number of crosspoints is reduced, increasing crossbar utilization. In this example, the total number of crosspoints for 10 stations is reduced from 100 to 48.
- There is more than one path through the network to connect two endpoints, increasing reliability.

Of course, a multistage network requires a more complex control scheme. To establish a path in a single-stage network, it is only necessary to enable a single gate. In a multistage network, a free path through the stages must be determined and the appropriate gates enabled. A consideration with a multistage space division switch is that it may be blocking. It should be clear from Stallings DCC8e Figure 10.5 that a single-stage crossbar matrix is nonblocking; that is, a path is always available to connect an input to an output. That this may not be the case with a multiple-stage switch can be seen in Stallings DCC8e Figure 10.6. The heavier lines indicate the lines that are already in use. In this state, input line 10, for example, cannot be connected to output line 3, 4, or 5, even though all of these output lines are available. A multiple-stage switch can be made nonblocking by increasing the number or size of the intermediate switches, but of course this increases the cost.
The technology of switching has a long history, most of it covering an era when analog signal switching predominated. With the advent of digitized voice and synchronous time division multiplexing techniques, both voice and data can be transmitted via digital signals. This has led to a fundamental change in the design and technology of switching systems. Instead of relatively dumb space division systems, modern digital systems rely on intelligent control of space and time division elements.

Virtually all modern circuit switches use digital time division techniques for establishing and maintaining "circuits." Time division switching involves the partitioning of a lower-speed bit stream into pieces that share a higher-speed stream with other bit streams. The individual pieces, or slots, are manipulated by control logic to route data from input to output. There are a number of variations on this basic concept, which are beyond the scope of this book.

<table>
<thead>
<tr>
<th>Time Division Switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>• modern digital systems use intelligent control of space &amp; time division elements</td>
</tr>
<tr>
<td>• use digital time division techniques to set up and maintain virtual circuits</td>
</tr>
<tr>
<td>• partition low speed bit stream into pieces that share higher speed stream</td>
</tr>
<tr>
<td>• individual pieces manipulated by control logic to flow from input to output</td>
</tr>
</tbody>
</table>
The latest trend in the development of circuit-switching technology is generally referred to as the softswitch. In essence, a softswitch is a general-purpose computer running specialized software that turns it into a smart phone switch. Softswitches cost significantly less than traditional circuit switches and can provide more functionality. In particular, in addition to handling the traditional circuit-switching functions, a softswitch can convert a stream of digitized voice bits into packets. This opens up a number of options for transmission, including the increasingly popular voice over IP (Internet Protocol) approach.

In any telephone network switch, the most complex element is the software that controls call processing. This software performs call routing and implements call-processing logic for hundreds of custom calling features. Typically, this software runs on a proprietary processor that is integrated with the physical circuit-switching hardware. A more flexible approach is to physically separate the call processing function from the hardware switching function. In softswitch terminology, the physical switching function is performed by a media gateway (MG) and the call processing logic resides in a media gateway controller (MGC).

Stallings DCC8e Figure 10.7b illustrates the softswitch architecture. In the latter case, the MG and MGC are distinct entities and may be provided by different vendors.
Stallings DCC8e Figure 10.7a contrasts the softswitch architecture with that of a traditional telephone network circuit switch.
The long-haul circuit-switching telecommunications network was originally designed to handle voice traffic, and resources within the network are dedicated to a particular call. With increasing use of data communications its limitations become more apparent.

In contrast a packet-switching network is designed for data use. Data are transmitted in short packets. A typical upper bound on packet length is 1000 octets (bytes). If a source has a longer message to send, the message is broken up into a series of packets. Each packet contains a portion (or all for a short message) of the user’s data plus some control information. The control information, at a minimum, includes the information that the network requires to be able to route the packet through the network and deliver it to the intended destination. At each node en route, the packet is received, stored briefly, and passed on to the next node.
Stallings DCC8e Figure 10.8 illustrates how a longer message is broken up into a series of packets, which are transferred over the packet-switched network.
A packet-switching network has a number of advantages over circuit switching:

- **Line efficiency**: A single node-to-node link can be dynamically shared by many packets over time. The packets are queued up and transmitted as rapidly as possible over the link. By contrast, with circuit switching, time on a node-to-node link is preallocated using synchronous time division multiplexing. Much of the time, such a link may be idle because a portion of its time is dedicated to a connection that is idle.

- **Data rate conversion**: Two stations of different data rates can exchange packets because each connects to its node at its proper data rate.

- **Packets accepted even when network is busy**: When traffic becomes heavy on a circuit-switching network, some calls are blocked; that is, the network refuses to accept additional connection requests until the load on the network decreases. On a packet-switching network, packets are still accepted, but delivery delay increases.

- **Priorities can be used**: Priorities can be used. If a node has a number of packets queued for transmission, it can transmit the higher-priority packets first. These packets will therefore experience less delay than lower-priority packets.
Switching Techniques

- station breaks long message into packets
- packets sent one at a time to the network
- packets can be handled in two ways
  - datagram
  - virtual circuit

If a station has a message to send through a packet-switching network that is of length greater than the maximum packet size, it breaks the message up into packets and sends these packets, one at a time, to the network. A question arises as to how the network will handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination. Two approaches are used in contemporary networks: datagram and virtual circuit.
In the **datagram** approach, each packet is treated independently, with no reference to packets that have gone before. This approach is illustrated in Stallings DCC8e Figure 10.9, which shows a time sequence of snapshots of the progress of three packets through the network. Each node chooses the next node on a packet's path, taking into account information received from neighboring nodes on traffic, line failures, and so on. So the packets, each with the same destination address, do not all follow the same route, and they may arrive out of sequence at the exit point. In this example, the exit node restores the packets to their original order before delivering them to the destination. In some datagram networks, it is up to the destination rather than the exit node to do the reordering. Also, it is possible for a packet to be destroyed in the network. For example, if a packet-switching node crashes momentarily, all of its queued packets may be lost. Again, it is up to either the exit node or the destination to detect the loss of a packet and decide how to recover it. In this technique, each packet, treated independently, is referred to as a datagram.
In the **virtual circuit** approach, a preplanned route is established before any packets are sent. Once the route is established, all the packets between a pair of communicating parties follow this same route through the network. This is illustrated in Stallings DCC8e Figure 10.10. Because the route is fixed for the duration of the logical connection, it is somewhat similar to a circuit in a circuit-switching network and is referred to as a virtual circuit. Each packet contains a virtual circuit identifier as well as data. Each node on the preestablished route knows where to direct such packets; no routing decisions are required. At any time, each station can have more than one virtual circuit to any other station and can have virtual circuits to more than one station.

So the main characteristic of the virtual circuit technique is that a route between stations is set up prior to data transfer. Note that this does not mean that this is a dedicated path, as in circuit switching. A transmitted packet is buffered at each node, and queued for output over a line, while other packets on other virtual circuits may share the use of the line. The difference from the datagram approach is that, with virtual circuits, the node need not make a routing decision for each packet. It is made only once for all packets using that virtual circuit.
If two stations wish to exchange data over an extended period of time, there are certain advantages to virtual circuits. First, the network may provide services related to the virtual circuit, including sequencing and error control. Sequencing refers to the fact that, because all packets follow the same route, they arrive in the original order. Error control is a service that assures not only that packets arrive in proper sequence, but also that all packets arrive correctly. Another advantage is that packets should transit the network more rapidly with a virtual circuit; it is not necessary to make a routing decision for each packet at each node.

One advantage of the datagram approach is that the call setup phase is avoided. Thus, if a station wishes to send only one or a few packets, datagram delivery will be quicker. Another advantage of the datagram service is that, because it is more primitive, it is more flexible. For example, if congestion develops in one part of the network, incoming datagrams can be routed away from the congestion. With the use of virtual circuits, packets follow a predefined route, and thus it is more difficult for the network to adapt to congestion. A third advantage is that datagram delivery is inherently more reliable. With the use of virtual circuits, if a node fails, all virtual circuits that pass through that node are lost. With datagram delivery, if a node fails, subsequent packets may find an alternate route that bypasses that node. A datagram-style of operation is common in internetworks.
There is a significant relationship between packet size and transmission time, as shown in Stallings DCC8e Figure 10.11, which assumes a virtual circuit exists from station X through nodes a and b to station Y. The message to be sent comprises 40 octets, with 3 octets of control information at the beginning of each packet in the header. If the entire message is sent as a single packet of 43 octets (3 octets of header plus 40 octets of data), then the packet is first transmitted from station X to node a (Figure 10.11a). When the entire packet is received, it can then be transmitted from a to b. When the entire packet is received at node b, it is then transferred to station Y. Ignoring switching time, total transmission time is 129 octet-times ($43 \times 3$ packet transmissions).

If we break the message into two packets with 20 octets of message and 3 octets of header each. In this case, node a can begin transmitting the first packet as soon as it has arrived from X. Because of this overlap in transmission, the total transmission time drops to 92 octet-times. By breaking the message into five packets, each intermediate node can begin transmission even sooner, with a total of 77 octet-times for transmission. This process of using more and smaller packets eventually results in increased, rather than reduced, delay as illustrated in Figure 10.11d, since each packet contains a fixed amount of header, and more packets. Also, processing and queuing delays at each node are greater when more packets are handled for a single message. However, we shall see in the next chapter that an extremely small packet size (53 octets) can result in an efficient network design.
Having looked at the internal operation of packet switching, we can now return to a comparison of this technique with circuit switching. In particular we are concerned with three types of delay:

- **Propagation delay**: The time it takes a signal to propagate from one node to the next. This time is generally negligible. The speed of electromagnetic signals through a wire medium, for example, is typically $2 \times 10^8$ m/s.

- **Transmission time**: The time it takes for a transmitter to send out a block of data. For example, it takes 1 s to transmit a 10,000-bit block of data onto a 10-kbps line.

- **Node delay**: The time it takes for a node to perform the necessary processing as it switches data.

Besides performance, there are a number of other characteristics that may be considered. Stallings DCC8e Table 10.1 summarizes the most important of these, many of which have already been discussed. A few additional comments follow. Circuit switching is essentially a transparent service. Once a connection is established, a constant data rate is provided to the connected stations. This is not the case with packet switching, which typically introduces variable delay, so that data arrive in a choppy manner, or even in a different order than they were transmitted. In addition there is no overhead required to accommodate circuit switching. Once a connection is established, the analog or digital data are passed through, as is. For packet switching, analog data must be converted to digital before transmission; in addition, each packet includes overhead bits, such as the destination address.
A simple comparison of circuit switching and the two forms of packet switching is provided in Stallings DCC8e Figure 10.12. The figure depicts the transmission of a message across four nodes, from node 1 to 4.

For circuit switching, there is a delay before the message is sent. First, a Call Request signal is sent. If the destination station is not busy, a Call Accepted signal returns. Note a processing delay is incurred at each node during the call request. On return, this processing is not needed because the connection is already set up. After the connection is set up, the message is sent as a single block, with no noticeable delay at the switching nodes.

Virtual circuit packet switching appears similar to circuit switching. A virtual circuit is requested using a Call Request packet, which incurs a delay at each node, and is accepted with a Call Accept packet. In contrast to the circuit-switching case, the call acceptance also experiences node delays, since each packet is queued at each node and must wait its turn for transmission. Once the virtual circuit is established, the message is transmitted in packets. Packet switching involves some delay at each node in the path. Worse, this delay is variable and will increase with increased load.

Datagram packet switching does not require a call setup. Thus, for short messages, it will be faster than virtual circuit and perhaps circuit switching. However, because each individual datagram is routed independently, the processing at each node may be longer than for virtual circuit packets. For long messages virtual circuits may be superior.
One technical aspect of packet-switching networks remains to be examined: the interface between attached devices and the network. We have seen that a circuit-switching network provides a transparent communications path for attached devices that makes it appear that the two communicating stations have a direct link. However, in the case of packet-switching networks, the attached stations must organize their data into packets for transmission. This requires a certain level of cooperation between the network and the attached stations. This cooperation is embodied in an interface standard. The standard used for traditional packet-switching networks is X.25, which is an ITU-T standard that specifies an interface between a host system and a packet-switching network. The functionality of X.25 is specified on three levels: Physical level, Link level, and Packet level.
X.25 - Physical

- interface between station node link
- two ends are distinct
  - Data Terminal Equipment DTE (user equipment)
  - Data Circuit-terminating Equipment DCE (node)
- physical layer specification is X.21
- can substitute alternative such as EIA-232

The physical level deals with the physical interface between an attached station (computer, terminal) and the link that attaches that station to the packet-switching node. It makes use of the physical-level specification in a standard known as X.21, but in many cases other standards, such as EIA-232, are substituted.
X.25 - Link

- Link Access Protocol Balanced (LAPB)
  - Subset of HDLC
  - see chapter 7
- provides reliable transfer of data over link
- sending as a sequence of frames

The link level standard is referred to as LAPB (Link Access Protocol - Balanced). LAPB is a subset of HDLC (see Chapter 7). The link level provides for the reliable transfer of data across the physical link, by transmitting the data as a sequence of frames.
The packet level provides a virtual circuit service. This service enables any subscriber to the network to set up logical connections, called virtual circuits, to other subscribers. In this context, the term virtual circuit refers to the logical connection between two stations through the network; this is perhaps best termed an external virtual circuit. Earlier, we used the term virtual circuit to refer to a specific preplanned route through the network between two stations; this could be called an internal virtual circuit. Typically, there is a one-to-one relationship between external and internal virtual circuits. However, it is also possible to employ X.25 with a datagram-style network. What is important for an external virtual circuit is that there is a logical relationship, or logical channel, established between two stations, and all of the data associated with that logical channel are considered as part of a single stream of data between the two stations.
An example of X.25 virtual circuits is shown in Figure 10.13 (compare Figure 10.1). In this example, station A has a virtual circuit connection to C; station B has two virtual circuits established, one to C and one to D; and stations E and F each have a virtual circuit connection to D. As an example of how these external virtual circuits are used, station D keeps track of data packets arriving from three different workstations (B, E, F) on the basis of the virtual circuit number associated with each incoming packet.
Stallings DCC8e Figure 10.14 illustrates the relationship among the levels of X.25. User data are passed down to X.25 level 3, which appends control information as a header, creating a packet. This control information serves several purposes, including identifying by number a particular virtual circuit with its associated data, and providing sequence numbers that can be used for flow and error control on a virtual circuit basis.

The entire X.25 packet is then passed down to the LAPB entity, which appends control information at the front and back of the packet, forming a LAPB frame (see Stallings DCC8e Figure 7.7). Again, the control information in the frame is needed for the operation of the LAPB protocol.

The operation of the X.25 packet level is similar to that of HDLC as described in Chapter 7. Each X.25 data packet includes send and receive sequence numbers. The send sequence number, P(S), is used to number sequentially all outgoing data packets on a particular virtual circuit. The receive sequence number, P(R), is an acknowledgment of packets received on that virtual circuit.
The traditional approach to packet switching makes use of X.25, which not only determines the user-network interface but also influences the internal design of the network. The following are key features of the X.25 approach:

- **Call control packets, used for setting up and clearing virtual circuits, are carried on the same channel and same virtual circuit as data packets.**
- **Multiplexing of virtual circuits takes place at layer 3.**
- **Both layer 2 and layer 3 include flow control and error control mechanisms.**

The X.25 approach results in considerable overhead. At each hop through the network, the data link control protocol involves the exchange of a data frame and an acknowledgment frame. Furthermore, at each intermediate node, state tables must be maintained for each virtual circuit to deal with the call management and flow control/error control aspects of the X.25 protocol. All of this overhead may be justified when there is a significant probability of error on any of the links in the network. This approach is not suitable for modern digital communication facilities.
Frame Relay

- designed to eliminate most X.25 overhead
- has large installed base
- key differences:
  - call control carried in separate logical connection
  - multiplexing and switching at layer 2
  - no hop by hop error or flow control
  - hence end to end flow and error control (if used) are done by higher layer
- a single user data frame is sent from source to destination and higher layer ACK sent back

Today's networks employ reliable digital transmission technology over high-quality, reliable transmission links, many of which are optical fiber. In addition, with the use of optical fiber and digital transmission, high data rates can be achieved. In this environment, the overhead of X.25 is not only unnecessary but degrades the effective utilization of the available high data rates. Frame relay is designed to eliminate much of the overhead that X.25 imposes on end user systems and on the packet-switching network. The standards for frame relay matured earlier than those for say ATM. Accordingly, there is a large installed base of frame relay products. The key differences between frame relay and a conventional X.25 packet-switching service are:

- Call control signaling, which is information needed to set up and manage a connection, is carried on a separate logical connection from user data. Thus, intermediate nodes need not maintain state tables or process messages relating to call control on an individual per-connection basis.
- Multiplexing and switching of logical connections takes place at layer 2 instead of layer 3, eliminating one entire layer of processing.
- There is no hop-by-hop flow control and error control. End-to-end flow control and error control are the responsibility of a higher layer, if they are employed at all.

Thus, with frame relay, a single user data frame is sent from source to destination, and an acknowledgment, generated at a higher layer, may be carried back in a frame. There are no hop-by-hop exchanges of data frames and acknowledgments.
The principal potential disadvantage of frame relay, compared to X.25, is that we have lost the ability to do link-by-link flow and error control. Although frame relay does not provide end-to-end flow and error control, this is easily provided at a higher layer. With the increasing reliability of transmission and switching facilities, this is not a major disadvantage.

The advantage of frame relay is that we have streamlined the communications process. The protocol functionality required at the user-network interface is reduced, as is the internal network processing. As a result, lower delay and higher throughput can be expected. Studies indicate an improvement in throughput using frame relay, compared to X.25, of an order of magnitude or more. The ITU-T Recommendation I.233 indicates that frame relay is to be used at access speeds up to 2 Mbps. However, frame relay service at even higher data rates is now available.
Stallings DCC8e Figure 10.15 depicts the protocol architecture to support the frame mode bearer service. We need to consider two separate planes of operation: a control (C) plane, which is involved in the establishment and termination of logical connections, and a user (U) plane, which is responsible for the transfer of user data between subscribers. Thus, C-plane protocols are between a subscriber and the network, while U-plane protocols provide end-to-end functionality.

The control plane for frame mode bearer services is similar to that for common channel signaling for circuit-switching services, in that a separate logical channel is used for control information. At the data link layer, LAPD (Q.921) is used to provide a reliable data link control service, with error control and flow control, between user (TE) and network (NT). This data link service is used for the exchange of Q.933 control signaling messages.

For the actual transfer of information between end users, the user-plane protocol is LAPF (Link Access Procedure for Frame Mode Bearer Services), which is defined in Q.922.
Only the core functions of LAPF are used for frame relay:

- Frame delimiting, alignment, and transparency
- Frame multiplexing/demultiplexing using the address field
- Inspection of the frame to ensure that it consists of an integral number of octets prior to zero bit insertion or following zero bit extraction
- Inspection of the frame to ensure that it is neither too long nor too short
- Detection of transmission errors
- Congestion control functions

- Form sub-layer of data link layer
  - Data transfer between subscribers only

The core functions of LAPF in the user plane constitute a sublayer of the data link layer. This provides the bare service of transferring data link frames from one subscriber to another, with no flow control or error control. Above this, the user may choose to select additional data link or network-layer end-to-end functions. These are not part of the frame relay service. Based on the core functions, a network offers frame relay as a connection-oriented link layer service with the following properties:

- Preservation of the order of frame transfer from one edge of the network to the other
- A small probability of frame loss
As with X.25, frame relay involves the use of logical connections, in this case called data link connections rather than virtual circuits. The frames transmitted over these data link connections are not protected by a data link control pipe with flow and error control. Another difference between X.25 and frame relay is that the latter devotes a separate data link connection to call control. The setting up and tearing down of data link connections is done over this permanent control-oriented data link connection.

The frame relay architecture significantly reduces the amount of work required of the network. User data are transmitted in frames with virtually no processing by the intermediate network nodes, other than to check for errors and to route based on connection number. A frame in error is simply discarded, leaving error recovery to higher layers.

<table>
<thead>
<tr>
<th>Frame Relay Data Link Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>• logical connection between subscribers</td>
</tr>
<tr>
<td>• data transferred over them</td>
</tr>
<tr>
<td>• not protected by flow or error control</td>
</tr>
<tr>
<td>• uses separate connection for call control</td>
</tr>
<tr>
<td>• overall results in significantly less work in network</td>
</tr>
</tbody>
</table>
The operation of frame relay for user data transfer is best explained by considering the frame format. This is the format defined for the minimum-function LAPF protocol (known as LAPF core protocol). The format is similar to that of LAPD and LAPB with one obvious omission: There is no Control field, which means there is only one frame type, used for carrying user data. There are no control frames. It is not possible to perform all control on the connection; a logical connection can only carry user data. It is not possible to perform flow control and error control, because there are no sequence numbers.

The Flag and Frame Check Sequence (FCS) fields function as in HDLC. The information field carries higher-layer data.

The address field has a default length of 2 octets and may be extended to 3 or 4 octets. It carries a data link connection identifier (DLCI) of 10, 16, or 23 bits. The DLCI serves the same function as the virtual circuit number in X.25. As in X.25, the connection identifier has only local significance: Each end of the logical connection assigns its own DLCI from the pool of locally unused numbers, and the network must map from one to the other.
Chapter 10 summary.

Summary

- circuit versus packet switching network approaches
- X.25
- frame relay